LREC-COLING 2024

Joint Workshop on Multiword Expressions and Universal Dependencies (MWE-UD 2024)

Workshop Proceedings

Editors
Archna Bhatia, Gosse Bouma, A. Seza Doğruöz, Kilian Evang, Marcos Garcia, Voula Giouli, Lifeng Han, Joakim Nivre and Alexandre Rademacher

25 May, 2024
Torino, Italia
Preface

Multiword expressions (mwe's) are word combinations that exhibit lexical, syntactic, semantic, pragmatic, and/or statistical idiosyncrasies (Baldwin and Kim, 2010), such as by and large, hot dog, pay a visit and pull someone's leg. The notion encompasses closely related phenomena: idioms, compounds, light-verb constructions, phrasal verbs, rhetorical figures, collocations, institutionalized phrases, etc. Their behavior is often unpredictable; for example, their meaning often does not result from the direct combination of the meanings of their parts. Given their irregular nature, mwe's often pose complex problems in linguistic modeling (e.g. annotation), NLP tasks (e.g. parsing), and end-user applications (e.g. natural language understanding and MT), hence still representing an open issue for computational linguistics (Constant et al., 2017). This joint workshop also marks the 20th anniversary of the mwe workshop series since 2003 (Bond et al., 2003). The organization of the workshops is sponsored by Siglex.¹

Universal Dependencies (ud; De Marneffe et al., 2021) is a framework for cross-linguistically consistent treebank annotation that has so far been applied to over 100 languages. The framework aims to capture similarities as well as idiosyncrasies among typologically different languages (e.g., morphologically rich languages, pro-drop languages, and languages featuring clitic doubling). The goal of developing ud was not only to support comparative evaluation and cross-lingual learning but also to facilitate multilingual natural language processing and enable comparative linguistic studies. Starting with the first ud workshop in 2017 (de Marneffe et al., 2017), this joint workshop is the 7th edition in the series.

For the current edition, the mwe and ud communities decided to organize a joint event, the mwe-ud workshop which is part of LREC-Coling 2024. This is a timely collaboration because the two communities clearly have overlapping interests. For instance, while ud has several dependency relations that are intended for annotation of mwe's, both annotation guidelines (i.e. is syntactic irregularity and inflexibility or semantic non-compositionality the leading criterion?) and annotation practice (both across treebanks for a single language and across languages) for mwe's can be improved (Schneider and Zeldes, 2021). For verbal mwe's, the Parseme corpora for 26 languages provide annotation of mwe's consistent with ud annotation (Savary et al., 2023). Both communities share an interest in developing guidelines, data-sets, and tools that can be applied to a wide range of typologically diverse languages, raising fundamental questions about tokenization, lemmatization, and morphological decomposition of tokens. Proposals for harmonizing annotation practice between what has been achieved in Parseme and ud and expanding Parseme annotation to non-verbal mwe's are also central to the recently started UniDive² cost action (CA21167). UniDive also co-organizes and sponsors this joint workshop.

We are happy to have received 53 submissions, 29 long, 15 short, and 9 non-archival. 19 long, 7 short, and 9 non-archival contributions were selected for presentation at the workshop, bringing the overall acceptance rate for archival papers to 59%. The distribution over tracks is almost even: 8 of 12 archival submissions were accepted for the ud track, 9 of 16 for the mwe track, and 9 of 16 for the mwe+ud track. One long paper was withdrawn after acceptance.

An important theme in both the ud and mwe community is increasing the number of languages and language families that can be used as the object of study, for instance by making annotated data available in a standard format. The current workshop makes a substantial contribution towards that goal, as it includes contributions to Arabic, Hindi, Old Egyptian, Vedic, Northern Kurdish, Slovene, Dutch, Bavarian, South Asian languages, Turkic languages, Gujarati, Saraiki, Swedish, and more. Another important theme for research on mwe's has been the question

¹https://siglex.org/
²https://unidive.lisn.upsaclay.fr
of to what extent Large Language Models deal adequately with the idiomatic meaning of multi-word expressions. This workshop also includes several contributions that explicitly deal with this question. Apart from these important and cross-disciplinary themes, there are also contributions on UD addressing such issues as assessing and enhancing the value of UD parsing for applications, improved automatic parsing procedures, and the interface between syntax and morphology. Contributions that are primarily concerned with MWEs address a.o. the role of lexical resources, automatic identification of MWEs, the proper annotation of idiomatic meanings in a corpus with fully structured meaning annotation, annotation in parallel corpora, and cross-linguistically consistent annotation of MWEs with word senses. Some of these themes re-occur in the contributions that address both UD and MWEs, such as the interplay of lexicon and corpus annotations, the annotation of multiword functional categories, the annotation of light verb constructions, and the use of UD and MWEs in the task of stance detection.

Archna Bhatia, Gosse Bouma, A. Seza Doğruöz, Kilian Evang, Marcos Garcia, Voula Giouli, Lifeng Han, Joakim Nivre, Alexandre Rademacher.

Acknowledgements

MWE-UD Workshop 2024 has been co-organised and funded by The Special Interest Group for the Lexicon of the Association for Computational Linguistics (ACL-SIGLEX) and the CA21167 COST Action UniDive, supported by COST (European Cooperation in Science and Technology). ACL-SIGLEX funded two (2) participants, while UniDive provided funds for the travel and stay of 29 participants.

References


Organizing Committee

Workshop Organizers
Archna Bhatia, Institute for Human and Machine Cognition, USA
Gosse Bouma, Groningen University, Netherlands
A. Seza Doğruöz, Ghent University, Belgium
Kilian Evang, Heinrich Heine University Düsseldorf, Deutschland
Marcos Garcia, University of Santiago de Compostela, Galiza, Spain
Voula Giouli, Institute for Language & Speech Processing, ATHENA RC, Greece
Lifeng Han, University of Manchester, United Kingdom
Joakim Nivre, Uppsala University and Research Institutes of Sweden, Sweden
Alexandre Rademaker, IBM Research, Brazil

Program Committee
Verginica Barbu Mititelu, Cherifa Ben Kehlil, Philippe Blache, Francis Bond, Claire Bonial, Julia Bonn, Tiberiu Boros, Marie Candito, Giuseppe G. A. Celano, Kenneth Church, Çağrı Çöltekin, Mathieu Constant, Monika Czerepowicka, Daniel Dakota, Miryam de Lhoneux, Marie-Catherine de Marneffe, Valeria de Paiva, Gaël Dias, Kaja Dobrovoljc, Rafael Ehren, Gülsen Eryiğit, Meghdad Farahmand, Christiane Fellbaum, Jennifer Foster, Aggeliki Fotopoulou, Stefan Th. Gries, Bruno Guillaume, Tunga Gungor, Eleonora Guzzi, Laura Kallmeyer, Cvetana Krstev, Timm Lichte, Irina Lobzhanidze, Teresa Lynn, Stella Markantoniou, John P. McCrae, Nurit Melnik, Johanna Monti, Dmitry Nikolaev, Jan Odijk, Petya Osenova, Yannick Parmentier, Agnieszka Patejuk, Pavel Pecina, Ted Pedersen, Prokopis Prokopidis, Manfred Sailer, Tanja Samardžić, Agata Savary, Nathan Schneider, Sabine Schulte im Walde, Sebastian Schuster, Matthew Shardlow, Joaquim Silva, Maria Simi, Ranka Stanković, Ivelina Stoyanova, Stan Szpakowicz, Shiva Taslimipoor, Beata Trawinski, Ashwini Vaidya, Marion Di Marco, Amir Zeldes, Daniel Zeman

Keynote Speakers
Natalia Levshina, Radboud University
Harish Tayyar Madabushi, University of Bath
Organized by

SIGLEX
ACL Special Interest Group

UniDive

Funded by
the European Union
# Table of Contents

Every Time We Hire an LLM, the Reasoning Performance of the Linguists Goes Up  
Harish Tayyar Madabushi .......................................................... 1

Using Universal Dependencies for testing hypotheses about communicative efficiency  
Natalia Levshina ........................................................................... 2

Automatic Manipulation of Training Corpora to Make Parsers Accept Real-world Text  
Hiroshi Kanayama, Ran Iwamoto, Masayasu Muraoka, Takuya Ohko and Kohtaroh Miyamoto  
4

Assessing BERT’s sensitivity to idiomaticity  
Li Liu and Francois Lareau ............................................................. 14

Identification and Annotation of Body Part Multiword Expressions in Old Egyptian  
Roberto Díaz Hernández .................................................................. 24

Fitting Fixed Expressions into the UD Mould: Swedish as a Use Case  
Lars Ahrenberg ............................................................................ 33

Synthetic-Error Augmented Parsing of Swedish as a Second Language: Experiments with Word Order  
Arianna Mascioliini, Emilie Francis and Maria Irena Szawerna .................. 43

The Vedic Compound Dataset  
Sven Sellmer and Oliver Hellwig ..................................................... 50

A Universal Dependencies Treebank for Gujarati  
Mayank Jobanputra, Maitrey Mehta and Çağrı Çöltekin .......................... 56

Overcoming Early Saturation on Low-Resource Languages in Multilingual Dependency Parsing  
Jiannan Mao, Chenchen Ding, Hour Kaing, Hideki Tanaka, Masao Utiyama and Tadahiro Matsumoto .................................................. 63

Part-of-Speech Tagging for Northern Kurdish  
Peshmerge Morad, Sina Ahmadi and Lorenzo Gatti .............................. 70

Diachronic Analysis of Multi-word Expression Functional Categories in Scientific English  
Diego Alves, Stefania Degaetano-Ortlieb, Elena Schmidt and Elke Teich ............. 81

Lexicons Gain the Upper Hand in Arabic MWE Identification  
Najet Hadj Mohamed, Agata Savary, Cherifa Ben Khelil, Jean-Yves Antoine, Iskandar keskes and Lamia Hadrich-Belguith ........................................ 88

Revisiting VMWEs in Hindi: Annotating Layers of Predication  
Kanishka Jain and Ashwini Vaidya .................................................. 98

Towards the semantic annotation of SR-ELEXIS corpus: Insights into Multiword Expressions and Named Entities  
Cvetana Krstev, Ranka Stanković, Aleksandra M. Marković and Teodora Sofija Mihajlov ... 106
<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>To Leave No Stone Unturned: Annotating Verbal Idioms in the Parallel Meaning Bank</td>
<td>115</td>
</tr>
<tr>
<td>Rafael Ehren, Kilian Evang and Laura Kallmeyer</td>
<td></td>
</tr>
<tr>
<td>Universal Feature-based Morphological Trees</td>
<td>125</td>
</tr>
<tr>
<td>Federica Gamba, Abishek Stephen and Zdeněk Žabokrtský</td>
<td></td>
</tr>
<tr>
<td>Combining Grammatical and Relational Approaches. A Hybrid Method for the Identification of Candidate Collocations from Corpora</td>
<td>138</td>
</tr>
<tr>
<td>Damiano Perri, Irene Fioravanti, Osvaldo Gervasi and Stefania Spina</td>
<td></td>
</tr>
<tr>
<td>Multiword Expressions between the Corpus and the Lexicon: Universality, Idiosyncrasy, and the Lexicon-Corpus Interface</td>
<td>147</td>
</tr>
<tr>
<td>Annotation of Multiword Expressions in the SUK 1.0 Training Corpus of Slovene: Lessons Learned and Future Steps</td>
<td>154</td>
</tr>
<tr>
<td>Jaka Čibej, Polona Gantar and Mija Bon</td>
<td></td>
</tr>
<tr>
<td>Light Verb Constructions in Universal Dependencies for South Asian Languages</td>
<td>163</td>
</tr>
<tr>
<td>Abishek Stephen and Daniel Zeman</td>
<td></td>
</tr>
<tr>
<td>Sign of the Times: Evaluating the use of Large Language Models for Idiomaticity Detection</td>
<td>178</td>
</tr>
<tr>
<td>Dylan Phelps, Thomas M. R. Pickard, Maggie Mi, Edward Gow-Smith and Aline Villavicencio</td>
<td></td>
</tr>
<tr>
<td>Universal Dependencies for Saraiki</td>
<td>188</td>
</tr>
<tr>
<td>Meesum Alam, Francis Tyers, Emily Hanink and Sandra Kübler</td>
<td></td>
</tr>
<tr>
<td>Domain-Weighted Batch Sampling for Neural Dependency Parsing</td>
<td>198</td>
</tr>
<tr>
<td>Jacob Striebel, Daniel Dakota and Sandra Kübler</td>
<td></td>
</tr>
<tr>
<td>Strategies for the Annotation of Pronominalised Locatives in Turkic Universal Dependency Tree-banks</td>
<td>207</td>
</tr>
<tr>
<td>Jonathan Washington, Çağrı Çöltekin, Furkan Akkurt, Bermet Chontaeva, Soudabeh Eslami, Gulnura Jumalieva, Aida Kasieva, Aslı Kuzgun, Büşra Marşan and Chihiro Taguchi</td>
<td></td>
</tr>
<tr>
<td>BERT-based Idiom Identification using Language Translation and Word Cohesion</td>
<td>220</td>
</tr>
<tr>
<td>Arnav Yayavaram, Siddharth Yayavaram, Prajna Devi Upadhyay and Apurba Das</td>
<td></td>
</tr>
<tr>
<td>Ad Hoc Compounds for Stance Detection</td>
<td>231</td>
</tr>
<tr>
<td>Qi Yu, Fabian Schlotterbeck, Henning Wang, Naomi Reichmann, Britta Stolterfoht, Regine Eckardt and Miriam Butt</td>
<td></td>
</tr>
</tbody>
</table>
Workshop Program

09:00–09:05 Welcome

09:05–09:50 Keynote Speaker 1

Every Time We Hire an LLM, the Reasoning Performance of the Linguists Goes Up
Harish Tayyar Madabushi

09:50–10:30 Session 1: Oral presentations

09:50–10:10 Assessing BERT’s sensitivity to idiomaticity
Li Liu and Francois Lareau

10:10–10:30 Sign of the Times: Evaluating the use of Large Language Models for Idiomaticity Detection
Dylan Phelps, Thomas M. R. Pickard, Maggie Mi, Edward Gow-Smith and Aline Villavicencio

10:30–11:00 Coffee Break

11:00–12:00 Session 2: Poster presentations

12:00–13:00 Session 3: Oral presentations

12:00–12:20 Universal Feature-based Morphological Trees
Federica Gamba, Abishek Stephen and Zdeněk Žabokrtský

12:20–12:40 Light Verb Constructions in Universal Dependencies for South Asian Languages
Abishek Stephen and Daniel Zeman

12:40–13:00 Strategies for the Annotation of Pronominalised Locatives in Turkic Universal Dependency Treebanks
Jonathan Washington, Çağrı Çöltekin, Furkan Akkurt, Bermet Chontaeva, Soudabeh Eslami, Gulnura Jumalieva, Aida Kasieva, Aslı Kuzgun, Büşra Marşan and Chihiro Taguchi

13:00–14:00 Lunch
14:00–14:45  Keynote Speaker 2
Using Universal Dependencies for testing hypotheses about communicative efficiency
Natalia Levshina

14:45–15:00  Booster Session: Remote presentations

15:00–16:00  Session 4: Oral presentations
15:00–15:20  To Leave No Stone Unturned: Annotating Verbal Idioms in the Parallel Meaning Bank
Rafael Ehren, Kilian Evang and Laura Kallmeyer
15:20–15:40  Annotation of Multiword Expressions in the SUK 1.0 Training Corpus of Slovene: Lessons Learned and Future Steps
Jaka Čibej, Polona Gantar and Mija Bon
15:40–16:00  Ad Hoc Compounds for Stance Detection
Qi Yu, Fabian Schlotterbeck, Henning Wang, Naomi Reichmann, Britta Stolterfoht, Regine Eckardt and Miriam Butt

16:00–16:30  Coffee Break

16:30–17:20  Session 5: Poster presentations

17:20–17:50  Community Discussion

17:50–18:00  Closing and Awards
Posters and Remote Presentations

Automatic Manipulation of Training Corpora to Make Parsers Accept Real-world Text
Hiroshi Kanayama, Ran Iwamoto, Masayasu Muraoka, Takuya Ohko and Kohtaroh Miyamoto

Identification and Annotation of Body Part Multiword Expressions in Old Egyptian
Roberto Díaz Hernández

Fitting Fixed Expressions into the UD Mould: Swedish as a Use Case
Lars Ahrenberg

Synthetic-Error Augmented Parsing of Swedish as a Second Language: Experiments with Word Order
Arianna Mascioli, Emilie Francis and Maria Irena Szawerna

The Vedic Compound Dataset
Sven Sellmer and Oliver Hellwig

A Universal Dependencies Treebank for Gujarati
Mayank Jobanputra, Maitrey Mehta and Çağrı Çöltekin

Overcoming Early Saturation on Low-Resource Languages in Multilingual Dependency Parsing
Jiannan Mao, Chenchen Ding, Hour Kaing, Hideki Tanaka, Masao Utiyama and Tadahiro Matsumoto

Part-of-Speech Tagging for Northern Kurdish
Peshmerge Morad, Sina Ahmadi and Lorenzo Gatti

Diachronic Analysis of Multi-word Expression Functional Categories in Scientific English
Diego Alves, Stefania Degaetano-Ortlieb, Elena Schmidt and Elke Teich

Lexicons Gain the Upper Hand in Arabic MWE Identification
Najet Hadj Mohamed, Agata Savary, Cherifa Ben Khelil, Jean-Yves Antoine, Iskandar keskes and Lamia Hadrich-Belguith

Revisiting VMWEs in Hindi: Annotating Layers of Predication
Kanishka Jain and Ashwini Vaidya

Towards the semantic annotation of SR-ELEXIS corpus: Insights into Multiword Expressions and Named Entities
Cvetana Krstev, Ranka Stanković, Aleksandra M. Marković and Teodora Sofija Mihajlov

Combining Grammatical and Relational Approaches. A Hybrid Method for the Identification of Candidate Collocations from Corpora
Damiano Perri, Irene Fioravanti, Osvaldo Gervasi and Stefania Spina

Multiword Expressions between the Corpus and the Lexicon: Universality, Idiosyncrasy, and the Lexicon-Corpus Interface
Verginica Barbu Mittelnu, Voula Giouli, Kilian Evang, Daniel Zeman, Petya Osenova, Carole Tiberius, Simon Krek, Stella Markantonatou, Ivelina Stoyanova, Ranka Stanković and Christian Chiarcos
Universal Dependencies for Saraiki
Meesum Alam, Francis Tyers, Emily Hanink and Sandra Kübler

Domain-Weighted Batch Sampling for Neural Dependency Parsing
Jacob Striebel, Daniel Dakota and Sandra Kübler

BERT-based Idiom Identification using Language Translation and Word Cohesion
Arnav Yayavaram, Siddharth Yayavaram, Prajna Devi Upadhyay and Apurba Das

MultiMWE: Building a Multi-lingual Multi-Word Expression (MWE) Parallel Corpora [non-archival]
Lifeng Han, Gareth Jones and Alan Smeaton

AlphaMWE-Arabic: Arabic Edition of Multilingual Parallel Corpora with Multiword Expression Annotations [non-archival]
Najet Hadj Mohamed, Malak Rassem, Lifeng Han and Goran Nenadic

A demonstration of MWE-Finder and MWE-Annotator [non-archival]
Jan Odijk, Martin Kroon, Tijmen Baarda, Ben Bonfil and Sheean Spoel

MaiBaam: A Multi-Dialectal Bavarian Universal Dependency Treebank [non-archival]
Verena Blaschke, Barbara Kovačić, Siyao Peng, Hinrich Schütze and Barbara Plank

Is Less More? Quality, Quantity and Context in Idiom Processing with Natural Language Models [non-archival]
Agne Knietaite, Adam Allsebrook, Anton Minkov, Adam Tomaszewski, Norbert Slinko, Richard Johnson, Thomas M. R. Pickard and Aline Villavicencio

A Corpus of Persian Sentences Annotated with Verbal Multiword Expressions: Development and Guidelines [non-archival]
Vahide Tajalli, Yaldasadat Yarandi, Mahtab Sarlak, Mehrnoush Shamsfard and Arezoo Haghbin

Annotating Compositionality Scores for Irish Noun Compounds is Hard Work [non-archival]
Abigail Walsh, Teresa Clifford, Emma Daly, Jane Dunne, Brian Davis and Gearóid Ó Cleirícín

Redefining Syntactic and Morphological Tasks for Typologically Diverse Languages [non-archival]
Omer Goldman, Leonie Weissweiler and Reut Tsarfaty

UCxn: Typologically Informed Annotation of Constructions Atop Universal Dependencies [non-archival]
Keynote Speech
Every Time We Hire an LLM, the Reasoning Performance of the Linguists Goes Up
Harish Tayyar Madabushi
University of Bath

Abstract

Pre-Trained Language Models (PLMs), trained on the cloze-like task of masked language modelling, have demonstrated access to a broad range of linguistic information, including both syntax and semantics. Given their access to both syntax and semantics, coupled with their data-driven foundations, which align with usage-based theories, it is valuable and interesting to examine the constructional information they encode. Early work confirmed that these models have access to a substantial amount of constructional information. However, more recent research focusing on the types of constructions PLMs can accurately interpret, and those they find challenging, suggests that an increase in schematicity correlates with a decline in model proficiency. Crucially, schematicity—the extent to which constructional slots are fixed or allow for a range of elements that satisfy a particular semantic role associated with the slot—correlates to the extent of “reasoning” needed to interpret constructions, a task that poses significant challenges for language models. In this talk, I will begin by reviewing the constructional information encoded in both earlier models and more recent large language models. I will explore how these aspects are intertwined with the models’ reasoning abilities and introduce promising new approaches that could integrate theoretical insights from linguistics with practical, data-driven approaches of PLMs.
Keynote Speech

Using Universal Dependencies for testing hypotheses about communicative efficiency

Natalia Levshina
Centre for Language Studies
Radboud University, Nijmegen, The Netherlands

Abstract

There is abundant evidence that language structure and use are influenced by language users’ tendency to be efficient, trying to minimize the cost-to-benefit ratio of communication (e.g., Hawkins, 2004; Gibson et al., 2019; Levshina, 2022). In my talk I will show how data from corpora annotated with Universal Dependencies can be used for testing hypotheses about the role of communicative efficiency in shaping up language structure and use. The hypotheses are as follows:

1. As discussed by typologists (Sapir, 1921; Sinnemäki, 2008), rigid word order can compensate for lack of formal marking of core arguments. The hypothesis is then that there are positive correlations between the entropy of subject and object in a transitive clause in a corpus and the relative frequency of disambiguating case forms or verb forms. These correlations are expected to minimize the articulation effort involved in the use of argument flags or indices.

2. There is a positive correlation between semantic tightness (Hawkins, 1986), operationalized as Mutual Information between lexemes and syntactic roles, and the relative frequency of verb-final clauses in a corpus. Strong associations between lexemes and roles help to avoid the costs of reanalysis in verb-final languages.

3. There is a negative correlation between the relative frequency of verb-final clauses in the clause and the average number of overt core arguments, which helps to save processing costs required for keeping longer dependencies in mind (cf. Ueno & Polinsky, 2009).

These hypotheses will be tested on corpus data annotated with Universal Dependencies, with the help of mixed-effects models with genealogical and geographic information as random effects.

References


Automatic Manipulation of Training Corpora
to Make Parsers Accept Real-world Text

Hiroshi Kanayama, Ran Iwamoto, Masayasu Muraoka, Takuya Ohko, Kohtaroh Miyamoto
IBM Research

{hkana@jp., ran.iwamoto1@, mmuraoka@jp., ohkot@jp., kmiya@jp.}ibm.com

Abstract
This paper discusses how to build a practical syntactic analyzer, and addresses the distributional differences between existing corpora and actual documents in applications. As a case study we focus on noun phrases that are not headed by a main verb and sentences without punctuation at the end, which are rare in a number of Universal Dependencies corpora but frequently appear in the real-world use cases of syntactic parsers. We converted the training corpora so that their distribution is closer to that in realistic inputs, and obtained better scores both in general syntax benchmarking and a sentiment detection task, a typical application of dependency analysis.

Keywords: syntax, parsing, Universal Dependencies

1. Introduction

In text processing applications that handle documents such as user reviews and contract documents, accurate syntax parsing is desired for semantic analysis and information extraction. The emerging generative approach also requires the analysis of given utterances to make systems reliable and explainable, such as in retrieval augmented generation (Lewis et al., 2020), and the language models can be improved by incorporating syntactic knowledge (Iwamoto et al., 2023).

Multilingual corpora in Universal Dependencies (UD) (Nivre et al., 2016, 2020) are easily available, and they are used for training and evaluation of syntactic analysis components including tokenizers, part-of-speech (PoS) taggers, and dependency parsers, such as Stanza (Qi et al., 2020), UDPipe (Straka, 2018), spaCy (Honnibal et al., 2020) and Trankit (Nguyen et al., 2021).

However, we found a gap between the standardized UD corpora and the real-world application scenarios. There are many noun phrases (NPs) in reviews such as hotel ones written by a customer as (1), instead of a formal sentence typically with a finite verb in a root node of the syntax tree such as in (2).

1. A very good hotel close to the park!
2. I think the hotel is very good because it is close to the park.

Another example of noun phrases is a description in a contract document, such as in (3).

3. total cost of the services

These noun phrases can appear in many kinds of text documents as the title of a document or section, items in enumeration, a header line of a table, and so on. In addition, in many cases, such strings do not have a period or other punctuation marks at the end.

When we apply a syntax analyzer trained on the UD corpora to such short noun phrases, we often find very wrong analysis results, as exemplified in the output syntax structures of English (4) and German (5). A “*” mark indicates the errors in PoS tags or dependency relations.

(4) Recapture *VERB of ADP the DET bridge NOUN

(5) sehr ‘very’ ADV schöne ‘nice’ ADJ besteckset ‘cutlery’ PUNCT

In (4), the first word “recapture” (which should be NOUN) was incorrectly tagged as VERB as if...
the input were an interrogative sentence that starts with a verb, and it causes an incorrect dependency relation between “recapture” and “bridge,” which should be nmod rather than obl. In (5), the noun “Besteckset” (“cutlery”) was tagged as PUNCT. The writing is not formal because German nouns should start with a capital letter, but the tagging result PUNCT is apparently incorrect. These are actual results by the Stanza parser that achieved very high scores in the UD parsing shared task (Zeman et al., 2018), and we found other taggers and parsers such as UDpipe and spaCy produced similar errors. These errors have already been recognized in the community and discussed in the GitHub issues of those implementations.

If most of the contents in the training corpora contain finite verbs in the sentence rather than only noun phrases, it is not surprising that the taggers and parsers trained on such corpora tend to produce incorrect results for the noun phrase inputs such as (4) and (5). Also, we can assume that very unusual tagging results such as in (5) are caused by the training corpus where most sentences end with a period (‘.’). Thus, they are problems in the difference between the training corpus and target input to be analyzed.

Figure 1 illustrates the problem that this paper addresses. Normally, the syntax analyzers are trained and evaluated on the UD corpora, but the real-world input documents have different distributions from those of the UD corpora, and the models trained on the UD corpora cause catastrophic errors in applications. Thus, we manipulate the UD corpora to alter distributions in terms of noun phrases and sentence-end punctuation. Although it is impossible to know the general distribution in the real-world inputs, we can make the parser more robust by manipulating the training corpus to reduce the bias in the current UD data.

The contributions of this paper are: (1) to handle the issue regarding noun phrases in addition to punctuation, (2) to provide an algorithm to manipulate training corpora without any manual annotation work, (3) to propose methods to evaluate this work from multiple viewpoints, including the automatic generation of an evaluation data set of noun phrases, and (4) to show the effects of the corpus manipulation in four languages.

Section 2 reviews the related work regarding UD and existing discussions on punctuation and noun phrases. In Section 3, we define the terms used in this paper. Section 4 shows the statistics in different corpora. In Section 5, we propose the algorithm to manipulate training corpora so that the parser can accept real-world inputs, and the effect is shown in Section 6.

2. Related Work

Universal Dependencies (UD) (Nivre et al., 2016) is a worldwide project to provide multilingual syntactic corpora. As of November 2023, 259 treebanks in 148 languages have been released. For all languages, the syntax is represented by dependency trees with 17 PoS tags and 37 dependency labels commonly used for all languages, and each treebank can have language specific extensions. The resources and documentations are available online and incrementally updated. A major shared task of multilingual parsing (Zeman et al., 2018) was held, and a result, UD treebanks is now a de facto standard of multilingual research and many tokenizers and parsers have been trained on them, including a multilingual single parser (Kondratyuk and Straka, 2019).

English Web Treebank (EWT) (Silveira et al., 2014) is one of the most commonly-used treebanks in UD. Originally, it was designed to cover more informal text, such as e-mail and review documents, which was not included in the treebanks of the Wall Street Journal (WSJ). After the emergence of Universal Dependencies, EWT was converted to a UD-style annotation. Thus, EWT contains noisy sentences with typos and abbreviations, and even sentence splitting is tricky (Udagawa et al., 2023), but their work showed that the parsers trained using EWT had a better capability to parse such informal text than the model trained only with WSJ. Due to this historical reason, the EWT corpus functions as an outlier in the experiments in this paper.

The effects of punctuation in a dependency parser have been discussed by Seggaard et al. (2018). They pointed out that dependency parsers, especially neural implementations, are highly sensitive to punctuation in training corpora, and training parsers without punctuation makes the models better. In this paper, we extend the discussion from punctuation to noun phrases, which are more critical in real-world applications. Nivre and Fang (2017) pointed out that punctuation highly affects the benchmarking scores in a number of corpora even if it is not significant in the actual analysis.

The analysis of noun phrase structures have been discussed (Nakov and Hearst, 2005; Vadas and Curran, 2011) but parsing confusion between noun phrases and finite sentences has been less studied. There was a report that a parser specific to noun phrases improved machine translation quality even if the LAS (labeled attachment score) of dependency parsing was not significantly changed (Green, 2011).

---


2. https://universaldependencies.org/
Corpus synthesis is a powerful method to adapt to specific tasks to enhance a production parser (El-Kurdi et al., 2020) and to broaden the supported languages (Tiedemann and Agic, 2016; Dehouck and Gómez-Rodríguez, 2020) and domains (Li et al., 2019; Jia and Liang, 2016). This paper shares a similar motivation with them but we propose a method to extend training corpora with linguistic knowledge to address specific issues without adding new data sources.

3. Terminology

In this section terms used in this paper are defined.

Unit A text string that is regarded as a single “sentence” in corpora. A unit is also given as an input to a PoS tagger, dependency parser, and their downstream applications, which may be a result of sentence splitting. In this paper we do not call it a “sentence” to distinguish it from the sentence defined as follows. All of (1), (2), (3), (4) and (5) in Section 1 can be a unit.

Sentence A unit that is governed by a finite verb, including nominal predicate sentences associated with a copula. A sentence corresponds to a non-terminal symbol ‘S’ in the phrase structure grammar, though this paper does not discuss its definition from a linguistic viewpoint. Example (2) in Section 1 is a sentence.

Noun Phrase (NP) A syntactic tree or subtree whose head word is a noun or a proper noun, namely, its universal PoS (UPOS) tag is either NOUN or PROPN. An NP also does not have a child node of a copula (where dependency relation label is cop). Note that in the content-head structure of UD, the head word of a sentence “She is a teacher.” is “teacher” rather than “is” (be-verb).

Noun Phrase Unit (NPU) A unit that forms an NP. Examples include (1), (3), (4) and (5) in Section 1.

Ending punctuation (end-punct) A punctuation mark at the end of a sentence or unit. Here, a punctuation mark is a word that is tagged as PUNCT in the UD corpora. In this paper, we only focus on a period (‘.’), an exclamation mark (‘!’) and a question mark (‘?’), which are used in European languages, and discard other PUNCT words like parentheses and quotation marks.

Punctuation Omitted Unit (POU) A unit without ending punctuation. Examples include (3), (4) and (5) in Section 1.

4. Observation of Corpora

To determine how many noun phrase units (NPUs) and punctuation omitted units (POUs) existed in the training corpora and expected input documents, we observed two types of corpora in four languages. One is Universal Dependencies (UD), which is used for the training of various syntax analyzers. Here, we observe the development portion in UD Version 2.13. The other is the review data used for the evaluation of sentiment analysis. We randomly selected 100 sentences of each language version of review data from the SemEval shared task data for aspect-based sentiment analysis (Pontiki et al., 2016) for English, French and Spanish, and Amazon reviews used in another shared task (Ruppenhofer et al., 2014) for German.

Table 1 shows the ratio of the NPUs and POUs in the UD corpora and review documents. Particularly in the UD corpora of French and Spanish, the ratios of NPUs and POUs are very low, that is, the UD corpora tend to consist of formal sentences with finite verbs having ending punctuation marks as their units. The UD English corpus has a relatively higher ratio of NPUs and POUs because there are many informally written documents in EWT corpus as mentioned in Section 2.

The review corpora tend to have many NPUs and POUs, except for the English SemEval data set. There are fewer POUs in SemEval data set (particularly the English one) as expected, that is, most of the units end with a period. The SemEval corpora are supposed to be controlled to have periods for the purpose of extraction of positive or negative expressions with aspects.

As we previously observed, the distribution of syntactic characteristics is very diverse, and those trends highly depend on the formality or cleanliness of the contents of the data set and languages. This shows that it is quite difficult to expect a fixed corpus such as those of UD to represent the distribution of real-world documents that are given to the applications of syntactic analyzers.

5. Corpus Extension

In this section, we propose a method to extend the training corpora for syntactic analyzers, to address the problem of differences in characteristics of corpora discussed in Section 4. Our goal is to build
<table>
<thead>
<tr>
<th>language</th>
<th>UD review</th>
<th>NPU ratio (%)</th>
<th>POU ratio (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>EWT SemEval</td>
<td>23.0</td>
<td>3.0</td>
</tr>
<tr>
<td>French</td>
<td>GSD SemEval</td>
<td>3.2</td>
<td>36.0</td>
</tr>
<tr>
<td>German</td>
<td>GSD Gestalt</td>
<td>6.1</td>
<td>28.0</td>
</tr>
<tr>
<td>Spanish</td>
<td>AnCora SemEval</td>
<td>4.5</td>
<td>25.0</td>
</tr>
</tbody>
</table>

Table 1: Ratios of noun phrase units (NPUs) and punctuation omitted units (POUs) in UD and review corpora of four languages.

![Figure 2: Extraction example of an NP (indicated as a box) from a sentence.](image)

Figure 2: Extraction example of an NP (indicated as a box) from a sentence.

models useful in real applications by reducing the bias in the training model to the UD corpora as illustrated in Figure 3.

### 5.1. Removal of punctuation

We assume that the incorrectly assigned PUNCT tag to a noun in (5) is caused by the PoS tagging model trained on the corpus where most of last word is tagged as PUNCT. A desirable model is robust to the existence of punctuation, that is, the result should be consistent with or without an end-punct.

A straightforward solution to the problem of the bias to the training corpora is to reduce end-punct at a certain ratio $p$, in other words, to add POUs, and then to retrain models. Most end-puncts do not have any child nodes in the dependency tree, and thus, it is quite easy to remove an end-punct from a unit, maintaining the validity of the tree$^5$.

---

5As an exception, the UD_English-EWT training corpus contained a unit in which a conjunction “and” attached to a period at the end of the sentence. In such a case, we did not apply the modification of punctuation removal.

---

### 5.2. Addition of noun phrases

In addition to sentences headed by a finite verb, a training corpus should contain noun phrases as units, to handle similar inputs in real applications. To make such a corpus, we add NPUs by extracting noun phrase subtrees from the original corpus in the following manner.

- Identify nouns (a word tagged as NOUN or PROPN) in the original dependency trees.
- Find noun phrases, selecting nouns whose subtree headed by the noun consists of more than three sequential words$^6$.
- Exclude a preposition and punctuation that should not be a part of a noun phrase. This treatment is needed because the syntactic structure in UD is designed in a content-head manner, and thus, a number of function words are included in a subtree of noun phrases. Functional words that attach to the head of the noun phrase with a dependency label case or punct are removed from the noun phrase. In the case shown in Figure 2, the preposition “in” is excluded from the noun phrase headed by “city” because it attaches to “city” with case relation.
- Pool the noun phrases extracted in this way, and randomly select a number of them in a given ratio (n) to add them to the training corpus, keeping all of the original units in the corpus.

---

6If the children or descendant nodes have a gap due to non-projectivity, such noun phrases are ignored.
In Section 6, we will show the effects of corpora conversion by changing the ratio of punctuation removal and noun phrase addition.

6. Experiments

6.1. Data for evaluation

We will evaluate the syntax analyzer trained on the extended corpora in three ways using three different data sets in four languages: English, French, German and Spanish.

6.1.1. Noun phrase

We evaluate the robustness of the syntax analyzers to the input strings of noun phrases, as a unit test of our approach. For this purpose, we automatically generated the test set of noun phrases in the following procedure.

- Obtain section titles of Wikipedia articles of four languages
- Extract section titles that consist of three or more words
- Exclude those that contain special characters such as numbers, symbols, quotation and punctuation marks
- Exclude those containing non-canonical upper/lower cases (e.g. “RNAb”, “AIESEC”)
- Exclude those that were judged as different languages from that of Wikipedia
- For English, French and Spanish, change the initial character of each word into lower case
- Remove duplication
- Diversify the first word so that there are no more than three entries that share the first word. This is to reduce frequent patterns such as “List of XX”
- Randomly select 1,500 entities for each language

This process almost perfectly extracts noun phrases in each language, and by definition, the last word is not punctuation. Table 2 shows examples in four languages.

In the experiments in this section, we will apply PoS taggers and dependency parsers to these data to calculate the following two scores:

- **Wrong punctuation** The number of cases where the last word is tagged as PUNCT or its dependency label is punct. A lower number is better.
- **NP detection** The ratio of the dependency trees of which the root node is tagged as NOUN. A higher ratio is better.

6.1.2. Universal Dependencies

We use the UD corpora for the intrinsic evaluation of dependency parsers. The F1 score of LAS is used as a representative evaluation metric. In our experiments, we extend the train and dev portions of the UD corpora with the methods presented in Section 5, and the test portion for evaluation is not changed. This means the distributions of units are different between the test and training corpora. As a result, the LAS score on the UD test corpus will be theoretically decreased, and thus, minimizing the downgrade of the LAS score indicates the success of our approach.

6.1.3. Sentiment detection

We also conduct an extrinsic evaluation using multilingual sentiment detection (Kanayama and Iwamoto, 2020; Iwamoto et al., 2021) as an application of dependency parsing. For the evaluation, we used sentiment analysis data sets that were observed in Section 4. Those data sets for four languages were derived from shared tasks (Pontiki et al., 2016; Ruppenhofer et al., 2014) and all of them are customer’s review data in a domain per language (restaurant for English, French and Spanish, cutlery for German). Each of them contains 500 units, and the annotations were simplified so that each unit has a unit-level polarity flag (either positive or negative) as shown in Table 3.

Similarly to the previous work on multilingual sentiment detection (Kanayama and Iwamoto, 2020), we calculated precision and recall as metrics. Precision depends on the quality of the sentiment lexicon and handling of syntax phenomena such as negation. Recall is related to the coverage of the sentiment lexicon and accuracy in detection of the root node in dependency analysis. The experiments in this paper have few factors that change the precision of sentiment detection, and thus, we focus on recall as it is affected by syntactic structures related to noun phrases.

6.2. Parser retraining

We applied the two kinds of conversion described in Section 5 to the training portions of the UD corpora in four languages (German-GSD, French-GSD, Spanish-AnCora and English-EWT), and retrained models of the Stanza version 1.1.1 (Qi et al., 2020).
Table 2: Examples of noun phrases in the Wikipedia section title data set.

<table>
<thead>
<tr>
<th>Language</th>
<th>Example</th>
<th>Polarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>This has got to be one of the most overrated restaurants in Brooklyn.</td>
<td>Negative</td>
</tr>
<tr>
<td>French</td>
<td>Aucune commande de dessert n’a été prise après une demie heure d’attente à la fin de le plat.</td>
<td>Negative</td>
</tr>
<tr>
<td>German</td>
<td>Die Griffe sind schön geformt, die Messer liegen angenehm in der Hand und sind scharf.</td>
<td>Positive</td>
</tr>
<tr>
<td>Spanish</td>
<td>El servicio es muy bueno y la calidad de la comida al mismo nivel.</td>
<td>Positive</td>
</tr>
</tbody>
</table>

Table 3: Examples of sentiment polarity data. The second example of each language is a noun phrase.

<table>
<thead>
<tr>
<th>Language</th>
<th>Example</th>
<th>Polarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>Best Pastrami I ever had and great portion without being ridiculous.</td>
<td>Positive</td>
</tr>
<tr>
<td>French</td>
<td>('No dessert order was taken after half an hour wait at the end of the dish.')</td>
<td>Negative</td>
</tr>
<tr>
<td>German</td>
<td>('The handles are beautifully shaped, the knives are comfortable to hold and sharp.')</td>
<td>Positive</td>
</tr>
<tr>
<td>Spanish</td>
<td>('A restaurant which I don’t want to come back to')</td>
<td>Negative</td>
</tr>
</tbody>
</table>

2020) with the extended training corpora. For all languages, we retrained PoS tagging models (pos) and dependency parsing models (depparse) with maximum iteration of 5,000 times\(^8\), and other models for tokenization (tokenize), multi-word tokens (mwt) and lemmatization (lemma) were not changed from the default ones.

We tested various ratios for the removal of punctuation \((p)\) and addition of noun phrases \((n)\). \(p = 0, n = 0\) means the original UD corpus as it is, and thus, it is the baseline for each language. We evaluated two scores using the noun phrase data sets described in Section 6.1: number of incorrect punctuation and ratio of NP detection. We also evaluated the LAS score using the UD test corpus, and the recall of sentiment detection using the review corpus.

Stanza’s retraining process is randomized and the resultant models are not deterministic, and thus, we conducted 10-times retraining on the baseline settings \((p = 0 and n = 0)\) to report the average and standard deviation of each score.

6.3. Results

Tables 4, 5, 6 and 7 show the results of all metrics for German, French, Spanish, and English, respectively. The top row \((p = 0, n = 0)\) shows the baseline scores with the model trained on the original corpus. The next section (remove punct) shows the effects of reducing end-punct by \(p\), and the last section (add NP) reports the scores by adding NPs to the training corpus varying \(n\), including combination of both modification with \(p\) and \(n\).

In the baseline models of German, French and Spanish, there were 3.2 to 4.2% of catastrophic punctuation errors. Removing end-puncts effectively reduced such errors, even with a small ratio of \(p\). By setting \(p = 20\%\), such errors were completely avoided in the four languages.

However, just removing punctuation did not improve the scores of other metrics, although there are a number of settings that improved NP detection in French and Spanish. Also, the changes of LAS and sentiment recall were marginal. The large decrease of LAS scores for \(p = 100\%\) (3 points decrease in German and French) is as expected because \(p = 100\%\) means all end-puncts were removed from the training corpora, and the punctuation marks that remain in the test corpora are difficult to handle with the model trained by the training corpora without any end-puncts.

\(^8\)Setting \(\text{max\_steps}=5000\), one tenth of the default setting. This is to reduce training time with small sacrifice of accuracy.
Table 4: Results of syntax analysis and sentiment detection in German using the models trained on the extended UD corpora with \( p \) punctuation removal and \( n \) noun phrase addition. In percent except for the number of incorrect punctuation marks. The top row (\( p = 0 \), \( n = 0 \)) shows the baseline scores with the original corpus, with the average score of 10 trials and standard deviation. In other rows, a bold number with a + mark indicates that the score is significantly better than the baseline with a difference higher than the standard deviation. A − mark indicates the score is worse against the baseline.

<table>
<thead>
<tr>
<th>( p )</th>
<th>( n )</th>
<th>Section title</th>
<th>UD Sentiment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Wrong punct (↓)</td>
<td>NP detection (↑)</td>
</tr>
<tr>
<td>baseline</td>
<td>0</td>
<td>0</td>
<td>3.2 ±1.75</td>
</tr>
<tr>
<td>remove punct</td>
<td>10</td>
<td>0</td>
<td>0 +</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0</td>
<td>0 +</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0</td>
<td>0 +</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0</td>
<td>0 +</td>
</tr>
<tr>
<td>add NP</td>
<td>0</td>
<td>10</td>
<td>0 +</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>10</td>
<td>0 +</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>10</td>
<td>0 +</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>20</td>
<td>0 +</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>50</td>
<td>0 +</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>50</td>
<td>0 +</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>100</td>
<td>0 +</td>
</tr>
</tbody>
</table>

Table 5: French results. See the caption of Table 4 for details.

<table>
<thead>
<tr>
<th>( p )</th>
<th>( n )</th>
<th>Section title</th>
<th>UD Sentiment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Wrong punct (↓)</td>
<td>NP detection (↑)</td>
</tr>
<tr>
<td>baseline</td>
<td>0</td>
<td>0</td>
<td>4.2 ±0.35</td>
</tr>
<tr>
<td>remove punct</td>
<td>10</td>
<td>0</td>
<td>1 +</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0</td>
<td>0 +</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0</td>
<td>0 +</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0</td>
<td>0 +</td>
</tr>
<tr>
<td>add NP</td>
<td>0</td>
<td>10</td>
<td>3 +</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>10</td>
<td>0 +</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>10</td>
<td>0 +</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>20</td>
<td>0 +</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>50</td>
<td>0 +</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>50</td>
<td>0 +</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>100</td>
<td>0 +</td>
</tr>
</tbody>
</table>

Table 6: Spanish results. See the caption of Table 4 for details.

The addition of noun phrases had larger impacts in all metrics. When the noun phrases were added (\( p = 0 \), \( n > 0 \)), NP detection ratio was improved in all four languages, and it was consistently increased with \( n \). Considering that the noun phrases extracted from the UD corpora and those in the Wikipedia section data are independent, we can say that the addition of noun phrases to the training corpora has a positive impact on the analysis of noun phrase inputs generally. There were cases...
that were not detected as nouns even for \( n = 100\% \), but a number of remaining errors were due to automatic noun phrase extraction from Wikipedia section titles.

The addition of NPs reduced the punctuation errors as well, even without explicit removal of punctuation (e.g. \( p = 0 \) cases). This is because the noun phrases added to the corpus did not have end-puncts, and thus, it helped models avoid bias to corpora consisting of POUs.

Although these treatments for noun phrase inputs obviously made positive impacts to the Wikipedia section title data, there is a potential risk of damage to the existing benchmarking. In the results of the LAS score in the UD test corpora, the decrease in general dependency parsing performance was observed in a number of cases with high ratios of \( p \) and \( n \), but in most of cases, LAS scores were equal to or better than the baseline settings.

Because our motivation in this work is to build a robust parser for real-world applications, an extrinsic evaluation should be a main focus. In French, German and Spanish, recall scores in sentiment detection were increased with a moderate ratio of end-punct removal or NP addition, even though the optimal ratio of \( p \) and \( n \) varies by languages.

In English, the sentiment detection was not improved from the baseline. These results can be supported by the observation in Section 4: UD_English-EWT data contains NPUcs and POUs with higher ratios compared to other corpora, and the English version of SemEval data was highly controlled with formal sentences without NPUcs and POUs, and thus, our approach to corpus expansion did not work for this settings, but it is notable that negative impacts were limited as well.

### 7. Conclusion

This paper presented methods to make robust PoS taggers and dependency parsers to inputs for real-world applications by reducing the discrepancy of the ratios of noun phrases and punctuation omitted units between the training corpora and expected input documents. In addition to the removal of punctuation, which has been attempted to build more consistent models, we added noun phrases to the training corpus by automatically extracting noun phrases from existing annotations using syntactic operations. The experimental results showed that retraining on the extended training corpora made positive impacts on all three experiments simultaneously; a unit test for noun phrases, intrinsic evaluation of the dependency parser, and extrinsic evaluation of it on sentiment detection. The selection of the optimal values in the corpus expansion (ratios of punctuation removal and noun phrase addition) is our future work.

In this paper we handled multiple European languages where the definition of noun phrases and punctuation is relatively easy. In other languages, the structure of noun phrases is more diverse and complicated, and thus, more linguistic discussion and empirical studies will be needed. We applied the proposed technique to the UD corpora, but this can be integrated with the corpus augmented method using raw corpora (El-Kurdi et al., 2020), so that more applicable syntax analyzers can be developed.

The results of our experiments suggest that the current UD corpora are not perfect to train models for practical syntactic analyzers, and that it is important to know the characteristics of corpora and input documents to analyze, and to adjust the corpora to generate better models not just for the benchmarking on UD, but also for the practical use cases.

<table>
<thead>
<tr>
<th>( p )</th>
<th>( n )</th>
<th>Wrong punct (↓)</th>
<th>NP detection (↑)</th>
<th>UD LAS (↑)</th>
<th>Sentiment Recall (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>0</td>
<td>0</td>
<td>0.7 ±0.67</td>
<td>91.6 ±0.63</td>
<td>83.84 ±0.14</td>
</tr>
<tr>
<td>remove punct</td>
<td>10</td>
<td>0</td>
<td>0 +</td>
<td>91.7</td>
<td>83.81</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0</td>
<td>0 +</td>
<td>91.1</td>
<td>84.06 +</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0</td>
<td>2</td>
<td>91.4</td>
<td>84.03 +</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0</td>
<td>0 +</td>
<td>90.1 -</td>
<td>83.46 -</td>
</tr>
<tr>
<td>add NP</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>93.9 +</td>
<td>84.09 +</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>10</td>
<td>0 +</td>
<td>94.2 +</td>
<td>83.71 +</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>10</td>
<td>1</td>
<td>93.7 +</td>
<td>83.96 +</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>20</td>
<td>0 +</td>
<td>94.6 +</td>
<td>83.91 +</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>50</td>
<td>0 +</td>
<td>95.3 +</td>
<td>83.88 +</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>50</td>
<td>0 +</td>
<td>95.4 +</td>
<td>83.75 +</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>100</td>
<td>0 +</td>
<td>95.3 +</td>
<td>84.00 +</td>
</tr>
</tbody>
</table>

Table 7: English results. See the caption of Table 4 for details.
8. Bibliographical References


Assessing BERT’s sensitivity to idiomaticity

Li Liu, François Lareau
OLST, Université de Montréal
Montréal, Canada
{li.liu.2, francois.lareau}@umontreal.ca

Abstract

BERT-like language models have been demonstrated to capture the idiomatic meaning of multiword expressions. Linguists have also shown that idioms have varying degrees of idiomaticity. In this paper, we assess CamemBERT’s sensitivity to the degree of idiomaticity within idioms, as well as the dependency of this sensitivity on part of speech and idiom length. We used a demasking task on tokens from 3,127 idioms and 22,551 tokens corresponding to simple lexemes taken from the French Lexical Network (LN-fr), and observed that CamemBERT performs distinctly on tokens embedded within idioms compared to simple ones. When demasking tokens within idioms, the model is not proficient in discerning their level of idiomaticity. Moreover, regardless of idiomaticity, CamemBERT excels at handling function words. The length of idioms also impacts CamemBERT’s performance to a certain extent. The last two observations partly explain the difference between the model’s performance on idioms versus simple lexemes. We conclude that the model treats idioms differently from simple lexemes, but that it does not capture the difference in compositionality between subclasses of idioms.

Keywords: phraseology, idioms, idiomaticity, multiword expressions (MWEs), language models

1. Introduction

Multiword expressions (MWEs) are characterized by the constrained selection of their components and their partial or complete lack of compositionality (Mel’čuk, 2023). In this paper, we focus on idioms, a prominent category of MWEs known for their non-compositional nature which have long presented a significant challenge for natural language processing (NLP) (Sag et al., 2002; Baldwin and Kim, 2010; Constant et al., 2017).

Idioms cannot be understood simply by the regular combination of the meanings of their components, e.g., *spill the beans* means ‘disclose a secret’, which cannot be obtained from ‘spill’+‘beans’. However, while all idioms violate compositionality, some idioms do include the meaning of some or even all of their components, making them more or less semantically transparent. Hence, compositionality in idioms falls on a continuum. According to the degree of inclusion of the meaning of their components, Mel’čuk (2023) classifies idioms into weak idioms, which include the meaning of all of their components along with some arbitrary meaning, as in (1), semi-idioms, which include the meaning of some but not all of their components along with some arbitrary meaning, as in (2), and strong idioms, which are completely non-compositional, as in (3). This is illustrated below with French idioms.

(1) étoile de mer
star of sea
‘starfish’ = ‘star-shaped marine animal’

(2) fruit de mer
fruit of sea
‘seafood’ = ‘food that comes from the sea’

(3) noyer le poisson
drown the fish
‘obfuscate things’

The contextualized language model BERT (Devlin et al., 2019), pre-trained on extensive linguistic data, has been widely used and has shown exceptional performance across diverse NLP tasks. Given the high degree of conventionality of idioms (Calzolari et al., 2002), there is a natural expectation for BERT to be good at handling them. Indeed, Tan and Jiang (2021) have validated the model’s ability to distinguish between the literal and idiomatic usage of potential idiomatic expressions. Nedumpozhimana and Kelleher (2021) have shown that BERT incorporates information from idioms and their surrounding context to process them. Tian et al. (2023) have demonstrated that BERT-like language models represent idioms differently from their literal counterparts at both sentence and word levels, with words in idioms receiving less attention than words in non-idiomatic contexts. Clearly, BERT has a strong ability at handling idioms. However, one question remains: is BERT sensitive to the degree of idiomaticity of idioms?

Our hypotheses are that:

1. CamemBERT should be better at predicting tokens within idioms as opposed to simple lexemes, because tokens within idioms are more strongly constrained.
2. Tokens within idioms with higher idiomaticity should be more likely to be accurately predicted compared to tokens within idioms with lower idiomaticity.

As far as we know, there has been limited research into this question. The closest research was...
by Garcia et al. (2021b), who conducted a series of probing tasks to examine whether and to what extent vector space models, including BERT, can appropriately represent idiomaticity in noun compounds (NCs) in English and Portuguese. However, their results do not address the following questions: Does BERT distinguish different degrees of idiomaticity in NCs and other types of idioms? What kinds of tokens within an idiom are more predictable? Does the length of an idiom influence BERT’s ability to predict tokens within it?

In this paper, we try to answer these questions by focusing on semantic idiomaticity in French idioms. We took our data from the French Lexical Network (LN-fr), a handcrafted lexical resource containing 3,127 idioms, 22,551 simple lexemes, and 47,395 contextual sentences for these entries. Our experiment used CamemBERT-base (Martin et al., 2020), a pre-trained BERT-derived model for French, in a demasking task on both simple lexemes and tokens embedded within idioms from our dataset.

We compared the prediction results of simple lexemes and tokens within idioms to observe performance differences under different conditions, thereby inferring the model’s representation of different level of idiomaticity. Moreover, we analyzed the effect of token part of speech (POS) and idiom length on performance.

2. Related work

In recent years, attention has been focused on detecting and representing idiomaticity. Handling a MWE within a context requires first recognizing its non-compositional nature and then accurately conveying its idiomatic meaning in this context. Currently, the primary approach involves generating embeddings for components of the MWE and then merging them using diverse composition functions to construct a comprehensive representation of the MWE. Ultimately, the idiomaticity can be evaluated by computing the cosine similarity between the merged vector and the vector representing the expression (Cordeiro et al., 2019).

To represent idiomatic meaning in MWEs, recent approaches typically utilize contextualized language models. Among these models, Shwartz and Dagan (2019) found that BERT outperforms other contextualized models implemented in classifiers for creating embeddings in tasks related to lexical composition. However, Nandakumar et al. (2019) and Garcia et al. (2021a,b) indicated that pre-trained contextual models cannot effectively encode idiomaticity in MWEs. In comparison, static models like word2vec perform better (King and Cook, 2018; Nandakumar et al., 2018, 2019; Cordeiro et al., 2019; Sarlak et al., 2023). Nevertheless, supervised approaches leveraging contextualized models tend to outshine in tasks specific to certain languages and types of MWEs with ample resources, as these models offer representations that encode linguistic features and contextual cues (Fakharian and Cook, 2021).

Idiomaticity has also become a topic of recent NLP conference tasks. For instance, SemEval-2022 task 2 (Tayyar Madabushi et al., 2022) focuses on idiomaticity detection and sentence embedding containing multilingual MWEs. Results of these tasks show that the models got better performance with available training data. Although the best-performing methods are based on deep neural models independent of the linguistic features of MWEs, mixed approaches are generally believed to be worth exploring. Additionally, the PARSEME shared task on automatic identification of verbal MWEs (Ramisch et al., 2020), particularly with the Seen2020 system (Pasquer et al., 2020), underscores the significance of incorporating linguistic features in MWE-related tasks as well.

In our study, we focused on evaluating language models’ sensitivity to idiomaticity. For this, we observed the contextualised model CamemBERT’s performance in a classic fill-mask task with simple and idiomatic tokens in French.

3. Experiment

3.1. Data

We extracted our data form LN-fr v3 (Polguère, 2009; Lux-Pogodalla and Polguère, 2011; Polguère, 2014; ATILF, 2023), released in October 2023. It is an extensive, openly accessible lexical resource constructed manually following the methodological principles of explanatory combinatory lexicology (ECL), the lexicological branch of Meaning-Text Theory (MTT) (Mel’čuk and Polguère, 1987; Mel’čuk et al., 1995; Apresjan, 2000). Every entry in LN-fr is a disambiguated lexical unit, i.e., either a simple lexeme or an idiom with a specific meaning, and each idiom is classified as a weak idiom, a semi-idiom or a strong idiom (see §1). Since our study follows MTT’s definition and classification of idioms, and because LN-fr contains explicit information about the idiomaticity level of idioms, it suited our purpose very well.

Each lexical unit has a POS tag, and that of an idiom is determined by its internal syntactic head rather than its function within a sentence (Mel’čuk, 2006). For instance, bien sûr (‘of course’, lit. ‘well sure’), because its head sûr is an adjective, is described as an adjectival idiom despite functioning as an adverb in sentences. There are a total of 11 POS tags for idioms in our dataset (see Table 2).¹

¹Interjective idioms are expressions that function as...
The POS of the tokens that are embedded within an idiom is not annotated directly in LN-fr, but one can retrieve it from the idiom’s syntactic pattern, which is a string representing a sequence of POS tags. For example, *pomme de terre* (‘potato’, lit. ‘apple of ground’), has the pattern N Prep N, so we know that the first and last tokens are nouns and the second is a preposition. We extracted from independent sentences, like interjections such as *Wow!*

### Table 1: Sample data from LN-fr

<table>
<thead>
<tr>
<th>Lexical unit</th>
<th>Idiomaticity</th>
<th>POS</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>pomme</td>
<td>simple lexeme</td>
<td>N</td>
<td>À la fin du repas, on a parfois droit à un petit morceau de brie et, en guise de dessert, selon la saison, des pommes, des noix, quelques fraises écrasées avec du sucre qu’on étale sur une tartine.</td>
</tr>
<tr>
<td>pomme de terre</td>
<td>weak idiom</td>
<td>N Prep N</td>
<td>Ils prenaient une demi-heure à midi pour manger un œuf sur le plat, une pomme de terre, du fromage blanc. Pierre avait peine à soulever des sacs de pommes de terre de 40 kg, quant à moi je fis un véritable travail de garçon de ferme.</td>
</tr>
</tbody>
</table>

### Table 2: Idiom types in the dataset

<table>
<thead>
<tr>
<th>Idiom type</th>
<th>Example</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal</td>
<td>coup de soleil</td>
<td>1579</td>
</tr>
<tr>
<td></td>
<td>‘blow of sun’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘sunburn’</td>
<td></td>
</tr>
<tr>
<td>Prepositional</td>
<td>à propos</td>
<td>730</td>
</tr>
<tr>
<td></td>
<td>‘at purpose’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘by the way’</td>
<td></td>
</tr>
<tr>
<td>Verbal</td>
<td>faire la tête</td>
<td>619</td>
</tr>
<tr>
<td></td>
<td>‘make the head’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘sulk’</td>
<td></td>
</tr>
<tr>
<td>Conjunctive</td>
<td>quand même</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>‘when even’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘anyway’</td>
<td></td>
</tr>
<tr>
<td>Adjectival</td>
<td>bien sûr</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>‘well sure’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘of course’</td>
<td></td>
</tr>
<tr>
<td>Phrasal</td>
<td>Un ange passe.</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>‘An angel passes.’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘awkward silence’</td>
<td></td>
</tr>
<tr>
<td>Adverbial</td>
<td>pas mal</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>‘not bad’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘quite good’</td>
<td></td>
</tr>
<tr>
<td>Propositional</td>
<td>qui se respecte</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>‘who respects oneself’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘self-respecting’</td>
<td></td>
</tr>
<tr>
<td>Numeral</td>
<td>un à un</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>‘one to one’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘one by one’</td>
<td></td>
</tr>
<tr>
<td>Pronominal</td>
<td>ici et là</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>‘here and there’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘there and here’</td>
<td></td>
</tr>
<tr>
<td>Interjective</td>
<td>Tonnerre de Dieu!</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>‘thunder of God’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘Good heavens!’</td>
<td></td>
</tr>
</tbody>
</table>

### Table 3

The POS tags for most of the embedded tokens. As some idioms did not have a syntactic pattern, we were not able to automatically retrieve the POS for their embedded tokens, which represent about 3.8% of all the tokens in our dataset; these tokens were not included in our second analysis (§4.2).

Each lexical unit has one or more lexicographic examples taken from corpora. These examples have been meticulously selected by lexicographers to reflect the authentic usage of a lexical unit. They aim to showcase various constructions that are possible for the lexical unit, to illustrate its usage and its syntactic and semantic selection (Lux-Pogodalla, 2014). Moreover, the annotation explicitly gives the position, within each sentence, of the tokens that belong to the lexical unit at hand. Note that a lexical unit may appear more than once in the same example; we counted those separately (which is why we have more tokens than examples even for simple lexemes in Table 3). We had in our dataset a total of 47,395 such sentences, with an average of 1.5 examples per idiom and 2 per simple lexeme, each sentence having around 38 tokens on average.

Finally, we counted the length in tokens of each lexical unit. For simple lexemes the length is 1; for idioms, we segmented by spaces and punctuations.

In total, we extracted from LN-fr 25,678 lexical units: 3,127 idioms and 22,551 simple lexemes. Table 3 breaks down these numbers. Compared to the NCs dataset used by Garcia et al. (2021b) covering 9,220 naturalistic and neutral sentences for 280 NCs in English and 180 NCs in Portuguese, our dataset encompasses a broader spectrum of idioms and a larger quantity of contexts.

Our dataset is available at [https://github.com/liliulng/idiomaticity-dataset](https://github.com/liliulng/idiomaticity-dataset).

### 3.2. Methodology

Our experiment consists in taking the sentences associated with a lexical unit in LN-fr and masking, one at a time in the case of idioms, the tokens that correspond to that lexical unit. We then submit these sentences to CamemBERT for demasking. The model predicts the masked token and provides
a list of candidates, each with a softmax score reflecting the model’s confidence in it being the missing token. We record the confidence score returned by the model for the correct answer (the masked token) and note whether the correct answer was ranked as the first candidate (R1). This is illustrated in Table 4. The R1 candidate is the model’s best guess and should be viewed as its “answer”. Its score tends to be close to 1 (indeed, the model is optimized for this), but sometimes it can be lower, which reflects the model’s confidence in its answer (or lack thereof). We want to take this into account, so if the masked token is guessed at rank 1, we note its score, and we will refer to it as “score@R1” in the rest of this paper.

We did not fine-tune the model because we aimed to evaluate the model’s ability to learn idioms without being explicitly trained for it. We used the model as-is with its default parameters.

CamemBERT, as a contextualised model, provides predictions of a masked token based on its context. In our case, the contexts are the sentences retrieved from LN-fr that illustrate the usage of simple lexemes and idioms. Because we mask each token within idioms one by one, the other tokens inside a given idiom are visible and are part of the context. Nedumpozhimana and Kelleher (2021) suggested that BERT’s ability to understand an idiom primarily relies on the idiom itself, so context inside idioms is crucial for CamemBERT to predict masked idiomatic tokens.

We utilized the model’s tokenizer to segment the tokens, guaranteeing that our tokenization was consistent with the model’s vocabulary. In cases where a token was segmented into subtokens, such as the token tigers being tokenized into tiger and s, we conducted the masking experiment for each subtoken and calculated the product of all subtokens’ confidence scores as the confidence score for that token. Furthermore, if the model correctly predicted each subtoken, we marked the whole token as correctly predicted as well.

We analysed the distribution of confidence scores of tokens, scores at rank 1 (scores@R1) and the percentage of correct predictions for masked tokens belonging to simple lexemes and idioms with different idiomaticity degrees, in order to determine how much the model’s prediction is related to masked token’s contextual idiomaticity degree. We further conducted statistical tests to validate the conclusions drawn from our observations.

4. Results and Discussion

In this section, we explore the impact of idiomaticity, POS, and idiom length on the model’s performance. We examine the confidence scores, scores@R1, and the probability of achieving correct predictions token (expressed as a percentage of R1). When analyzing the scores and scores@R1, we take into account the median and mean for tokens across various categories. These are represented, respectively, by a thick line and a triangle in our figures. When there is a notable difference between them, our focus will be on the median.

4.1. Does CamemBERT distinguish different levels of idiomaticity?

Figure 1 shows that 75% of non-idiomatic tokens score below 0.2, with only 10% achieving a high score above 0.8. Conversely, over 40% of idiomatic tokens are predicted with scores exceeding 0.8, highlighting the model’s significant challenge in predicting non-idiomatic tokens. Regarding idiomatic tokens, the model’s confidence scores for correct answers often fall into polarized categories of high or low scores. However, discerning between varying levels of idiomaticity remains difficult, as indicated by similar score distributions across the three types of idioms.

The Kruskal-Wallis test proved the significant difference between the confidence score distribution for tokens corresponding to simple lexemes and that of tokens belonging to idioms ($p < 0.01$, $\eta^2 = 0.15$). There is no significant difference between scores for tokens in the three types of idioms ($p < 0.01$, but with negligible effect size $\eta^2 < 0.01$).

When comparing the mean and median confidence scores (Figure 2), we further notice a
Table 4: Sample fill-mask inputs and results

<table>
<thead>
<tr>
<th>Lexical unit</th>
<th>Token</th>
<th>POS</th>
<th>Sentence</th>
<th>Score</th>
<th>R1</th>
</tr>
</thead>
<tbody>
<tr>
<td>pomme</td>
<td>pommes</td>
<td>N</td>
<td>À la fin du repas, on a parfois droit à un petit morceau de brie et, en guise de dessert, selon la saison, des &lt;mask&gt;, des noix, quelques fraises écrasées avec du sucre qu’on étale sur une tartine.</td>
<td>0.10</td>
<td>F</td>
</tr>
<tr>
<td>pomme de terre</td>
<td>pomme</td>
<td>N</td>
<td>Ils prenaient une demi-heure à midi pour manger un œuf sur le plat, une &lt;mask&gt; de terre, du fromage blanc.</td>
<td>0.99</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>de</td>
<td>Prep</td>
<td>Ils prenaient une demi-heure à midi pour manger un œuf sur le plat, une pomme &lt;mask&gt; terre, du fromage blanc.</td>
<td>0.99</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>terre</td>
<td>N</td>
<td>Ils prenaient une demi-heure à midi pour manger un œuf sur le plat, une pomme de &lt;mask&gt;, du fromage blanc.</td>
<td>0.99</td>
<td>T</td>
</tr>
</tbody>
</table>

Table 5: Percentage of correctly predicted tokens for content and function tokens

<table>
<thead>
<tr>
<th></th>
<th>All</th>
<th>Content</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple lexemes</td>
<td>25</td>
<td>24</td>
<td>50</td>
</tr>
<tr>
<td>Weak idioms</td>
<td>62</td>
<td>55</td>
<td>86</td>
</tr>
<tr>
<td>Semi-idioms</td>
<td>58</td>
<td>48</td>
<td>83</td>
</tr>
<tr>
<td>Strong idioms</td>
<td>62</td>
<td>49</td>
<td>81</td>
</tr>
</tbody>
</table>

Figure 2: Score given to the masked token at all ranks and at R1

significant difference between idiomatic and non-idiomatic tokens. Idiomatic tokens consistently exhibit higher median and mean scores, typically around 0.5 or above. Still, there is no substantial distinction among the three classes of idioms, as tokens within each category demonstrate fairly similar median and mean scores. However, it is worth noting that score@R1, which represents the model’s overall confidence in its predictions, tends to correlate positively with the degree of idiomaticity, which aligns with our previous hypothesis. Additionally, tokens within strong idioms consistently receive the highest median and mean scores, compared to other idiomatic tokens.

**R1 predictions:** The model correctly guesses the masked token around 60% of the time for tokens within idioms, compared to only 25% for simple lexemes. There is no significant difference between the three types of idioms: 62% for weak idioms, 58% for semi-idioms and 62% for strong idioms. This reveals again the model’s higher capacity in predicting tokens within idioms than simple lexemes.

**Statistical analysis:** We calculated the Spearman’s \( \rho \) correlation to unveil the dependence of the model’s prediction results (confidence scores and scores@R1) on tokens’ idiomaticity levels.

Between the free versus idiomatic nature of masked tokens and their prediction results, there is a moderately positive correlation that confirms the model’s capability to distinguish tokens on these two general levels, with \( p < 0.01 \), Spearman’s \( \rho = 0.36 \) for scores and \( p < 0.01 \), Spearman’s \( \rho = 0.39 \) for score@R1. Specifically for all the four levels of idiomaticity (simple lexeme, weak idiom, semi-idiom, strong idiom), this moderately positive correlation still exists between idiomaticity levels and the prediction results (with \( p < 0.01 \), Spearman’s \( \rho = 0.36 \) for confidence scores and \( p < 0.01 \), Spearman’s \( \rho = 0.38 \) for score@R1). As observed in Figure 2, no significant correlation is found between the scores and the three subtypes of idioms (\( p = 0.04 \), \( \rho = 0.02 \)).

This indicates again that, in general, the model is unable to differentiate between varying levels of idiomaticity within idioms, although it effectively distinguishes between free and idiomatic tokens. A chi-squared test between the idiomaticity levels and correct prediction aligns with this conclusion: \( p < 0.01 \) and a moderate effect size Cramér’s \( V = 0.3 \) for all idiomaticity levels and the generally free and idiomatic levels, but \( p < 0.01 \), Cramér’s \( V = 0.03 \) between the three types of idioms.

**4.2. What kinds of tokens are more predictable within idioms?**

We aimed to pinpoint which kinds of tokens present greater predictive challenge and to understand how this might contribute to the observations above. To accomplish this, we broke down our data by the POS of both free and idiomatic tokens. This data was readily available in LN-fr, which distinguishes
Figure 3: Score by token POS

a total of 16 POS tags (distinct from the 11 for idioms listed in Table 2) that can be divided into two categories: content and function tokens. Content tokens represent 94% of the tokens in our dataset and include nouns (N), verbs (V), adjectives (Adj), adverbs (Adv), numerals (Num), interrogative pronouns (ProInter) and interjections (Interj). Function tokens represent the other 6% and include pronouns (Pro), prepositions (Prep), articles (Art), preposition-article amalgams (PrepArt), conjunctions (Conj), personal pronouns (ProPer), pronominal determiners (ProDet), adjectival determiners (AdjDet) and relative pronouns (ProRel). Three of these categories had very low counts, namely Interj (4 occurrences), ProInter (14) and ProRel (5), so the scores reported here for those categories are to be taken with a grain of salt (this explains why the mean is outside of the box for ProInter).

As Figure 3 shows, the median and mean scores for all function tokens are notably higher than those for content tokens, exceeding 0.5. Conversely, the median and mean confidence scores for content tokens are low, with mean scores below 0.3 and median scores below 0.1. This suggests that overall, disregarding idiomaticity, the model excels in predicting function tokens. The score@R1 exhibits the same trend, hence we omit the graph here.

R1 predictions: 82% of function tokens were correctly predicted, against only 28% of content tokens.

Statistical analysis: Spearman’s $\rho$ test demonstrated a moderately positive correlation between predictions and type of POS (content or function token): with $p < 0.01$, $\rho = 0.31$ for confidence scores and $p < 0.01$, $\rho = 0.39$ for score@R1. The chi-squared test also detected a certain level of dependence between the correct prediction of tokens and their POS status ($p < 0.01$, Cramér’s $V = 0.3$).

These results are not surprising, because function words belong to closed classes, thus there are far fewer options for the model to choose from. However, given the model’s adeptness at managing function tokens, we wondered if this could explain its better performance on idioms. Indeed, there is a stark contrast between the distribution of content and function tokens in simple lexemes versus idioms: function tokens comprise only 0.5% of the simple lexemes, while they account for 28.6% of the tokens within idioms. This is because idioms are phrases, so they often contain function words, especially in French, where compounds are much less common than in some other languages such as English or Chinese. Hence, could this imbalance account for the elevated median and mean scores observed for tokens within idioms reported in Figure 2?

We analyzed separately the confidence scores of content and function tokens with varying degrees of idiomaticity. As shown in Figure 4, regarding content tokens, the median and mean scores of idiomatic tokens generally fall below 0.5 but still remain significantly higher than those for simple lexemes. Similarly, there is no substantial disparity in scores among tokens in different types of idioms for content tokens. As for function tokens, those within idioms receive higher confidence scores overall, with mean scores surpassing 0.7 and median scores nearing 1. The variance among different types of idioms is minimal. Conversely, scores for simple function tokens are notably lower than those for idiomatic function tokens, below 0.5. Thus, regardless of the degree of idiomaticity, the model’s prediction of function tokens consistently outperforms that of content tokens. As for content tokens, the model’s prediction of tokens within idioms surpasses that of simple lexemes, and its prediction ability for tokens within idioms with varying degrees of idiomaticity remains stable. This corresponds to our previous conclusion in the first analysis (see §4.1).

R1 predictions: The percentages of correct predictions for content tokens across various levels of idiomaticity further support our findings (see Table 5). Specifically, more than 50% of the content tokens within idioms were correctly predicted, compared to only 24% for simple content tokens. In addition, while roughly half of simple function tokens were correctly predicted, this figure exceeded 80% for idiomatic function tokens.
**Statistical analysis:** We conducted the same statistical analysis for prediction results across idiomaticity levels for content and function tokens separately. Spearman’s $\rho$ correlation between idiomaticity levels and confidence scores or score@R1 always yielded $p < 0.01$ but with no significant $\rho$ values. The chi-squared test showed only modest dependence between correct prediction and idiomaticity level (either considering all four levels or only free versus idiomatic), for both content and function tokens: $p < 0.01$, Cramér’s $V = 0.2$. There is no clear dependence between prediction results and the three idiomaticity levels across idiom subtypes ($p < 0.01$, Cramér’s $V = 0.05$). Thus, the moderate correlation between idiomaticity levels and correct prediction observed in the first analysis no longer exists when we separate content and function tokens. This suggests that the variation in prediction performance of the model between free and idiomatic tokens may actually be at least partly due to the differing proportions of content and function words in these tokens.

No specific POS within content or function tokens appears to significantly influence the model’s performance. The primary types of content words include nouns, verbs, and adjectives. In both simple lexemes and idioms, nouns comprise most of the words, accounting for approximately 61% in simple lexemes, 75% in weak idioms, 78% in semi-idioms, and 66% in strong idioms. Verbs represent a similar portion in simple lexemes (21%) and strong idioms (16%), while they only make up 4% and 6% in weak idioms and semi-idioms. There is no significant difference in the proportion of adjectives across simple lexemes and idioms, ranging from approximately 12% to 18%. Confidence scores for nouns, verbs, and adjectives do not show significant differences. As for function tokens, pronouns (59%), conjunctions (24%), and personal pronouns (10%) are the primary function token types in simple lexemes, while prepositions constitute the main portion of the function tokens in idioms, comprising 74% in weak idioms, 78% in semi-idioms, and 60% in strong idioms. Additionally, preposition-articles are the second major type, accounting for 17%, 14%, and 13% respectively in the aforementioned subtypes of idioms. Notably, the proportion of articles in strong idioms is higher at 15% compared to weak and semi-idioms (2% and 4%).

To sum up, function words tend to be accurately predicted by the model in all types of expressions regardless of the level of idiomaticity, because they belong to closed classes with a small number of members. In free context, their predictability arises from governing syntactic relations and sentence coherence. Meanwhile, within idioms, they contribute to idiomaticity by maintaining the structural integrity and idiomatic meaning of the expression.

**4.3. Is CamemBERT sensitive to the length of idioms?**

When a token in an idiom is masked, CamemBERT utilizes contextual information to predict the masked one, and that context includes the remaining tokens in the idiom. Therefore, the more tokens an idiom contains, the more context it provides. Consequently, does CamemBERT achieve better prediction results for tokens within longer idioms? In our dataset, 99% of idioms comprise 7 tokens or fewer, whereas longer idioms amount to only 1% of the idioms. Most idioms, specifically 91% of weak idioms, 87% of semi-idioms, and 59% of strong idioms, consist of 2 or 3 tokens. Additionally, a small proportion (8% of semi-idioms and 20% of strong idioms) extend to 4 tokens, while another 10% of strong idioms span 5 tokens. No statistically significant relation is found between the level of idiomaticity and the length of idioms.

We compared the score and score@R1 for tokens in lexical units of varying lengths. Here again, the results for score@R1 are not different, so we only present the results for confidence scores in Figure 5. They suggest that as the length of lexical units increases, both the mean and median confidence scores tend to rise (we disregard the drop for lengths over 7 tokens, which we attribute to the scarcity of data in that range).
R1 predictions: Similarly, as shown in Figure 6, when the length of idioms is 7 tokens or fewer, there is a generally increasing trend between idiom length and the percentage of correct predictions.

Statistical analysis: With \( p < 0.01 \), the Spearman’s \( \rho \) coefficient between lexical unit length and scores is 0.36, while it is 0.4 for score@R1, suggesting a moderate positive correlation. Similarly, correct prediction displays a moderate positive association with idiom length in the chi-squared test \( (p < 0.01, \text{Cramér's} \ V = 0.32) \). These findings suggest that the length of idioms significantly impacts CamemBERT’s prediction of idiomatic tokens. The model evidently demonstrates sensitivity to the length of idioms when interpreting tokens within them.

Due to the small proportion (1%) of idioms with lengths exceeding 7 tokens, and despite their proportion of correct predictions not aligning with the general trend, their impact has been disregarded in our analysis.

5. Conclusion

We aimed to assess CamemBERT’s ability to capture varying degrees of idiomaticity within idioms. We measured this by comparing the model’s off-the-shelf performance on fill-mask tasks with tokens pertaining either to simple lexemes or idioms, further distinguishing three levels of idiomaticity among idioms: weak idioms, semi-idioms and strong idioms. We collected 59,092 tokens with illustrative examples from LN-fr, including 45,563 simple lexemes and 13,529 idiomatic tokens from more than 3,000 idioms.

In §1, we posited two hypotheses:

1. CamemBERT should be better at predicting tokens within idioms as opposed to simple lexemes.
2. Tokens within idioms with higher idiomaticity should be more likely to be accurately predicted.

Our main observations are:

1. The model is significantly better at predicting tokens that belong to an idiom as opposed to simple lexemes.
2. It is not sensitive to varying levels of idiomaticity among subtypes of idioms.
3. It exhibits a heightened performance in predicting function words, regardless of idiomaticity.
4. There is a positive correlation between idiom length and performance.

These observations validate our first hypothesis (see §1), but invalidate the second.

Our findings corroborate those of Garcia et al. (2021b), who showed that vector space models, including BERT, cannot capture the semantic overlap between idiomatic NCs and one or none of their components. Furthering their research, we additionally considered weak idioms, which have a semantic overlap with all of their components, as well as a broader range of idioms, not only NCs.

Our analysis of the effects of POS and the length of idioms suggest that these factors may at least partially explain the model’s heightened proficiency at predicting tokens within idioms compared to tokens corresponding to simple lexemes. Nonetheless, this does not explain why CamemBERT is not sensitive to varying levels of idiomaticity among idioms. The very notion of idiomaticity is ambiguous, and the distinction between various types of idiomaticity is often overlooked and tends to be conflated into semantic aspects, i.e., non-compositionality. In our study, we explored both lexical and semantic idiomaticity. Lexical idiomaticity implies that idiomatic tokens exhibit stronger constraints on lexical selection compared to free tokens, i.e., they cannot be replaced by their synonyms while preserving their idiomatic meaning and grammatical correctness. On the other hand, the varying degrees of idiomaticity are indicative of their semantic idiomaticity, which denotes the contribution of internal components to their overall semantic meaning. So CamemBERT’s performance in our experiment suggests that in fact the model is more sensitive to lexical idiomaticity than semantic idiomaticity.

This raises questions about other aspects of idiomaticity. Indeed, idioms exhibit idiomaticity on multiple levels simultaneously: lexical, semantic, syntactic, morphological, etc. For instance, faire la tête (‘sulk’, lit. ‘make the head’) is a strong idiom in French that exhibits not only lexical and semantic idiomaticity, but also prohibits syntactic operations like passivisation, dislocation, etc., as well as morphological inflection to tokens other than the head faire. While there is no theoretical consensus on the classification of idiomaticity, our experience may offer valuable insights to address the matter.

In future research, we would like to refine our experiment, extend it to other types of MWEs and explore other forms of idiomaticity. Moreover, we intend to carry out further analyses on language model representations of idiomaticity, exploring additional potential influencing factors such as idiom frequency, or extending our investigation to more complex tasks. We also aim to replicate our experiments with different language models and available datasets in other languages.

6. Acknowledgements

We express our gratitude to the anonymous reviewers for their valuable and constructive feed-
back. We would like to thank ATILF for the original dataset and our colleagues at OLST for engaging in helpful discussions. Li Liu acknowledges the financial support of the China Scholarship Council (#202008310177).

7. Bibliographical References


Veronika Lux-Pogodalla. 2014. Integrating lexicographic examples in a lexical network (intégration relationnelle des exemples lexicographiques dans un réseau lexical) [in French]. In Proceedings of TALN 2014, volume 2, pages 586–591, Marseille, France. ATALA.


Navnita Nandakumar, Timothy Baldwin, and Bahar Salehi. 2019. How well do embedding models


Identification and Annotation of Body Part Multiword Expressions in Old Egyptian

Roberto Antonio Díaz Hernández
University of Jaén
Campus Las Lagunillas, Building D2, Office 352, 23071, Jaén
radiaz@ujaen.es

Abstract

This paper presents the preliminary results of an ongoing study on the diachronic and synchronic use of multiword expressions (MWEs) in Egyptian, begun when I joined the COST Action Universality, Diversity and Idiosyncrasy in Language Technology (UniDiVe, CA21167). It analyzes, as a case study, Old Egyptian body part MWEs based on lexicographic and textual resources, and its aim is both to open up a research line in Egyptology, where the study of MWEs has been neglected, and to contribute to Natural Language Processing studies by determining the rules governing the morpho-syntactic formation of Old Egyptian body part MWEs in order to facilitate the identification of other types of MWEs.

Keywords: Old Egyptian, Multiword Expression, Body Part

1. Introduction

Egyptian is one of the longest lived languages in history. This Afroasiatic language knew the following phases:

- Old Egyptian (ca. 2700–2000 BC).
- Middle Egyptian (ca. 2000–1400 BC).
- Late Egyptian (ca. 1300–700 BC).
- Demotic (7th century BC to 5th century CE).
- Coptic (4th century to 14th century CE).

This paper shows the existence of MWEs in one of the oldest known languages in human history, as they are attested in texts dating from the early third millennium BC (see example 15, below). It focuses on the use of body part MWEs in Old Egyptian, analyzes their typology and identifies rules for their formation. This paper has seven parts. It begins with a brief introduction to the topic (§ 1) and a definition of "body part multiword expression" (§ 2). The methodological approach applied to the identification and annotation of Old Egyptian body part MWEs (§ 3) is followed by examples of each body part noun used in Old Egyptian MWEs (§ 4). A typology of Old Egyptian body part MWEs (§ 5) and an explanation of the rules governing their formation (§ 6) are developed on the basis of the examples collected during the research. Finally, the next phases of this research are outlined in the conclusion (§ 7).

2. A Definition of a Body Part Multiword Expression

It is assumed that "body" and "body parts" are universal concepts (Wierzbicka, 2007) which can be used with a metonymic and metaphoric meaning (Ganfi, Piunno and Mereu, 2023). A body part MWE may be defined as a sequence of at least two lexicalized components, one of which is a body part name, whose semantic idiosyncrasy results from the association between the body part with a figurative meaning and another component(s) (cf. the definition of MWE in Savary et al., 2018; Baldwin and Kim, 2010). Body part MWEs are common in modern and ancient languages, e.g.:

1. English:
LM: "Listen to your heart."  
FT: "Act according to your feelings."

2. Latin (Plaut., Asin. 729):
   nec caput nec pes
   neither-neg head nor-neg foot
LM: "Neither head nor foot."
FT: "Completely wrong."

3. Arabic:
   al-qalb
   the heart-M.SG. DET
   weak-M.SG
LM: "A weak one of heart."
FT: "A coward."

3. Methodology

Although multiple forms of figurative language, such as simile and metaphor have been extensively studied in Egyptology, the study of MWEs remains unexplored. Old Egyptian body part MWEs was chosen as a case study for this work because of the occasional metonymic and metaphoric use of body part nouns (see § 2, above)—a factor that facilitates the identification of MWEs in any language (see examples 1–3, above). Lexical compounds with an idiosyncratic meaning consisting of a body part noun in a close relationship with its head word were considered as MWEs, as for example:

---

1 Earlier instances of MWEs may be found in Sumerian texts from the Early Dynastic Period (ca. 2900 BC).
2 LM stands for "literal meaning" and FL for "free translation".
3 For the state of the art in Egyptian figurative language, see Hsu 2023.
4. CG 20543, 5:

enter:PTCP(M.SG) heart-M.SG mistress-F.SG =3SG.M

LT: “One who enters the heart of his mistress.”

FT: “A confidant of his mistress.”

Metaphorical expressions used to establish a figurative comparison of two entities by means of a “comparison marker”, such as mr “like” in Old Egyptian were disregarded in this research, for example:

5. Pyramid Texts § 293a:

soar-SBJV =2SG.M like:PREP heron-M.SG

LM: “You shall soar (skyward) as a heron.”

FT: “You shall fly over the clouds.”

A fuzzy boundary represents the case where the body part noun has a metonymic meaning, while the head word retains its literal meaning. Such cases were included as MWEs in this research (see identification tests 2 and 3, below), as for example:

6. Pyramid Texts § 1592e:

love:REL.PRS heart-M.SG in:PREP =3SG.M

LM: “(... any place) which his heart (i.e. will) loves.”

FT: “(... any place) which he desires.”

Body part MWEs are clearly identified when its figurative meaning results from the close association of the body part noun with its head word, as for example:

7. Pyramid Texts § 22b:

be cool:SBJV heart-M.SG =2SG.M

LM: “Your heart may be cool.”

FT: “You may be calm (i.e. satisfied).”

In Egyptian the figurative meaning of a body part MWE is often related to the idiosyncrasy of this language, as the following example shows:

8. Pyramid Texts § 417b:

im(i) rt “one who is in the foot”

The figurative meaning of this expression is “enemy”, for it derives from the Egyptian custom of decorating sandals with the image of foes:

Although MWEs are not identified in Hannig’s Old Egyptian dictionary, it provides extensive references to the meaning of each Egyptian word and lexical compound:

---

Fig. 1: Foot-end of mummy cartonnage (Veldmeijer, 2014)

Fig. 2: Textual references to the MWE wḏʒ ib “be happy” (Hannig 2002: 398-399)

I checked the references of body part nouns potentially used in MWEs against the editions of hieroglyphic texts. Instances of body part nouns with a literal meaning have been disregarded (see validation test 1, below), while instances of body part nouns in figurative association with other words have been considered body part MWEs according to the definition given in section 2 (see above). In addition, I used the textual database of the Thesaurus Linguae Aegyptiae to find further instances of body part MWEs in Old Egyptian texts. After selecting and entering them into an Excel list, I manually annotated the most eloquent examples of Old Egyptian body part MWEs.
in a Word file for lack of a digital resource. Such examples have a clear meaning and syntactic structure. As it can be seen here, they were annotated with the reference source and following the Leipzig Glossing Rules. They are provided with a literal meaning (LM) and a free translation (FT), for example:

9. Pyramid Texts § 293a:


The selection and identification of Old Egyptian body part MWEs was carried out using a series of verification tests:

Test 1. Does the body part noun have a literal meaning?
— Yes ⇒ It is not an MWE, for example:

10. Pyramid Texts § 49 Nt:

LM: “Seize for yourself his arm.” FT: “Seize his arm!”
— No ⇒ Test 2.

Test 2. Does the body part noun have a metonymic meaning?
— Yes ⇒ It is a potential MWE ⇒ Test 3. Ex.:

11. Pyramid Texts § 1675b:

LM: “Your heart shall guide you.” FT: “Your will shall guide you.”
— No ⇒ It is not an MWE, see test 1.

Test 3. Is the body part noun used with an idiosyncratic meaning in close syntactic relationship with a head word?
— Yes ⇒ It is an MWE, for example:

12. Pyramid Texts § 116a:

LM: “May (I) ask your face.” FT: “Hail to you!”

Test 4. Is the body part noun used in a lexicalized expression with an idiosyncratic meaning?
— Yes ⇒ It is an MWE. This is the usual case for complex prepositions (CPs), for example:

13. Pyramid Texts § 54b:

LM: “Lift up in front of his face.” FT: “Lift up before him”

It should also be noted that body part MWEs are occasionally attested in some scenes, which are annotated here in order to illustrate their meaning, for example:

14. Davies, 1900, pl. III, cf. fig. 3:

LM: “Your heart shall be hale concerning the crocodile.” FT: “You shall be happy of having escaped from the crocodile.”

Fig. 3: A cow escapes from the crocodile

In Old Egyptian the frequency of body part MWEs varies depending on the body part noun—the commonest body part MWEs are those consisting of lb “heart” (no less than 63 types of MWEs) and “arm” (no less than 24 types of MWEs), while the less common body part MWEs are those consisting of ỉr “eye”, ỉn “nail” and ḫpḥ “biceps” which are attested in less than five types of MWEs. The following Old Egyptian body part nouns are used in MWEs (see examples in § 4, below):

4 I am working on the first treebank of Egyptian sentences syntactically analyzed in Universal Dependencies. Its initial release is planned for May 2024. This treebank will include MWEs to be published as a corpus in PARSEME.

5 Di Blasi-Dyson, Kemmerzell and Werning (2009) adapted the Leipzig Glossing Rules for the study of Egyptian texts.

6 I discussed the use of Old Egyptian MWEs containing lb in a poster I presented at the second general meeting of UniDive (Università di Napoli “L’Orientale”, 8–9 February 2024). The 63 types of Old Egyptian MWEs containing lb are analyzed one by one in my forthcoming article “Old Egyptian Multitword Expressions consisting of a head word and lb ‘heart’”.

26
Table 1: Body part nouns used in Old Egyptian MWEs

<table>
<thead>
<tr>
<th>Spelling</th>
<th>Transcription</th>
<th>Literal meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ꜣm</td>
<td>lwf</td>
<td>flesh</td>
</tr>
<tr>
<td>ꜣb</td>
<td>ib</td>
<td>heart</td>
</tr>
<tr>
<td>ꜣr.t</td>
<td>ir.t</td>
<td>eye</td>
</tr>
<tr>
<td>ꜣr</td>
<td>Ṳ</td>
<td>arm</td>
</tr>
<tr>
<td>ꜣn.t</td>
<td>ʾn.t</td>
<td>nail</td>
</tr>
<tr>
<td>rʾ</td>
<td>rmn</td>
<td>shoulder</td>
</tr>
<tr>
<td>Ṳṭ</td>
<td>rṭ</td>
<td>mouth</td>
</tr>
<tr>
<td>ꜣb</td>
<td>bꜣnt</td>
<td>mouth-M.SG</td>
</tr>
<tr>
<td>Ṣ</td>
<td>nḥ</td>
<td>neck</td>
</tr>
<tr>
<td>ṣḥ</td>
<td>šḥ</td>
<td>forehead</td>
</tr>
<tr>
<td>ṣḥꜣ.t</td>
<td>šḥꜣ.t</td>
<td>heart</td>
</tr>
<tr>
<td>ṣḥpꜣ.t</td>
<td>šḥpꜣ.t</td>
<td>strong arm (biceps)</td>
</tr>
<tr>
<td>bꜣt</td>
<td>bꜣt</td>
<td>belly</td>
</tr>
<tr>
<td>ẖ</td>
<td>ẖ</td>
<td>back</td>
</tr>
<tr>
<td>ẖn</td>
<td>ḫn</td>
<td>hair</td>
</tr>
<tr>
<td>ṣp</td>
<td>ṣp</td>
<td>head</td>
</tr>
<tr>
<td>ṣpꜣ.t</td>
<td>ṣpꜣ.t</td>
<td>body</td>
</tr>
<tr>
<td>ṣbꜣ.t</td>
<td>ṣbꜣ.t</td>
<td>finger</td>
</tr>
</tbody>
</table>

4. Evidence

The earliest instances of body part MWEs in Egyptian date from the Early Dynastic Period (ca. 2900–2730 BC), for example:

15. Petrie, 1901 (vol. 2, pl. III), cf. fig. 4:

Fig. 4: An Abydos tablet

As the following examples show, body part nouns listed in table 1 (see above) are used to form Old Egyptian MWEs.

16. Example of lwf (Sethe, 1933, 14,5):

\[ prr \frac{sf}{m} of \frac{m}{m} lWF \]

LM: “He goes off for him with his flesh.”
FT: “He goes off, certainly at his own risk.”

17. Example of ib (Brunner, 1937, 62,79):

\[ \frac{m}{m} ib \]

be hostile-PTCP(M.SG) heart-M.SG every-M.SG
LM: “Everyone who is hostile of heart.”
FT: “Any evil-minded person.”

18. Example of ir.t (Černý, 1961, 7):

\[ \frac{n}{n} psd \frac{-}{-} \frac{m}{m} ir.t \frac{n}{n} nfr \]

LM: “(I) did not spit in the two eyes of a good one.”
FT: “(I) did not spit on the eyes of a good man (i.e. I did not humiliate a good man).”

19. Example of ¢ (Pyramid Texts, § 213a):

\[ \frac{r.l}{r.l} \frac{ψ}{ψ} \]

LM: “(…) in the interior of your arm.”
FT: “(…) within your embrace.”

20. Example of ʾn.t (Moussa/Altenmüller, 1977, 79 and fig. 10), cf. fig. 5:

Fig. 5: A man pedicuring another man

As the following examples show, body part nouns listed in table 1 (see above) are used to form Old Egyptian MWEs.
22. Example of rmn (Pyramid Texts, § 813a):

\[
\text{ḥmś.y ⸗f ḥr rmn(wi) ⸗f sit:FUT =3SG.M on:PREP shoulder-M.DU =3SG.M}
\]

LM: "He will sit on his two shoulders."
FT: "He will sit himself beside him."

23. Example of \( r(t) \) (Kanawati 1997, fig. 41):

\[
ič n sk r(t(wi)) sk
\]

take:IMP for:PREP =2SG.M foot-M.DU =2SG.M

LM: "Take for you your two feet."
FT: "Move!"

24. Example of \( hr \) (Pyramid Texts, § 613a):

\[
\text{ḥꜣ rṭ hr sk}
\]

make bright:SUBJV =3PL face-M.SG =2SG.M

LM: "They shall make your face bright."
FT: "They shall make you glad."

25. Example of \( bꜣt \) (Pyramid Texts, § 407d):

\[
iw mk.t Wnḫ m bꜣt
\]

PTCL place-f.SG Unas-KG at:PREP forehead-f.SG

LM: "The place of Unas is at the forehead."
FT: "Unas’ place is ahead."

26. Example of \( bꜣt.t \) (Pyramid Texts, § 2024a):

\[
\text{ḥꜣ bꜣt.t sk}
\]

be great:SUBJV heart-M.SG =2SG.M

LM: "Your heart shall be great."
FT: "Be proud!"

27*. Example of \( ḥpꜣš \) (Fischer, 1961, 47):

\[
\text{ḥr m ḥpꜣš ⸗f}
\]

act:PTCP(M.SG) with:PREP biceps-M.SG =3SG.M

LM: "One who acted with his biceps."
FT: "One who acts on his own."

28. Example of \( bꜣt.t \) (Pyramid Texts, § 1c):

\[
\text{sꜣ pw Ttì n(.t) bꜣt.t (at)}
\]

son-M.SG COP Teti-KN of-M.SG belly-f.SG =1SG

LM: "Teti is the son of (my) belly."
FT: "Teti is (my) bodily (i.e. biological) son."

29. Example of \( ṣꜣ \) (Sethe, 1933, 111,8):

\[
\text{ḥr ṣꜣ ḥb.t}
\]

on:PREP back-M.SG foreign land-f.SG

LM: "(...) on the back of the foreign land."
FT: "(...) at the far end of the foreign land."

30. Example of \( sꜣn (Petrie, 1900, pl. XXVB)\):

\[
\text{sꜣn tī}
\]

hair-M.SG earth-M.SG

LM: "Hair of the earth."
FT: "Vegetation."

31. Example of \( ṭp \) (Pyramid Texts, § 989a):

\[
\text{ṭp hrw}
\]

in:PREP head-M.SG day-M.SG

LM: "(...) in the head of the day."
FT: "(...) at dawn."

32*. Example of \( c.t \) (Pyramid Texts, § 762b):

\[
mw sk c.t sk
\]

speak:SUBJV =2SG.M body-M.SG =2SG.M

LM: "You shall speak (of) your body."
FT: "You shall speak (of) yourself."

33. Example of \( ḫb' \) (Pyramid Texts, § 372a):

\[
r ḫb'(wi) of
\]

to:PREP finger-M.DU =3SG.M

LM: "(...) to his two fingers."
FT: "(...) at his side."

34. Example of \( ēr.t \) (Brunner, 1937, 42,3):

\[
\text{ink pg tī ēr.t}
\]

1SG open:PTCP(M.SG) hand-f.SG

LM: "I am one who opens the hand."
FT: "I am a generous one."

5. Typology

Old Egyptian body part MWEs can be classified according to universal typology as nominal, prepositional and verbal. In nominal multiword expressions (NMWEs) the head word accompanying the body part noun can be a noun, an infinitive, an adjective or a participle. The head word of prepositional multiword expressions (PMWEs) can only be a preposition. In verbal multiword expressions (VMWEs) the head word must be a verb form (except if it is a nominalized verb form which is considered an NMWE).

5.1 Nominal Multiword Expressions

A body part noun can be the head or the modifier of an NMWE. If it is the former, it usually means a physical object, for example:
35. Goedicke, 1994, 73, l.9, cf. fig. 6:
LM: “An arm (made) of wood.”
FT: “An incense burner (in the shape of an arm).”

Fig. 6: A ritualist holding an incense burner
(Walters Art Museum 22216)

If the body part noun is used as a modifier, the head of the NMWE can be a noun, an infinitive, and an adjective or a participle:
36. Example of a noun as the head of an NMWE (Junker, 1943, fig. 43):
LM: “A warm man of heart.”
FT: “A hard-working man.”

37. Example of an infinitive as the head of an NMWE (Paget, 1898, pl. XXXVIII), cf. fig. 7:
LM: “Bringing the foot.”
FT: “Erasing the footprint (a ritual ceremony).”

Fig. 7: A ritualist “erasing the footprint”

38. Example of an adjective as the head of an NMWE (Pyramid Texts, § 195c):
LM: “How beautiful is your (f.) face.”
FT: “How nice is to see you.”

39. Example of a participle as the head of an NMWE (Pyramid Texts, § 1a):
LM: “(...) Teti who opened (my) belly.”
FT: “(...) Teti, (my) first-born.”

5.2 Prepositional Multiword Expressions

Body part nouns are used as modifiers in prepositional multiword expressions. Two types of PMWEs can be found in Old Egyptian: prepositional idioms (PIs) and complex prepositions (CPs).
40. Example of a prepositional idiom (Sethe, 1933, 162,11):
LM: “(...) under the arm of (my) eldest son.”
FT: “(...) under the care of (my) eldest son.”

41. Example of a complex preposition (Sethe, 1933, 126,2):
LM: “(...) in his back.”
FT: “(...) behind him.”

5.3 Verbal Multiword Expressions

Body part nouns are also modifiers in VMWEs. Old Egyptian body part VMWEs are usually verbal idioms (VIs) consisting of a verb as a head and a body part noun with a figurative meaning, for example:
42. Pyramid Texts, § 425a:
LM: “(...) he filled the mouth of Unas.”
FT: “(...) he fed Unas.”

Light Verb Constructions consisting of a “light” verb and a noun denoting an event or a state, such as “make a speech” are hardly found in Old Egyptian body part VMWEs. However, the metonymic meaning of body part nouns occasionally refers to an action which modifies the meaning of the expression, for example:

\[ \text{Cf. Savary et al., 2018, 99 and 102; Baldwin and Kim, 2010, 277.} \]
43. Duell, 1938, pl. 162, cf. fig. 8:

LM: “(I) shall do according to your will (lit.: heart).”
FT: “(I) will do what you want.”

Fig. 8: A boy following the instructions of his friends

6. Formation Rules

The formation of Old Egyptian body part MWEs follows strict morpho-syntactic rules, which are useful not only for understanding how an MWE was used in Old Egyptian, but also for identifying other types of MWEs. Five formation rules are derived from the morpho-syntactic analysis of Old Egyptian body part MWEs:

1) A verb stem in a VMWE can be transformed into an infinitive in an NMWE, cf.:

44. Example of a VMWE consisting of the subjunctive ḥw + ḫb (Pyramid Texts, § 715c):

LM: “(He) will give the gods shall be long in Teti.”
FT: “The gods shall be glad over Teti.”

45. Example of an NMWE consisting of the infinitive ḥw.t + ḫb (Pyramid Texts § 1175a):

LM: “The earth is in length of heart.”
FT: “The earth is in joy.”

2) A verb stem in a VMWE can be transformed into a participle in an NMWE, cf.:

46. Example of a VMWE consisting of the verb form ḥw + ḫr (Pyramid Texts, 391c):

LM: “The face of the god will be open to Unas.”
FT: “The god will view the king with favour.”

47. Example of an NMWE consisting of the participle ḥw + ḫr (Sethe, 1933, 149,1):

LM: “One who opens the face to the troops.”
FT: “One who views the troops with favour.”

Note that deverbal constructions resulting from a VMWE into an NMWE are also found in other languages, such as English:

“She makes decisions quickly” > “She is a quick decision maker” (see Savary et al., forthcoming).

3) A preposition in a PMWE can be transformed into a nisba adjective in an NMWE, cf.:

48. Example of a PMWE consisting of the preposition ḥr + (Sethe, 1933, 162,1):

LM: “(... under the arm of (my) eldest son.”
FT: “(... under the care of (my) eldest son.”

49. Example of an NMWE consisting of the nisba adjective ḥr.(i)w + (Pyramid Texts, § 1236b):

LM: “Those who are under the arm of Osiris.”
FT: “Those who are under the care of Osiris.”

4) The nisba adjective resulting from a preposition can be used as a noun in an NMWE, for example:

50. Goedicke, 1968, 27:

LM: “One who is under the arm.”
FT: “One who is under the care (i.e. assistant).”

Note that the usual transformation of a preposition in a PMWE into a nisba adjective or a noun in an NMWE is an idiosyncratic feature of Old Egyptian hardly found in other languages. This is a common way for the formation of Egyptian titles, for example the title ḥr.(i) tp “great chief” is derived from the PMWE ḥr tp “on the head”, cf.:

Schulz 2010, 86). The addition of the nisba ending to prepositions to form adjectives and nouns is a common feature in Egyptian.
51. Pyramid Texts 1487a:

\[
\begin{array}{c}
\text{šw} & \text{k} & \text{ḥr} & \text{ṭp} & \text{k} \\
\text{shade-M.SG} & \text{=2SG.M} & \text{on:PREP} & \text{head-M.SG} & \text{=2SG.M}
\end{array}
\]

LM: “Your shade is on your head.”
FT: “Your shade is over you.”

52. Sethe 1933, 254, 4:

\[
\begin{array}{c}
\text{ḥr} & \text{n.ḥ} & \text{ṣp} & \text{t} \\
\text{one who is on-M.SG} & \text{head-M.SG} & \text{of-M.SG} & \text{name-F.SG}
\end{array}
\]

LM: “One who is on the head of the nome.”
FT: “Great chief of the nome (i.e. nomarch).”

5) An NMWE consisting of a noun as its head word can be transformed into a PMWE by adding a preposition before the noun, cf.:

53. Example of an NMWE consisting of the nouns ṣ.t + īb (CG 1485):

\[
\begin{array}{c}
\text{ḥm-nār} & \text{ṣ.t} & \text{ib} & \text{nb} \\
\text{priest-TITLE} & \text{place-F.SG} & \text{heart-M.SG} & \text{lord-M.SG} & \text{=3SG.M}
\end{array}
\]

LM: “The priest of the place of the heart of his lord.”
FT: “The priest beloved of his lord (i.e. the favourite priest of his lord).”

54. Example of a PMWE consisting of the preposition mr + ṣ.t ḫb (Sethe, 1933, 56, 19):

\[
\begin{array}{c}
\text{mr} & \text{ṣ.t} & \text{ḥb} & \text{n.ṭ} & \text{ḥm} \\
\text{like:PREP} & \text{place-HEART} & \text{of-F.SG} & \text{majesty-} & \text{=3SG.M}
\end{array}
\]

LM: “I (used to act) like the place of the heart of his majesty.”
FT: “I (used to act) at the request of his majesty.”

7. Conclusion

This research leads to the following preliminary results:

1) The existence of MWEs is indisputable in Old Egyptian, which means that they are as old as the Pyramids of Giza.

2) Body part nouns are used in Old Egyptian to form MWEs, which means that Old Egyptian phrases containing a body part noun with a metonymic meaning are potential candidates to be identified as MWEs.

3) The typology of body part MWEs in Old Egyptian is similar to that applying to MWEs in other languages. Research on MWEs in Egyptian will be continued in these two phases:

1) Publication of the selected examples in PARSEME after having annotated them manually in the Universal Dependencies treebank “Egyptian-UJaen”.

2) Identification and classification of new Old Egyptian MWEs following the rules discussed in this paper and the identification tests suggested in Savary et al., 2018.

Once the synchronic study of MWEs in Old Egyptian is completed, their analysis in later stages of Egyptian will follow in order to detect changes during their historical development. This will contribute not only to the confirmation of the universal categorization of MWEs, based mostly on modern Indo-European languages, but also to the development and refinement of universal rules concerning the formation of MWEs. The end result of this research will be a manually annotated digital corpus of Egyptian MWEs published in PARSEME and a lexicon of Egyptian MWEs.

8. Acknowledgments

I thank UniDive for having introduced me to the study of NLP.

9. Bibliographical References


Kanawati, N. and Hassan, A. (1997). The Teti Cemetery at Saqqara. Volume II. The Tomb of...
Ankhmahor. The Australian Centre for Egyptology Reports 9, Wiltshire.

10. Language Resource References
Thesaurus Linguae Aegyptiae: https://thesaurus-linguae-aegyptiae.de/home
Fitting Fixed Expressions into the UD Mould: Swedish as a Use Case

Lars Ahrenberg
Department of Computer and Information Science
Linköping University
lars.ahrenberg@liu.se

Abstract

Fixed multiword expressions are common in many, if not all, natural languages. In the Universal Dependencies framework, UD, a subset of these expressions are modelled with the dependency relation fixed, targeting the most grammaticalized cases of functional multiword items. In this paper we perform a detailed analysis of 439 expressions modelled with fixed in two Swedish UD treebanks in order to reduce their numbers and fit the definition of fixed better. We identify a large number of dimensions of variation for fixed multiword expressions that can be used for the purpose. We also point out several problematic aspects of the current UD approach to multiword expressions and discuss different alternative solutions for modelling fixed expressions. We suggest that insights from Constructional Grammar (CxG) can help with a more systematic treatment of fixed expressions in UD.

Keywords: Multiword expressions, fixed expressions, constructions, Swedish

1. Introduction

Multiword expressions (MWEs) are ubiquitous in many, if not all, natural languages. They are usually divided into different classes with fixed, word-like expressions at one end and flexible phrase- and clause-like expressions at the other. Common English examples of these two kinds are illustrated in (1) and (2):

(1) at first, by and large, of course
(2) give X the creeps, beat around the bush

How do you search for MWEs in a treebank annotated in the Universal Dependencies (UD) framework? That would depend on the type of MWE you are interested in. UD offers three relations to represent MWEs: compound, flat and fixed (de Marneffe et al., 2021). The first is focused on compounding of nouns and other content words, the second on fixed expressions with similar behavior as function words, and the third primarily on multiword names. For definitions see Table 1. If your interest is with the flexible ones, however, you would have to use the key words of the MWE such as creeps or around the bush, as there is no particular relations devoted to them; they are annotated the same way as compositional phrases and clauses. Alternatively, you can turn to treebanks with more flexible annotations such as those developed in the PARSEME project with special annotations for verbal multiword expressions (Savary et al., 2023a).

The stated purpose of UD is to develop crosslinguistically consistent morphosyntactic annotation for as many languages as possible. The main purposes are to support research in language typology and natural-language processing, parsing in particular. Given that MWEs sometimes show deviant morphosyntactic behaviour and that the knowledge of MWEs crosslinguistically appears to be scarce (Masini, 2019) we can argue that MWEs should be given adequate representations in UD annotation. Then it is a problem that it does not cover all types of MWEs. While this problem has been recognized (Savary et al., 2023b), no solution has been agreed upon so far.

A framework that places MWEs at the center of linguistic modelling is Construction Grammar (CxG) (Fillmore et al., 1988; Booij, 2017; Hoffmann, 2022). The most radical view of CxG holds that everything in language, from morphs to sentences, are instances of form-meaning pairs of the same basic kind, called constructions. A form is a pattern of some sort and the meaning may be more or less specific. In contrast, UD only recognizes the existence of certain MWEs and by using the syntactic level of annotation it actually blurs the fact that MWEs often have a transparent syntactic structure; MWEs don’t have to be syntactically deviant.

The empirical basis of the paper is a detailed analysis of the formal and structural variation in MWEs currently annotated as fixed in two Swedish UD treebanks. All expressions in this dataset have been annotated for the type of variation they accept, their distribution if regarded as a UD word, and for their structure. The latter aspect takes inspiration from the treatment of MWEs in Construction Grammar, in particular the idea that structures can enter into hierarchical relations. While the data is primarily taken from Swedish they illustrate general types of problems in relation to fixed MWEs. Comparisons are made with the use of fixed in UD treebanks for English.
Table 1: Definitions of the three dependency relations used for MWEs in UD cited from (de Marneffe et al., 2021)[266]

<table>
<thead>
<tr>
<th>Relation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>compound</td>
<td>any kind of word-level compounding (noun compound, serial verb, phrasal verb)</td>
</tr>
<tr>
<td>fixed</td>
<td>fixed multiword expression; links elements of grammaticalized expressions that behave as function words or short adverbials</td>
</tr>
<tr>
<td>flat</td>
<td>flat multiword expression; links elements of headless semi-fixed multiword expressions like names</td>
</tr>
</tbody>
</table>

The paper is structured as follows. The next section provides background on fixed MWEs as found in general overviews, in Usage CxG, and in UD. Section 3 presents our dataset and how it has been annotated. In Section 4 we review a number of common types of fixed MWEs found in the dataset and discuss how they can be analysed with or without the fixed relation. Section 5 proposes alternative ways to annotate them in UD. Section 6, finally, holds the conclusions.

2. Multiword Expressions in Different Frameworks

A common taxonomy for MWEs splits them first into lexicalized phrases and institutional phrases (Baldwin and Kim, 2010). Only the lexicalized phrases provide examples of syntactically deviant structures. They are in turn divided into fixed, semi-fixed, and syntactically flexible. This division can be seen as points on a scale from the most rigid to the fully compositional phrases (Masini, 2019). Here the focus will be on the fixed MWEs.

(Baldwin and Kim, 2010) defines fixed MWEs as expressions ‘that undergo neither morphosyntactic variation nor internal modification, often due to fossilisation of what was once a compositional phrase.’ Expanding on this definition we have identified a number of ways in which a fixed MWE can vary, which is detailed in Section 3.

An interesting aspect of this definition is that it views fixed MWEs as isolated examples. Similarity of structure to other fixed MWEs seems to play little role. However, to determine whether an expression is fixed or flexible it is important to look for structural patterns that are common to sets of expressions, a key feature of Construction Grammar.

2.1. On Constructions

There are a number of variants of Construction Grammar but all of them use a notion of construction as a pairing of form and meaning. This applies to words and morphs as well as to phrases and clauses. The form level may include phonetic and/or orthographic information as well as morphological and syntactic information. Meaning may include semantic as well as pragmatic information (Hoffmann, 2022).

The morphosyntactic information is not restricted to parts-of-speech and morphological features. Depending on the scope of a construction the application of a category may be constrained in various ways, for instance to a subset of nouns or adjectives. Moreover, constructions are related to one another via inheritance links and horizontal links. In this way a phrase that seems deviant or special may be linked to a more regular pattern as a specialisation.

2.2. An Example

There is a set of Swedish time adverbials that are marked by the simultaneous occurrence of the preposition i, ‘in’ and a final suffix -(a)s on the following noun. The nouns are restricted to a finite number of words referring to week-days, seasons, or parts of the day. The suffix only occurs in this pattern. All expressions of the pattern are deictic and the meaning is, roughly, a reference to the most recent period of the kind signified by the noun:

- i lördags this past Saturday
- i våras this past spring
- i julas this past Christmas
- i förmiddags this past (late) morning

It is important to note that the nouns cannot be put in other nominal positions, not even as possessive modifiers. While -s is a genitive suffix in Swedish, the nouns in this group are seldom seen as possessive modifiers. For example, to say the equivalent of English ‘the events of Saturday’, in Swedish, we need to use a definite form, lörda-gens händelser, whereas an indefinite form such as *lördags händelser on its own is out.

A construction in Usage Construction Grammar (Hoffmann, 2022) representing this set of time adverbials may be written as in Table 2.

Instances of this pattern that are found in Swedish UD treebanks are all annotated with the

---

1 The label kalenderplacering.genitiv, ‘calendar placement, genitive’, which is found in the Swedish Construction (Borin et al., 2012; Lyngfelt et al., 2018) for these expressions is therefore unfortunate.
FORM: \( [i \text{ NOUN}_\text{temp} + (a)s] \)
MEANING: this past \( TIME^{1}\)

Table 2: A construction in the style of a Usage CxG. The index links the noun in the FORM part to its corresponding predicate class in the MEANING part.

relation \textit{fixed}. While there are only a finite number of them there is a clear pattern that capture their form as well as their meaning.

In a CxG patterns can be related to each other via inheritance, or as specifications of a common more general pattern. In the example we refer to more specific variables than ordinary parts-of-speech, such as week-days or seasons. This option is not available in UD, nor is UD concerned with meanings. However, a similar reasoning can be applied by relating the expression to a more general pattern captured by the part-of-speech variables ADP and NOUN. The normal relation assigned to an adposition in UD in front of a noun is \textit{case} and the structure of the pattern can be captured as for other prepositional phrases as shown in Figure 1. Now we capture the syntactic structure of these expressions reasonably well. However, the information that we are dealing with a fixed expression has been lost. In the current UD framework we cannot say both at the same time. In the wording of (Gerdes and Kahane, 2016) the framework has created a catastrophe.

![Figure 1: Two competing analyses of a fixed MWE, one as syntactically transparent and another as fixed.](https://universaldependencies.org/u/dep/fixed.html)

Moreover, the pattern is similar to that of an adverbial expression consisting of a preposition and a non-inflected noun such as \textit{på lördag} 'on Saturday', and \textit{i morgon} 'tomorrow'. Yet another similar structure employs rest morphemes such as \textit{i går}, 'yesterday' and \textit{i fjol}, 'last year'. Generalising further we can observe that other parts-of-speech such as adjectives can follow a preposition in expressions such as \textit{inom kort}, 'shortly'. In UD we could view all of these as specializations of a common general pattern, ADP + ANY\(^2\).

2.3. More on \textit{fixed} in UD-treebanks

As stated in the introduction, \textit{fixed} is only one of the three relations used for MWEs in UD. These relations have different properties, however. The \textit{compound}-relation can go both to the left and the right and be embedded under a different \textit{compound}-relation. This is not the case for \textit{fixed} and \textit{flat}; they are headless in principle but have the leftmost part as the head by default. Moreover, a dependent of \textit{fixed} or \textit{flat} can’t have dependents of its own. Another UD relation with the same property is \textit{goeswith}, which is primarily used for tokens that have been split accidentally. Structurally \textit{fixed}, \textit{flat} and \textit{goeswith} can all be regarded as the same relation, just labelled differently for complementary information.

A special feature of \textit{fixed}, according to its description on the UD web\(^3\), is that it should be restricted to the most grammaticalized cases and be treated as a closed class. It is recommended that language-specific documentation is developed where the expressions for which \textit{fixed} is applied are listed. The main reason for this is to enforce annotation consistency across treebanks in a way that can be validated automatically. This is definitely a worthy aim as the variation in its use is quite considerable. See Table 3 for figures on \textit{fixed} in a sample of UD Treebanks, version 2.13. It can be noted that there are differences even for treebanks sharing the same language. In fact, some treebanks not shown in the table, like the Norwegian ones and UD_German-HDT do not use \textit{fixed} at all. This shows that recommendations are motivated. It is likely that the differences are not due to language differences but to different annotation principles.

There are published lists only for a few languages, including English and Finnish. The English list has some 40 items, Finnish has around 90. The number of fixed expressions in the largest Finnish treebank is larger, however.

The idea to restrict fixed MWEs in UD to a smaller group raises the question how well it aligns with the notion of a fixed MWE as characterized in general works on the topic such as (Baldwin and Kim, 2010). Is it actually possible to find general criteria that could restrict the application of \textit{fixed} in a principled way? This is investigated in Section 4.

3. Dataset and annotation

The main empirical data for the analysis are taken from the two Swedish UD treebanks UD_Swedish-Talbanken and UD_Swedish-Lines of version 2.13. In addition, we have looked at the list of proposed

\(^{2}\)Instead of ANY we could specify a disjunction of UPOS categories.

\(^{3}\)https://universaldependencies.org/u/dep/fixed.html
Table 3: Usage of fixed in a sample of UD treebanks. The column In TB shows the number of all tokens in the treebanks that carry fixed as their dependency.

<table>
<thead>
<tr>
<th>Treebank</th>
<th>Listed</th>
<th>In TB</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>UD_Dutch-Alpino</td>
<td>-</td>
<td>1161</td>
<td>2.75</td>
</tr>
<tr>
<td>UD_English-EWT</td>
<td>44</td>
<td>40</td>
<td>0.50</td>
</tr>
<tr>
<td>UD_English-GUM</td>
<td>44</td>
<td>44</td>
<td>0.64</td>
</tr>
<tr>
<td>UD_English-LineS</td>
<td>44</td>
<td>117</td>
<td>1.06</td>
</tr>
<tr>
<td>UD_Finnish-FTB</td>
<td>90</td>
<td>198</td>
<td>0.66</td>
</tr>
<tr>
<td>UD_Finnish-FTB</td>
<td>90</td>
<td>27</td>
<td>0.37</td>
</tr>
<tr>
<td>UD_French-Rhapsodie</td>
<td>-</td>
<td>70</td>
<td>2.62</td>
</tr>
<tr>
<td>UD_French-Sequela</td>
<td>-</td>
<td>82</td>
<td>1.45</td>
</tr>
<tr>
<td>UD_Icelandic-IcePaHC</td>
<td>-</td>
<td>20</td>
<td>0.14</td>
</tr>
<tr>
<td>UD_Icelandic-Modern</td>
<td>-</td>
<td>2</td>
<td>0.05</td>
</tr>
<tr>
<td>UD_Italian-ISDT</td>
<td>-</td>
<td>79</td>
<td>0.66</td>
</tr>
<tr>
<td>UD_Italian-TWITTIRO</td>
<td>-</td>
<td>23</td>
<td>0.55</td>
</tr>
<tr>
<td>UD_Swedish-LinES</td>
<td>-</td>
<td>117</td>
<td>1.59</td>
</tr>
<tr>
<td>UD_Swedish-Talbanken</td>
<td>-</td>
<td>392</td>
<td>3.12</td>
</tr>
</tbody>
</table>

Table 3: Usage of fixed in a sample of UD treebanks. The column In TB shows the number of all tokens in the treebanks that carry fixed as their dependency.

*English fixed expressions*.

Together the two Swedish treebanks have 439 different MWEs annotated with fixed. Of these 71 are common to both treebanks, and 216 are hapaxes. For a few common MWEs, such as som om, 'as if', and mer än, 'more than' the two treebanks have made opposite decisions. Yet, the large majority satisfies the loose criterion of being multword sequences that behave as function words, adverbs, or are special in some other way. As the treebanks are not very big we can safely assume that there are many more expressions that satisfy the same tolerant criteria as those in the treebanks. To compare, Wikipedia has 649 expressions listed under the label Swedish idioms and a recent dictionary of Swedish idioms (Luthman, 2020) contains 5000 items, although the majority of these are flexible.

Starting with the properties listed in the definition above (Baldwin and Kim, 2010) other properties were added as cases were found. Previous work on idioms in Swedish such as (Anward and Linell, 1976; Sköldberg, 2004) have largely focused on flexible idioms, but they define various criteria for recognizing MWEs including fixed expressions that we have considered. The expressions in the dataset have also been checked against larger Swedish corpora and concordances generated from the Korp interface on news media. In the end we came up with 13 different properties as listed below. The first two relate to the expression's function and pattern, while the rest focus on some aspect of variation.

- **UPOS tag**: Part-of-speech if regarded as a single UD word, using the UPOS set of tags.
- **Syntactic pattern**: The syntactic pattern is expressed in terms of UPOS tags and regarded as the best generalisation of a more specific CxG pattern.
- **Morpheme status**: Takes the values Roots, Inflected, Foreign, Abbr(eviation) and Special where Special includes rest morphemes and rare (obsolete) inflections.
- **Inflection variation**: Does any part of the expression allow inflectional variants? Yes or No.
- **Internal modification**: Does any part allow one or more modifiers? Yes or No.
- **Synonyms**: Is it possible to replace any part with synonyms? Yes or No.
- **Iterability**: Can a part be repeated? This is rare but occurs for several expressions that signify repeated events: om och om (och om) igen, 'again and again (and again)' Yes or No.
- **Order change**: Can the order among parts be different? Yes or No.
- **Optional part**: Is any part optional, or can an optional part be added? The answer is Yes or No and an example is under det (att), 'while'.
- **Separability**: Can (or must) some part be separated from the rest by other material? Possible values are No, Obligatory, and Optional.
- **Idiom part**: Does the expression mainly occur as part of a longer idiom, in the treebank and generally? If so the value is Yes, otherwise No.
- **Abbreviation**: Does an abbreviated form exist? Yes or No.
- **Collapsibility**: Does a single token equivalent exist? Often this is the result of omitting spaces as in över allt : överaltt, 'everywhere'. Yes or No.

Every expression in the dataset has been described with these attributes. An illustration is given in Table 4 for the expression i våras.

Table 4: Illustration of fixed expression i våras.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPOS tag</td>
<td>Single UD word</td>
<td></td>
</tr>
<tr>
<td>Synonyms</td>
<td>Yes</td>
<td>Has synonyms</td>
</tr>
<tr>
<td>Iterability</td>
<td>Yes</td>
<td>Can be repeated</td>
</tr>
<tr>
<td>Order change</td>
<td>Yes</td>
<td>Order can change</td>
</tr>
<tr>
<td>Separability</td>
<td>Yes</td>
<td>Parts can be separated</td>
</tr>
<tr>
<td>Idiom part</td>
<td>No</td>
<td>Not part of a longer idiom</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>No</td>
<td>No abbreviation</td>
</tr>
<tr>
<td>Collapsibility</td>
<td>No</td>
<td>No single token equivalent</td>
</tr>
</tbody>
</table>

4. Types of fixed MWEs

Given the requirement that fixed expressions in UD should be a restricted closed class we want to
reduce the number of expressions currently annotated with *fixed* in the Swedish treebanks. This entails two main things: identifying criteria that make *fixed* correspond well to a natural class of fixed expressions, and finding alternative dependency analyses for those expressions that are removed.

There are many different types of expressions in the dataset and the available space does not allow us to discuss all of them. We start with one type of variation that may be more common in a Swedish dataset than for other languages, the alternative renderings captured by the property of Collapsibility.

### 4.1. Collapsible MWEs

Swedish language planning authorities are generally quite tolerant towards variation in written Swedish. As a result many multiword expressions have alternative renderings as single tokens or, in case of three-part expressions, two tokens. As UD maintains that tokenisation should follow the orthographic rendering as far as possible, in particular that in-token spaces should be avoided, these expressions pose a special challenge.

In the dataset we find 75 collapsible MWES, which is about 17% of all. The large majority of them has an alternative rendering by omitting spaces. Examples are *till buds :: tillbuds*, 'at hand', *i dag :: idag*, 'today', *över huvud taget :: överhuvud taget*, 'actually'. The share of a certain rendering differs with individual expressions. We have investigated their distribution in two subsets of the Swedish Gigaword Corpus (Redven Eide et al., 2016), news and fiction. The numbers support a division into three different groups, one where the two renderings are about equally common. However, the relevance of this variation lies not so much in the exact proportions but that both renderings occur. A treebank should as far as possible assign the same analysis to both alternatives; they contain the same lexemes, but are just written differently. If spoken they would come out identical. Compare the two renderings below of the same sentence:

(3) *Hon kan när som helst komma i kapp*
(4) *Hon kan närsomhelst komma ikapp*

'She may catch up at any moment'

Given the aversion against token internal spaces in UD one option is to regard the multipart variants as basic and treat the shorter variants as multiword tokens. This solution aligns well with the long-term proposal for modelling synthetic compounds in UD put forward by (Savary et al., 2023b). A drawback is of course that this solution is sofar unseen in any Swedish treebank. Conversely, the existence of the single-token forms may be taken as an argument that they are perceived as single lexemes.

Using multiword tokens for the tokenisation of sentence (4) we would get the tokenisation in Table 5.

| 1 | Hon | hon |
| 2 | kan | kunna |
| 3-5 | närsomhelst_ |
| 3 | när | när |
| 4 | som | som |
| 5 | helst | helst |
| 6 | komma | komma |
| 7-8 | ikapp_ |
| 7 | i | i |
| 8 | kapp | kapp |

Table 5: Proposed tokenisation for single token alternatives to Swedish fixed MWEs.

### 4.2. Syntactic alternatives to fixed

For many of our expressions in the dataset we can find patterns that are shared with other expressions, as in Section 2.2. We may distinguish self-contained patterns from patterns with outward-looking parts. In the first type all included words except one have their head within the pattern. They are easy to provide a syntactic analysis for. With outward-looking parts two words have their heads outside of the pattern. Usually one of them is the last token which may be a preposition, subjunction or conjunction.

**Self-contained expressions.** The most common type of self-contained fixed expression in the dataset consists of a preposition followed by an uninflected noun. There are 66 such prepositional
phrases with examples such as i dag, 'today', i allmänhet, 'in general'. Other two-part expressions beginning with a preposition has a noun in definite form as head, på vippen, 'on the verge', an adjective, på nytt, ' anew', or a pronoun, före detta, 'ex-'. For some the UPOS is even hard to determine på sistone, ' lately', på glänt, 'slightly open', as the token is invariable and only occurs in this special expression. In addition there are three-part expressions with a nominal head of some sort. Taken together prepositional phrases account for almost 40% of the expressions in the dataset.

The syntactic structure of these prepositional phrases need not deviate from compositional phrases of the same patterns, see Figure 2. The fact that the correct UPOS tag for some words may be hard to determine does not prevent the assignment of an appropriate structure either. Moreover, the treatment of prepositions would actually be more consistent if they always are assigned the relation case when followed by a candidate head word.

We note that no more than four of the English MWEs in the list of English fixed MWEs are prepositional phrases, (in order, of course, in case, at least) and see this as support for treating prepositional phrases as non-fixed in the general case.

![Figure 2: Syntactic dependency analysis for expressions beginning with a preposition.](image)

Coordinations can be handled in the same way as prepositional phrases, since their syntactic structure is transparent when a coordinating conjunction is present. The most common type coordinates two adverbs but Swedish also shows instances of coordinated prepositions. Both structures can be viewed as specializations of a more general pattern for coordinations that need not require the two conjuncts to have the same part-of-speech. Thus, a fixed MWE as English by and large could be dealt with in the same way. The proposed structures are shown in Figure 3.

Another common type of pattern has an adverb or adjective as head modified by another adverb. Examples are så pass (stor), 'that (big)' and illa nog, 'bad enough'. They also can be assigned the same structure as their compositional counterparts with the adverb serving as an advmod.

There are also expressions where an adverb seemingly modifies a preposition as in i in, 'into'

![Figure 3: Syntactic dependency analysis for expressions employing coordinations.](image)

or fram till, 'up to'. This is generally forbidden in the UD framework. To avoid annotating the adverb as a modifier we may regard the two parts as independently modifying the head.

Some of the expressions annotated with fixed end with a verb form of some sort most often a participle. Examples are strängt taget, 'actually', allvarligt talat, 'seriously speaking'. Regarded as verb phrases these expressions have obvious syntactic annotations: the participle is the head and the adverb an adverbial modifier. In relation to its context it may be annotated as an adverbial clause, advcl.

Outward-looking parts. A number of two- or three-word expressions have a last part that normally begins a phrase or clause of some sort. This applies to expressions ending in a preposition, a subjunction or one of the comparative conjunctions än, 'than' and som. 'as'.

The most common type of these are three-part sequences starting and ending with a preposition and a noun or nominal word in between. There are 48 expressions of this type in the dataset; examples are på grund av, 'because of', and i samband med, 'in connection with'.

Sometimes the final preposition introduces an optional phrase. An example is med hjälp av, 'with the aid of', where med hjälp can act as an adverb phrase on its own. In those cases it is perfectly reasonable to view the noun in the middle as the head. See Figure 4. If the preposition is required, however, as in på grund av, 'because of', this solution can be questioned. We note though that in the English list of fixed expressions, this type of three-part expression is rare. For example, in spite of is not included so that spite comes out as the head of a noun phrase such as in spite of the problems giving the same structure as in Figure 4.

Expressions ending with a subjunction are also quite common; in the data set we find 9 ending in att, 'that', 2 ending in om, 'if', and 10 ending in som, 'as'. Here a different analysis may be advocated: assigning the different parts separate functions as

---

3For example, the tree with sent-id `weblog-blogspot.com_alaindewitt_20060924104100_ENG_20060924_104100-0031 in en_ewt-ud-train.conllu`
Figure 4: Syntactic relations for the three-part expression med hjälpe av en sko, 'with the aid of'.

mark or case depending on the part-of-speech. For example, in the case of som om and, similarly, as if, one may argue that each of the two parts has a function of its own. The first, som/as indicates that we are dealing with a comparison, the second, om/if that we are dealing with something unreal or assumed. In Swedish, such an analysis gains some support from the fact that the if-clause in certain circumstances can be replaced by a clause without the subjunction:

(5) Han beter sig som vore han ...
'He behaves as were he ...
(6) Han uppför sig som om han var ...
'He behaves as if he were ...

There are eight expressions ending with the comparative conjunction än, 'than'. The majority are introduced by an adjective or adverb in comparative form, such as mer än, 'more than', lägre än, 'lower than' or 'less than'. The comparatives actually all accept modifiers such as mycket, 'much', or lite, 'a little', and for this reason they may not qualify as fixed expressions. Syntactically they can be treated as other expressions with outward-looking parts, letting the conjunction find its head to the right and the whole of that complex be a dependent to the word in the comparative.

In the English treebanks the expressions more than and less than are regarded as fixed when they modify a quantity as in more than 90 percent but not in other contexts. This is a bit awkward as there is no difference in the possibility of adding the modifier much: much more than I have and much more than 90 percent sound equally well-formed.

Similar arguments apply to comparison using the conjunction som, 'as'. They are common both in our dataset and in the English list. But they often share a pattern as the English as many/much/few/little as where virtually any adjective and a number of adverbs may occur in the middle. This indicates that we are dealing with a construction that can be annotated as such with the adjective/adverb as the head.

4.3. Types based on variation

Another basis for grouping expressions is the amount of variation that they admit. For our dataset we may distinguish three groups. At one end there are expressions with no or almost no variation based on the variational attributes that may be called rigid. At the other end we find several expressions that allow inflectional variation, replacement with synonyms and/or internal modification. Those will be called semi-flexible.

Semi-flexible expressions. 57 of the expressions that are currently annotated with the relation fixed can actually be varied enough to be called semi-flexible. This applies to expressions with parts that can be inflected in accordance with their part of speech, be replaced by synonyms, and/or take modifiers. Expressions of this type are

• när det gäller, 'concerning', (inflectional alternatives gällt, gällde, other alternative vad det gäller.

• vem som helst, 'whoever', (modifiers fan, 'the devil', av dem, 'of them', and similarly for other expressions of the same pattern: nä som helst, var som helst, 'whenever', 'wherever'.

• den här, 'this', den där, 'that'. with variants de, den, det, dom for the first part, and här, där for the second part. The second parts are also found after så, sådan, sådant, sådana giving expressions meaning 'like this' or 'like that'.

For these types we argue that they shouldn’t be regarded as fixed MWEs at all because of the amount of variation they accept. Instead syntactic analyses need to be found.

Rigid expressions There are 96 expressions in the dataset that show no variation at all. By including those that are collapsible and/or have an abbreviated form we reach 146 expressions. The most common are som om, 'as if', så att, 'so that', i dag, 'today', därfor att, 'because', på grund av, 'because of', för att', '(in order) to', i stället, 'instead', till exempel, 'for example', all of which occur more than 30 times in the treebanks. We note that in case the English counterparts are MWEs they are listed as fixed for English.

Rigidity may thus be regarded as a characteristic property of expressions to be annotated as fixed.

Also included in this group are expressions from other languages and abbreviations. They are not so numerous but illustrate general types of interest.

---

6In the case of in order to, however, only order is taken as a dependent of in, while to finds its head in a verb to the right.
There are expressions of Latin origin such as *a priori* and *vice versa* and one of English origin, *to date*. Abbreviations include short forms of academic degrees such as *med lic.*, ‘licentiate in medicin’ and common phenomena in academic prose, such as *a. a.*, short for ‘anfört arbete’, and a counterpart to the Latin ‘op. cit.’.

In UD foreign material may be annotated in different ways. If regarded as a borrowing it should be given a suitable UPOS tag and different parts be connected via the relation *flat* (sic!). If regarded as truly foreign each part should have the UPOS X, and, in addition, carry the feature information FOREIGN=Yes. The parts should again be connected via *flat*. With one exception, the expression *ad calendas graecas*, the examples in the dataset are sufficiently common in Swedish to be regarded as borrowings. Depending on their status as functional (*vice versa*) or not (*ad hoc*) they could fit either *fixed* or *flat*.

Abbreviations should be marked by the feature Abbr=Yes. The UPOS tag should reflect the part-of-speech of the abbreviated word. The expanded versions of our two examples both consist of an adjective and a noun so the dependency analysis could use the *amod*-relation rather than *fixed*. See Figure 5.

```
ADJ NOUN
med  lic
ABBR=Yes ABBR=Yes
```

Figure 5: Dependency analysis of the abbreviated title *med lic*, ‘licentiate in medicine’.

**4.4. Candidates for the *fixed* list**

A large number of MWEs currently marked as *fixed* can be excluded as candidates for the list of *fixed* expressions on the basis of their morphosyntactic variation. With a fairly strict criterion on rigidity, not excluding MWEs that are collapsible or can be abbreviated, there are 146 items left. By considering that *fixed* should be restricted to items with function word distribution another seven can also be removed, leaving 139. This is still a large number, however, especially considering that the treebanks only cover a subset of the Swedish MWEs. On the other hand, many of them have a transparent syntactic structure; being self-contained expressions of the kinds described in Section 4.2. By consistently preferring a headed structure when the MWE satisfies such a pattern the numbers can be reduced further. Other types that may be excluded are those where different parts of the MWE can be separately annotated with a dependency to an outside head as was argued in the case of *som om*, ‘as if’ and as is done in English treebanks with many MWEs of the form ‘ADP NOUN ADP’.

As UD is reluctant to see function words as heads the most likely MWEs to put on the list of items annotated with *fixed* are two-word MWEs ending in a preposition or a conjunction. Examples of the first kind are such *in i*, ‘into’ and *rent av*, ‘actually’ and of the second *så att*, ‘so that’, *för att*, *(in order)* to, and *ifråga om*, ‘as regards’. Another set of likely candidates come from adverbial and prepositional MWEs where the head word is not an adverb or a preposition as for *tack vare*, ‘because of’, *till synes*, ‘seemingly’.

**5. Alternative annotations of fixed expressions in UD**

The current UD guidelines on fixed expressions hide their, in many cases, apparent syntactic structure. (Gerdes and Kahane, 2016) have pointed out this as a ‘catastrophe’ problem and makes a proposal to subcategorize syntactic dependencies with a special identifier such as *mwe*. A disadvantage of this solution is that it will proliferate the *mwe* subcategory in the trees. Moreover it annotates the property of being a multiword expression at a single level to the exclusion of other properties that an MWE may have. The proposal in (Kahane et al., 2017) to insert extra lines for fixed expressions such as *top of the range*, which may carry a dependency relation of its own seems more accurate for capturing the lexical character of fixed expressions.

An alternative is to unify the shallow headless relations to one, say *flat*\(^9\), and treat a property such as fixedness with a feature in the same way as is done with foreignness. This would make the annotation similar to that for split words, where the relation *goeswith* is used in tandem with the feature Typo=Yes\(^10\). The features for a fixed MWE could then be applied to its head and be interpreted as including the dependents by default.

This solution would also solve the problem of choosing between *fixed* and *flat*. As shown above the properties of phrases as being fixed, abbreviated, or from a different language sometimes converge. An expression such as *vice versa* could actually be annotated as foreign and fixed at the same time. Then the *fixed* is in conflict with *flat* which is recommended for foreign material. Annotating these properties at the level of features allows them to be combined.

\(^9\)A similar proposal is made in (Savary et al., 2023b) using the label *headless*.

\(^10\)https://universaldependencies.org/u/dep/goeswith.html
A third more radical alternative is not to deal with fixed expressions at all in the current UD format. While there is a need to mark headlessness in the syntactic trees, it is evident that not all kinds of MWEs can be handled as part of UD dependency trees. It is also evident that the current feature annotation is insufficient. It is restricted to words and thus cannot cover subtrees with one feature. The CUP format (CoNLL-U Plus Format) as used by the PARSEME:MWE framework for annotating verbal MWEs allows more complex feature annotation and may be used for many types of MWEs including fixed expressions. This seems to be the future that is also envisioned by (Savary et al., 2023b).

With this alternative appropriate syntactic dependencies need to be found. We have suggested that a Construction Grammar perspective on fixed MWEs is helpful for this purpose. UD has a general principle of a tight relation between UPOS categories and dependency relations. This principle could be extended to UPOS sequences that share enough common features to be related hierarchically to a dependency template as suggested in Section 2.2.

6. Conclusions

We have analysed 439 expressions currently annotated as fixed expressions in Swedish UD treebanks with the aim of producing a well-defined subset that meets UD requirements on the use of the relation **fixed**. We have found a way to reduce this set by closely studying their variational properties and the structural patterns that they share. Although we find a number of rigid MWEs, i.e., expressions admitting no or almost no variation at all, they often have a transparent syntactic structure which is not accounted for when **fixed** is used. And many of them share structure with other MWEs. These structures can be represented in more detail in Construction Grammar frameworks, as we have shown with examples. Although UD does not allow such detail we can nevertheless often generalise the structure to something that can be expressed in UD-terms. Moreover, to capture all kinds of MWEs, whether fixed or flexible, requires a more versatile format than CoNLL-U such as the CUSP-format used for annotating verbal MWEs in the PARSEME:MWE project.

Annotating fixed expressions with a specific relation as part of the dependency structure, as is currently done in UD, prevents the annotation of its syntactic structure. A better solution would be to isolate the structural properties of **fixed**, which it shares with other UD relations such as **flat** and **goeswith**, in a single relation and use features to indicate the character of the expression, something which now is done only for typos.

Another problem we discovered, which may be specific to Swedish, is the large numbers of collapsible MWEs. The best solution we could propose for these, in order to ensure that the dependency analysis would come out the same whether the MWE is collapsed or not is to make use of UD’s provision of multiword tokens.

7. Acknowledgements

I am indebted to Joakim Nivre for valuable comments on an earlier version of the paper and to an anonymous reviewer for detailed critique and pointing me to the article (Savary et al., 2023b). The work has been partially funded by the Swedish Research Council through the Swe-Clarin part of the Swedish National Language Bank.

8. Optional Supplementary Materials

A spreadsheet with our analysis of 439 MWEs currently analysed as fixed in Swedish treebanks is provided as supplementary material.

8.1. Extra space for ethical considerations and limitations

This work is based on open resources and, as far as we can see, pose no ethical problems. A limitation is that it is based on treebank data from one language only and some comparisons with English data. We are certain, though, that the types of problematic multiword expressions discussed here can be found also in other UD treebanks. However, the restriction to one language means that the list of types is likely to be incomplete.

9. Bibliographical References


Synthetic-Error Augmented Parsing of Swedish as a Second Language: Experiments with Word Order

Arianna Masciolini, Emilie Marie Carreau Francis, Maria Irena Szawerna
Språkbanken Text
Department of Swedish, Multilingualism, Language Technology
University of Gothenburg
{arianna.masciolini, emilie.francis, maria.szawerna}@gu.se

Abstract

Ungrammatical text poses significant challenges for off-the-shelf dependency parsers. In this paper, we explore the effectiveness of using synthetic data to improve performance on essays written by learners of Swedish as a second language. Due to their relevance and ease of annotation, we restrict our initial experiments to word order errors. To do that, we build a corrupted version of the standard Swedish Universal Dependencies (UD) treebank Talbanken, mimicking the error patterns and frequency distributions observed in the Swedish Learner Language (SwELL) corpus. We then use the MaChAmp (Massive Choice, Ample tasks) toolkit to train an array of BERT-based dependency parsers, fine-tuning on different combinations of original and corrupted data. We evaluate the resulting models not only on their respective test sets but also, most importantly, on a smaller collection of sentence-correction pairs derived from SwELL. Results show small but significant performance improvements on the target domain, with minimal decline on normative data.

Keywords: Dependency Parsing, Data Augmentation, Second Language Acquisition, L2 Swedish

1. Introduction and Background

In recent years, off-the-shelf dependency parsers have reached remarkably high performance on standard evaluation sets. This applies to many high and medium-resourced languages, including Swedish. Nonstandard language, however, still poses significant challenges. In a study on dependency parsing of learner English, Huang et al. (2018) showed that the tools available at the time were not robust to grammatical errors, despite misleadingly high overall accuracy scores. In a more recent study on L2 Swedish (Swedish as a second language), Volodina et al. (2022) note that, dependency parsing is especially problematic for standard tools, even when they perform reasonably well on other linguistic annotation tasks such as part-of-speech tagging.

A notable attempt to address this issue is the error-repairing parser introduced by Sakaguchi et al. (2017), specifically meant for ungrammatical texts. This approach combines parsing with Grammatical Error Correction (GEC). In many contexts, such as Second Language Acquisition (SLA) research, it can however be preferable to analyze learner texts as they are and, in some cases, to compare originals with their normalized versions. We therefore test the more straightforward approach of fine-tuning a Bidirectional Encoder Representations from Transformers (BERT, Devlin et al. 2018) model for dependency parsing on data that resembles our target domain, L2 Swedish.

With an approach loosely inspired by Stymne et al. (2023), we use the MaChAmp (Massive Choice, Ample tasks) toolkit (van der Goot et al., 2021) to fine-tune an array of models on different combinations of a treebank of standard Swedish and an artificially corrupted version of the same dataset. Crucially, the evaluation step involves not only normative data and artificial errors, but also authentic L2 Swedish sentences.

For this first experiment, we restrict ourselves to word order errors. This is out of both principled and practical reasons. On the one hand, as illustrated by the example in Figure 1, it seems reasonable to assume syntax errors to be challenging for a tool that performs syntactic analysis. When it comes to word order errors specifically, this should be especially true for a language with relatively strict word order such as Swedish. At the same time, word order errors appear to be easier to generate and automatically annotate than most other error types: as tokens are swapped without being altered, token-
level linguistic annotation can be easily transferred from a sentence in standard language to its corresponding corrupted version.

2. Data

We utilize three datasets: an L2 Swedish test set, described in Section 2.1, a standard Swedish treebank and an artificially corrupted version of the latter (cf. Section 2.2). Train-dev-test split sizes are outlined in Table 1.

2.1. SweLL

Our target domain data comes from the SweLL Swedish Learner Language corpus (Volodina et al., 2019), a collection of over 500 essays written by learners of L2 Swedish. More specifically, we use SweLL-gold, the manually pseudonymized version of the corpus (Volodina et al., 2022).\(^1\) L1 backgrounds vary, as well proficiency levels, which range from beginner to advanced. Learner texts are paired with correction hypotheses\(^2\) and each error is classified according to the taxonomy discussed in Rudebeck and Sundberg (2021).

For our purposes, the relevant categories are, in decreasing order of frequency, S-Adv (misplaced adverbia), S-FinV (misplaced finite verb), and S-WO, which encompasses all other word order errors. About 15% of SweLL sentences are marked with one of these labels. In the vast majority of the cases, however, word order errors co-occur with other issues, often overlapping in ways that make the former hard to isolate. After filtering out these cases, we were left with a 69-sentence evaluation set. Regrettably, the resulting sentences tend to be shorter than the corpus-wide average.

2.1.1. Linguistic Annotation

While a linguistically annotated version of SweLL is available, it is not manually validated nor does it follow the UD standard. We therefore opted for completely re-annotating our test set. We started by parsing the correction hypotheses with the UDPipe 2 parser (Straka, 2018) using the UD 2.12 model (Straka, 2023) trained on Talbanken (cf. Section 2.2). The first and third authors, both graduate students in Computational Linguistics, manually validated the resulting parse trees with particular attention to the segments that diverged from the corresponding original learner sentences. This manual annotation step only concerned the DEPREL and HEAD columns of the fully-annotated CoNLL-U files obtained from UDPipe 2, as our models are only trained for UD parsing in its strictest sense. To annotate L2 originals, we used an ad-hoc script which transfers token-level annotations from gold-annotated corrections to L2 originals. Each sentence is first rewritten in the vertical format customary for CoNLL-U files. Then, each token is annotated as follows:

- a token ID is assigned sequentially;
- all other fields except HEAD (syntactic head) are copied from the first unused token of the sentence’s correction hypothesis presenting the same word FORM. Such token is then immediately marked as used, to deal with cases where the same word occurs multiple times in the same sentence;
- the HEAD field is assigned the ID of the nearest token in the learner sentence whose FORM matches that of the syntactic head of the corresponding corrected token.

Choosing syntactic heads based on the closest homograph is a heuristic that occasionally produces ill-formed trees. For this reason, we also inspected the results of this processing step and made the necessary manual edits.

2.2. Talbanken

For training, we used the UD 2.12 version of Talbanken, a widely used treebank of written and spoken modern Swedish (Einarsson 1976, Nivre and Smith 2023). Due to MaChAmp not supporting the enhanced UD format, the treebank was preprocessed with the cleanup script provided as part of the toolkit itself. Its training portion was then used to fit our baseline model with no further changes. Mimicking the error patterns observed in SweLL, we also built a corrupted version of such a treebank, which we used in conjunction with the original upon training our specialized models (cf. Section 3).

2.2.1. Corruption Process

Synthetic error generation is a common task in the field of GEC. Closest to this work is the text corruption method described in Casademont Moner and Volodina (2022), which has been used to build a corpus of Swedish sentences presenting verb

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>SweLL</td>
<td>-</td>
<td>-</td>
<td>69</td>
</tr>
<tr>
<td>Talbanken</td>
<td>4303</td>
<td>504</td>
<td>1219</td>
</tr>
<tr>
<td>Corrupted</td>
<td>4303</td>
<td>504</td>
<td>1219</td>
</tr>
</tbody>
</table>

Table 1: Sizes of the training, development and test splits of our datasets in number of sentences.
order errors using L2 Swedish textbooks as a starting point. We propose a simpler but more general method that covers all three classes of word order errors mentioned in Section 2.1 while preserving UD annotation.

From an operational point of view, such an approach resembles that of Şahin and Steedman (2018), who rely on dependency annotation to “rotate” sentences by swapping subtrees around roots. When it comes to misplaced adverbials (S-Adv), subtrees labelled as adverbial modifiers (advmod) or clauses (advcl) are swapped with their syntactic heads (see Figure 2a for an example). S-FinV errors are generated by swapping finite verbs with their subjects (a nsubj- or csubj-labelled subtree\(^3\), cf. Figure 2b). As for S-WO, with a drastic simplification, we always swap two randomly selected adjacent tokens. After each rotation, the IDs of the corrupted sentence are reassigned sequentially and dependency heads adjusted accordingly, thus ensuring the correctness of the annotation for the resulting corrupted tree.

We tried as much as possible to replicate the error distribution observed in SweLL. For each Talbanken sentence, our corruption script tries to generate three different scrambled sentences (one per error category) and chooses one based on its label’s relative frequency in the corpus. Obviously, however, the S-Adv corruption rule cannot be applied to sentences with no adverbials. There are also instances where finite verbs (typically imperatives) lack an explicit subject or, more rarely, where

\(^3\)If the finite verb in question is an auxiliary, we look for the subject of the head lexical verb.

Figure 2: Two corrupted sentences obtained via subtree swapping. The rearranged segments are highlighted in bold; their syntactic heads, acting as pivot elements, are underlined.

<table>
<thead>
<tr>
<th>Name</th>
<th>% Normative</th>
<th>% Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASELINE</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>MIX15</td>
<td>85</td>
<td>15</td>
</tr>
<tr>
<td>MIX50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>SEG10</td>
<td>100 (step 1)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0 (step 2)</td>
<td>100</td>
</tr>
<tr>
<td>SEG20</td>
<td>100 (step 1)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0 (step 2)</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 2: Our models and the data configurations they were trained on.

3. Models

We used the MaChAmp toolkit to fine-tune a BERT model for dependency parsing using the original and corrupted Talbanken datasets in different configurations, summarized in Table 2. MaChAmp simplifies the fine-tuning of language models for a variety of NLP tasks including dependency parsing (van der Goot et al., 2021). It is relatively simple to set up with the desired hyperparameters and allows for the fine-tuning of various contextualized word embeddings. While we do not leverage the toolkit’s multi-task learning functionalities, we have selected it for its ease of use and sequential fine-tuning. We ran the toolkit with the default hyperparameters, with the exception of changing the default model to the monolingual Swedish BERT (Malmsten et al., 2020) and altering the number of epochs in one
of our sequential models (seq10 was only further fine-tuned on 10 epochs of corrupted data, not the default 20).

All in all, we have fine-tuned BERT for Swedish five times, resulting in five final models. The first model we fine-tuned purely on Talbanken, as a baseline (baseline), in order to know what results fine-tuning only on normative data yields. Our first specialized model, mix15, utilized a combination of normative data and synthetic errors that was meant to mimic the relative frequency of this kind of errors in the learner data. In order to see whether increasing that relative frequency would have a detrimental effect on a model, we fine-tuned one with equal parts of normative and corrupted data, mix50. We also experimented with sequential training to further fine-tune the baseline model with 10 and 20 epochs of only corrupted data (seq10 and seq20, respectively), to investigate whether the performance of an existing dependency parser could be improved by retraining it on non-normative language.

4. Evaluation

Model accuracy was evaluated in terms of Labelled and Unlabelled Attachment Scores, LAS and UAS. To check for statistical significance, these were calculated for each parse tree and compared against a baseline trained on standard Talbanken data to determine if the difference in model performance was significant. A paired t-test with a 95% confidence interval and $\alpha = 0.05$ was used with the Bonferroni correction to compensate for multiple tests against the baseline. Both the UAS score and LAS score were tested against the baseline, so it is possible for only one of the scores to be statistically significant. For nearly all cases, with the exception of Seq20 SweLL (Table 4), either both scores or neither were found to be significant.

Performance on target domain data was assessed on the SweLL-derived test set described in Section 2.1. The models were also evaluated on the original Talbanken test set and its corrupted version (cf. Section 2.2). Talbanken was included to assess whether the addition of ungrammatical examples resulted in a performance decline on normative data, while SweLL allowed for comparison of results on actual learner errors. The expectation was to see a substantial performance increase on corrupted Talbanken instances and a smaller improvement on authentic examples. When it comes to normative data, the ideal outcome would be for the fine-tuning on artificial errors to not have any negative repercussions.

**Targeted Evaluation** To further analyze how this method affects word order errors, a more targeted evaluation was performed using a modified version of the SweLL test set. Following Berzak et al. (2016), we assumed tokens belonging to erroneous segments to be more likely to be incorrectly parsed, even though annotation errors might cascade to other parts of the sentences. Errors were isolated from learner sentence-correction pairs by removing tokens preceding and following the diverging segment. Attachment scores were then recomputed on the resulting sentence fragments.\(^3\)

4.1. Results and Discussion

Overall average scores are summarized in Table 3. Performance results suggest that exposure to synthetic word order errors in training has a positive effect on the models' ability to handle the (in-domain) corrupted sentences, matching our expectations. Simultaneously, performance decline on normative data is contained. Addressing the central question of whether improvement on synthetic data transfers to actual learner sentences, a slight positive effect on similar errors in out-of-domain texts can be observed. Smaller performance gains on out-of-domain texts may be attributed to synthetic errors not being sufficiently similar to authentic examples, to differences between training and test domains beyond mere grammaticality, or a combination of the two. It must also be taken into account that the margin of improvement on learner sentences is smaller than on artificial errors. On artificially corrupted sentences, the baseline's performance

<table>
<thead>
<tr>
<th>Talbanken</th>
<th>Corrupted</th>
<th>SweLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAS UAS</td>
<td>LAS UAS</td>
<td>LAS UAS</td>
</tr>
<tr>
<td>Baseline</td>
<td>92.42 94.30</td>
<td>80.20 83.29</td>
</tr>
<tr>
<td>mix15</td>
<td>92.23 94.05</td>
<td>87.96 90.50</td>
</tr>
<tr>
<td>mix50</td>
<td>91.54 93.58</td>
<td>89.59 92.00</td>
</tr>
<tr>
<td>seq10</td>
<td>92.20 94.06</td>
<td>90.47 92.75</td>
</tr>
<tr>
<td>seq20</td>
<td>92.53 94.32</td>
<td>90.95 93.08</td>
</tr>
</tbody>
</table>

Table 3: Overall attachment scores sets for all fine-tuned models. Cells with a grey background indicate that the difference between the scores for the baseline and fine-tuned models is statistically significant.
drops by about 10% for both metrics, while scores stay reasonably high on SweLL. Notably, on the other hand, specialized models perform very similarly on both non-normative datasets. The seq10 model performed best across all test sets except Talbanken.

### 4.1.1. Talbanken

The Talbanken set showed the highest performance overall, with the baseline achieving a LAS of 92.42% and an UAS of 94.3%. This observation is expected, as the models were for the most part trained on the same domain (Talbanken data). Performance with the fine-tuned models generally decreased, but only mix50 and seq10 showed a result that was significantly different compared to the baseline. It appears that exposing the model to atypical word order has little impact on performance for the Talbanken domain.

### 4.1.2. Corrupted Talbanken

Results for the corrupted Talbanken set showed the largest increase in performance compared to the baseline, about an 8 to 10% increase, and the differences were statistically significant.\(^5\) The seq10 and seq20 models showed the biggest improvement, with a 10% increase over the baseline. This confirms the viability of the fine-tuning approach for specialized UD parsers, at least when target domain data is available.

### 4.1.3. SweLL

Most specialized models exhibited small performance improvements against the baseline. However, just the seq10 model’s improvement was significant. Interestingly, the only model that declined in performance, mix15, was the one exposed to a percentage of errors corresponding to the one observed in SweLL-gold, which appears not to be enough to produce a positive effect.

A further encouraging signal comes from the targeted evaluation. When we focus on ungrammatical fragments, we see that the performance gap between the baseline and all the specialized models widens (cf. Table 4). Not only does this confirm the baseline’s vulnerability to grammatical errors, but it also suggests that the models are learning something about non-normative word order, rather than just exhibiting a general improvement due to exposure to additional training data.

### 5. Conclusions and Future Work

We generated synthetic word order errors and used them to fine-tune a number of dependency parsers.

\(^5\) \(p=0.0000000000000022\), per paired t-test.

<table>
<thead>
<tr>
<th></th>
<th>LAS</th>
<th>UAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>82.80</td>
<td>86.02</td>
</tr>
<tr>
<td>mix15</td>
<td>84.41</td>
<td>89.25</td>
</tr>
<tr>
<td>mix50</td>
<td>87.10</td>
<td>90.32</td>
</tr>
<tr>
<td>seq10</td>
<td>87.10</td>
<td>89.78</td>
</tr>
<tr>
<td>seq20</td>
<td>86.02</td>
<td>89.78</td>
</tr>
</tbody>
</table>

Table 4: Attachment scores for the targeted evaluation on the SweLL-based test set. Cells with a grey background indicate that the difference between the scores for the baseline and fine-tuned models is statistically significant.

We evaluated them on (1) normative data, (2) synthetic error data, and (3) authentic L2 sentences containing errors of the same kind. The improvement on the latter was small, but significant. No substantial decrease in performance on normative data was observed, which suggests this is a promising method to increase parser robustness.

Future work aimed at achieving a more significant performance increase on target domain data should revolve around improving the corruption pipeline, especially when it comes to S-WO errors. The choice of material to corrupt is also important. In fact, we believe that applying our method to sentences from a domain closer to learner essays could result in better performance. It would also be beneficial to either have a larger test set or compare models in terms of multi-run averages in the future in order to more confidently assert that the differences between fine-tuning methods are not accidental. Other interesting possibilities are trying to run a hyperparameter search for at least some of the models and seeing how a multilingual model compares to the monolingual one we employed.

To ensure that our method is actually applicable to learner data in a more general sense, a possibility is to add one more test set where word order errors co-occur with other issues. Finally, a central question is to what extent our approach can be generalized to handle other kinds of errors (such as missing or redundant tokens, lack of agreement, etc.), and, most importantly, whether it can be adapted to handle sentences with multiple errors of various kinds.

### 6. Data and Code

The SweLL-derived test set and code are available at [github.com/spraakbanken/seapass](https://github.com/spraakbanken/seapass).

### 7. Ethical Concerns

While linguistic data can contain personal information, raising privacy concerns, neither of the datasets used in this experiment is likely to leak sen-
sitive information. Aside from its age, Talbanken consists of texts from genres like textbooks and articles, which are unlikely to contain information that should not be shared. As for SweLL-gold, a corpus that is both more recent and more likely to contain sensitive information due to its domain (L2 learner essays), all of the elements considered to be sensitive have been replaced with pseudonyms during corpus creation, and appropriate written consent had been obtained during the data collection step. Therefore, we consider the privacy risks of using these two datasets to be minimal.

8. Acknowledgments

SweLL-gold is part of Språkbanken, the Swedish national research infrastructure. Furthermore, the experiments presented in this paper are preliminary to the release of a UD version of such corpus, which will in turn enrich the existing infrastructure. On this basis, this research is supported by Nationella Språkbanken, funded jointly by the Swedish Research Council (2018–2024, contact 2017-00626) and the ten participating partner institutions.

9. Bibliographical References


10. Language Resource References


The Vedic Compound Dataset

Sven Sellmer, Oliver Hellwig

Institute of Oriental Studies, Adam Mickiewicz University Poznań
Department of Comparative Language Science, University of Zürich
sven@amu.edu.pl, oliver.hellwig@uzh.ch

Abstract

This paper introduces the Vedic Compound Dataset (VCD), the first resource providing annotated compounds from Vedic Sanskrit, a South Asian Indo-European language used from ca. 1500 to 500 BCE. The VCD aims at facilitating the study of language change in early Indo-Iranian and offers comparative material for quantitative cross-linguistic research on compounds. The process of annotating Vedic compounds is complex as they contain five of the six basic types of compounds defined by Scalise and Bisetto (2005), which are, however, not consistently marked in morphosyntax, making their automatic classification a significant challenge. The paper details the process of collecting and preprocessing the relevant data, with a particular focus on the question of how to distinguish exocentric from endocentric usage. It further discusses experiments with a simple ML classifier that uses compound internal syntactic relations, outlines the composition of the dataset, and sketches directions for future research.

Keywords: Compounding, Sanskrit, Vedic, Dependency annotation

1. Introduction

Since the beginnings of modern linguistics, Sanskrit compounds have played a special role in research on compounding (see, e.g., Wujastyk, 1982), which is reflected by the fact that even some terms of the Indian grammatical tradition have entered current linguistic terminology (see Tab. 1). Sanskrit – especially its oldest form, known as Vedic, which was used from ca. 1500–500 BCE – is also of fundamental importance for Indo-European and cross-linguistic studies. Up until now, there exists a substantial collection of annotated compounds for classical and Neo-Sanskrit. Many of these annotations, originating from works composed in the 19th and 20th c. CE, offer, however, only limited insights for historical linguistics due to their relatively recent composition. The Vedic Compound Dataset (VCD) introduced in this paper is the first resource to provide annotated compounds from Vedic, making it particularly well-suited for studying language change in the formative period of Sanskrit. Concerning annotation, Vedic compounds constitute an interesting challenge. As will be discussed in Section 4, they contain five of the six basic types of compounds that are used in Guevara and Scalise 2009. In addition, neither the compound internal relation between the words constituting them (see the examples in Section 3) nor the relation between a compound and the rest of the sentence are consistently marked in morphosyntax, which poses a challenge to their automatic classification. Over the past decade, several attempts at automatic classification of classical Sanskrit compounds have been undertaken. While Krishna et al. (2016) obtain 74% F-score for a dataset with four coarse compound categories by applying a Random Forest classifier to a set of manually defined linguistic markers, Sandhan et al. (2019) achieved a comparable F-score of 73% using an approach that combined a recurrent architecture with static word embeddings, bypassing the need for extensive feature engineering. Most recently, Sandhan et al. (2022) argued that compound classification needs to take syntactic properties of the surrounding text into account. They therefore combined compound classification with morphosyntactic tagging and dependency parsing in a joint learning task. Using a deep learning architecture with contextualized word embeddings, they report an F-score of 85.7% for coarse compound classification.

While these contributions have significantly advanced automatic Sanskrit compound classification, the present study did not use these systems for compound annotation for several reasons. Firstly, previous studies used classical Sanskrit data, but our focus is on Vedic compounds. The significant lexical differences between Vedic and classical Sanskrit can make applying these systems...
to Vedic texts problematic. Secondly, while an F-score of 85.7% is remarkable, it does not meet the high standards required for creating a reference dataset. Thirdly, the compound categories employed by these studies do not encompass all categories proposed by Bisetto and Scalise, limiting their applicability to our research. In what follows, we will present how we collected and prepared our data (Sec. 3 and 4), devoting particular attention to the recognition of their endocentric-exocentric dimension, and discuss experiments with a simple ML classifier. We will then discuss the composition of the dataset (Sec. 5) and draw conclusions for future research (Sec. 6).

3. Data collection

Our data is derived from two closely linked resources. The Digital Corpus of Sanskrit (Hellwig, 2010–2024) offers lexical and morphosyntactic annotations for Vedic and classical Sanskrit texts. Within the DCS, compounds that have a non-lexicalized reading (see below) are divided into their constituent parts. For instance, the coordinate compound indrāgni- ‘Indra and Agni’ is separated into the words indra- and agni-, each with its own morphosyntactic information. This preprocessing of the source data makes the identification of compounds significantly easier. The Vedic Treebank (VTB, Hellwig et al. 2020), containing approximately 32,000 sentences, supplements the DCS with a layer of Universal Dependencies (UD) annotations. The syntactic annotation of the VTB was carried out by a team of experts, who employed enhanced annotation guidelines (see Hellwig et al., 2023).

The standard UD guidelines offer only limited possibilities for a differentiated treatment of compounds, which is unsatisfactory in view of the versatile role of compounds as an interface between syntax and lexicon and especially of the fact that Vedic compounds – like Sanskrit compounds in general (Lowe, 2015) – contain various syntactic structures, which tend to become diachronically increasingly complex. Therefore, the team extended the annotation guidelines (Biagetti et al., 2020) with the aim of enabling the annotator to make explicit the internal syntactic structure of a compound in the same way as UD labels show the relations obtaining between the words in a sentence. For instance, the compounds indra-agni-4 ‘Indra and Agni’ (as a pair), deva-loka- ‘world of the gods’, and ardhamaśa- ‘half-month’ are annotated as follows:

\[
\begin{array}{c}
\text{compound:coord} \\
\text{nmod} \\
\text{amod}
\end{array}
\]

\[
\text{indra agni-}, \text{deva loka-}, \text{ardha måsa-}
\]

\[
\text{Indra Agni god world half month}
\]

The information – not immediately obvious in the latter two examples – that a word is a non-final member of a compound was incorporated into the VTB via the “Compound” feature (to be distinguished from the label compound, which is only used for coordinate compounds in the VTB). Compounds can include a limited number of particles and adverbs in addition to nominal forms (e.g. sa-ratha-, lit. ‘with-chariot’, i.e. “having a chariot”). Most of these indeclinables do not exist as standalone words. Since they constitute a closed lexical set, they can be directly integrated into compound detection. Adverbs that are part of compounds but do not belong to this closed set (e.g. su- ‘well’, which also occurs independently) were addressed individually during annotation.

To detect compounds in the VTB, we conducted a scan of the VTB’s conllu file for instances of the “Compound” feature and built compounds by tracing the syntactic arcs of the non-terminal members until we reached an inflected word form, which had to be the terminal member. In the example deva-loka-given above, deva- is labeled with the Compound feature in the VTB. Following the arc with the nmod label, we arrive at loka- which has an inflectional ending in a real world case and thus must constitute the terminal member of the compound.

The VCD is specifically designed to contain only two-word compounds. Apart from time restrictions, this focus is due to the fact that longer compounds of \( n \) words can typically be analyzed as multi-level mixed types consisting of \( n - 1 \) elements. Furthermore, the oldest Vedic texts predominantly contain two-word compounds (see e.g. Macdonell, 1910, 143). By limiting our data selection to these short compounds, we ensure that our data covers the entire Vedic period. We equally did not include compounds that were identified as lexicalized by the annotator of the DCS. These compounds are typically technical terms. For instance, the term agnihotra- is a compound of the words agni- ‘sacri-ficial fire’ and hotra- ‘sacri-ficial libation’. However, an agnihotra- is not merely a ‘libation into the sacri-ficial fire’, but a specific type of such a libation (see e.g. Renou, 1953). Despite their semantic transparency, such lexicalized compounds are annotated as single words in the DCS and are not identified as compounds in its dictionary. As a result, we lack access to information indicating that agnihotra- is a compound, as well as its internal syntactic relation. The integration of such lexicalized compounds into the VCD remains an open issue for future research.

\[\text{See https://universaldependencies.org/docs/en/dep/compound.html.}\]

\[\text{4For convenience, all euphonic (‘sandhi’) changes have been removed in this paper.}\]
4. Compound classification

Among the various possible classification schemes for compounds, we adopted the version proposed by Scalise and Bisetto (2005), for three reasons: 1. it is not only well-suited for Sanskrit (as can be seen in Biagetti, 2024) but also for many other languages; 2. it has already been employed in cross-linguistic studies (Scalise and Guevara, 2006; Guevara and Scalise, 2009), so that it facilitates the reusability of our dataset in such contexts; 3. it is convenient due to its conceptual simplicity, as opposed to its refined version in Scalise and Bisetto, 2009, which was too finegrained for our time budget.

This scheme has two-dimensions, as exemplified by the rows and columns of Tab. 1. Firstly it classifies compounds as endo- vs. exocentric, where an exocentric compound is understood as one that is not a hyponym of its formal head (see Bauer, 2017, 37, e.g., a redneck is not a kind of neck; for other definitions see Bauer, 2008 and Moyna, 2019). In Sanskrit grammar, these are called bahuvrīhis: compounds that, as a whole, are (sometimes secondarily nominalized) adjectives though their final member is a noun. Secondly, it encodes the relation between the first and the final member, which may either be coordinate (i.e., dvandvas in the strict sense, Ralli 2019), subordinate; A. = attributive); indigenous terms in brackets.

Table 1: Compound classification according to Scalise and Bisetto 2005 (C. = coordinate; S. = subordinate; A. = attributive); indigenous terms in brackets.

<table>
<thead>
<tr>
<th>Internal label</th>
<th>Compound type</th>
</tr>
</thead>
<tbody>
<tr>
<td>compound:coord</td>
<td>coordinate</td>
</tr>
<tr>
<td>nmod, obj, obl, iobj</td>
<td>subordinate</td>
</tr>
<tr>
<td>advmod, amod, nummod, acl, det, xcomp, nmod:appos, advcl</td>
<td>attributive</td>
</tr>
</tbody>
</table>

Table 2: Internal labels and the dimension coordinate/subordinate/attributive.

2. Coordinate compounds being endocentric by default in Sanskrit, subordinate und attributive compounds were then classified under the aspect of their exocentricity.

2a. In about 1/5 of the cases, this can be done automatically, namely, where a mismatch between the gender of the compound and the gender of its final member as an independent noun can be observed. For instance, the compound aśva-mukha-can be either endocentric (‘face of a horse’) or exocentric (‘horse-faced’). Now, mukha- ‘face’ is a neutral noun, so wherever aśva-mukha- features a non-neuter ending it must refer as an adjective to a masculine or feminine noun (e.g., aśva-mukhaḥ rākṣasah ‘a horse-faced demon’) and so be an exocentric compound.

2b. Further, a sizeable subgroup of exocentric compounds (ca. 600 tokens) could be classified on the basis of their morphology: the so-called root or synthetic compounds with a verbal root noun as final member are always exocentric (Scarlata, 1999); e.g., prathama-ja ‘first-born’, from ṣan ‘to be born’. Detecting them could not be fully automatized as there is no appropriate POS tag in the DCS flagging them as verbal roots.

2c. In the remaining ca. 1,700 cases, the decision to classify a given compound as exo- or endocentric could only be made by a human expert on the basis of its use in the actual context. Dictionary information could be used in cases in which the translation indicated exclusively exo- or endocentric usage. But such hints were not available for all compounds, and in addition turned out to be not always reliable. Opposite to what one may expect, the UD label of the final compound member did not allow to decide between exo- and endocentric usage. For example, in their prototypical role as adnominal modifiers, bahuvrīhis are linked by acl to their referents (Biagetti et al., 2020, Sec. 5

In accented texts, also the location of the accent in a compound often is indicative of exocentricity (Wackernagel, 1905, § 113), but this information was not available to us as it is lacking in the DCS.
However, even this label cannot serve as a reliable indicator of exocentricity, because it also appears with endocentric compounds, for instance, in relative clauses. In addition, only about 30% of all bahuvrihi s are used as adnominal modifiers, as they are, for instance, often substantivized and function as independent nouns. As a consequence, the annotation of these 1,700 compounds had to be done manually, which turned the present step into the most time-consuming one.

The description of the annotation process suggests that many decisions are rule-based, i.e., can be made based on the internal and external syntactic relations of compounds and their morphosyntactic information. We hypothesized that a simple classification algorithm with access to the syntactic gold information of the VCD could learn these rules. To test this hypothesis, we implemented a multinomial regression model. The predictors for this model include the aforementioned compound-internal and external syntactic labels, as well as the part-of-speech tags and lemmata of the two words constituting a compound. The model is trained to predict which of the five classes in the scheme of Bisetto and Scalise (Table 1) a compound belongs to. The results of a tenfold cross-validation (see Table 3) show that the system achieves F-scores above 80%, even for complicated classes that involve decisions between endo- and exocentric use. As the F-scores of the two ablation tests in columns 5 and 6 of Table 3 (-I: no compound internal syntactic labels; -O: no outer labels) indicate, this success is mainly due to the availability of compound-internal syntactic relations from the VTB. While ignoring the labels that connect compounds with the rest of the sentence and thus indicate their syntactic function (-O) keeps the F-scores largely unchanged, ignoring their inner syntactic labels (-I) leads to substantially lower F-scores for three out of five types. Specifically, the low F₁-score for coordinate endocentric compounds likely results from the fact that they are not distinguished by POS information from subordinate compounds, but occur with much lower frequency (see Tab. 5.). These findings can inform future research in automatic compound classification.

### 5. Composition of the dataset

The VCD contains almost 7,000 two-word compounds together with information on morphology, internal and external syntactic relations, chronology, and Vedic subtraditions. A few plots and tables may serve to give an overview of the composition of our dataset. Tab. 4 lists the most frequent compound internal labels in the VCD. It thus gives insights into the syntactic processes active during compounding and so can serve as a starting point for cross-linguistic comparison and for the construction of fine-grained semantic frames. Tab. 5 shows the distribution of the tokens over Scalise and Bisetto’s classification. The numbers confirm the general cross-linguistic observations in Guevara and Scalise 2009, 118–119, that in terms of frequency S. > A. > C. Regarding the endo-/exocentric distinction, exocentric compounds make up 41.8% of all compounds in the VCD. This is a remarkably high percentage compared with the statistics in Scalise et al. 2009, where this ratio ranges from 8.4% (Germanic languages) to 35.4% (Romance languages). The ratio for Vedic gets even higher when the diachronic dimension of our dataset is taken into account. As can be seen in Fig. 1, right, it drops from an extreme ratio of 72.4% in the archaic Rig Vedic period, a figure reminiscent of what Bauer (2008, 68) reports for some African and Australian languages, to 30.3% in the late Sūtras. Notably, this development runs counter to the general rise in compound usage, as shown in Fig. 1, left.

<table>
<thead>
<tr>
<th>Type</th>
<th>Pₐₗ</th>
<th>Rₐₗ</th>
<th>Fₐₗ</th>
<th>F₋₁</th>
<th>F₋₀</th>
</tr>
</thead>
<tbody>
<tr>
<td>attrib/endo</td>
<td>81.8</td>
<td>86.4</td>
<td>84.0</td>
<td>80.2</td>
<td>79.9</td>
</tr>
<tr>
<td>attrib/exo</td>
<td>81.0</td>
<td>80.9</td>
<td>80.9</td>
<td>74.8</td>
<td>80.0</td>
</tr>
<tr>
<td>coord/endo</td>
<td>97.6</td>
<td>98.6</td>
<td>98.1</td>
<td>29.6</td>
<td>98.1</td>
</tr>
<tr>
<td>subord/endo</td>
<td>91.4</td>
<td>92.3</td>
<td>91.8</td>
<td>82.3</td>
<td>90.5</td>
</tr>
<tr>
<td>subord/exo</td>
<td>87.2</td>
<td>81.1</td>
<td>84.1</td>
<td>80.0</td>
<td>78.8</td>
</tr>
</tbody>
</table>

Table 3: Results of the multinomial classifier for compound types, 10-fold cross-validation. All: all predictors, -I: no internal syntactic labels, -O: no outer labels.

<table>
<thead>
<tr>
<th>Deprel</th>
<th>#Tok.</th>
<th>Deprel</th>
<th>#Tok.</th>
</tr>
</thead>
<tbody>
<tr>
<td>nmod</td>
<td>2260</td>
<td>nummod</td>
<td>574</td>
</tr>
<tr>
<td>advmod</td>
<td>1089</td>
<td>obl</td>
<td>460</td>
</tr>
<tr>
<td>amod</td>
<td>800</td>
<td>acl</td>
<td>191</td>
</tr>
<tr>
<td>obj</td>
<td>721</td>
<td>det</td>
<td>189</td>
</tr>
<tr>
<td>compound:coord</td>
<td>632</td>
<td>iobj</td>
<td>26</td>
</tr>
</tbody>
</table>

Table 4: Most frequent compound-internal dependency relations in the VCD.
6. Conclusion

Up until now, the diachronic, geographical and sociolinguistic development of Vedic literature remains incompletely understood (Witzel, 1997). The compounds collected in the VCD, showing clear diachronic trends regarding their endo-/exocentric dimension (see Fig. 1), thus provide valuable data for gaining deeper insights into the linguistic developments during this period as well as for time-stamping Vedic texts (Hellwig, 2024). They can further prove fruitful for comparative studies in an Indo-European and cross-linguistic framework, as they contain data about one of the earliest attested Indo-European languages.

The rule-based parts of collecting the dataset were comparatively straightforward, but to distinguish between exocentric and endocentric compounds of the attributive and subordinate types turned out to be a time-consuming process. It is important to note that this work would have been unnecessary if such a distinction could be directly established on the basis of the UD labels. It would be therefore helpful to add an appropriate UD sublabel to, e.g., nmod and amod, to indicate bhāuvṛīhi in various languages. This would be a small extra effort, because for a human expert annotating a whole sentence it is usually evident whether a given compound is exocentric. It is to be expected that DL dependence parsers will then be able to process these annotations and to determine the exocentricity of compounds with high precision. This would be a highly desirable outcome for the research on compounds in general, as their exocentric/endocentric dimension is of fundamental importance. In addition, due to the general tendency of exocentric compounds for having a metonymic meaning (Bauer, 2008; Barcelona, 2008), such a sublabel would also be relevant for metonomy recognition.

7. Ethical considerations

We are not aware of any ethical issues arising from the composition or use of our data set.

8. Limitations

Four limitations of our dataset should be mentioned. Firstly, it must be understood that – though of considerable size for an ancient language – it is based on only about 35% of the extant Vedic literature – nevertheless, its chronologically balanced composition and the wide variety of texts it draws on make it useful for quantitative linguistic studies. Secondly, as discussed on p. 2 above, we did not consider compounds that were treated as lexicalized in the DCS. Thirdly, for the reasons explained on p. 2, we restricted ourselves to two-word compounds for the time being. We plan to overcome these limitations in future versions of the VCD. Finally, it should be noted that the POS tags taken over from the VTB are not completely reliable. In the VCD, they have been manually corrected in a number of instances, but not in the form of a systematic revision.

9. Acknowledgements

We would like to express our gratitude to three anonymous reviewers, who made a number of very helpful remarks and suggestions.

Research for this paper was funded by the German Federal Ministry of Education and Research, FKZ 01UG2121.

10. Bibliographical References


Oliver Hellwig. 2010–2024. DCS - The Digital Corpus of Sanskrit.

Oliver Hellwig. 2024. To compound or not to compound? A diachronic Bayesian analysis of compounds and equivalent constructions in Vedic Sanskrit. Indogermanische Forschungen, 129.


A Universal Dependencies Treebank for Gujarati

Mayank Jobanputra1, Maitrey Mehta2∗, Çağrı Çöltekin3
1Department of Language Science and Technology, Saarland University
2Kahler School of Computing, University of Utah
3Department of Linguistics, University of Tübingen
mayank@lst.uni-saarland.de, maitrey@cs.utah.edu, ccoltekin@sfs.uni-tuebingen.de

Abstract
The Universal Dependencies (UD) project has presented itself as a valuable platform to develop various resources for the languages of the world. We present and release a sample treebank for the Indo-Aryan language of Gujarati – a widely spoken language with little linguistic resources. This treebank is the first labeled dataset for dependency parsing in the language and the script (the Gujarati script). The treebank contains 187 part-of-speech and dependency annotated sentences from diverse genres. We discuss various idiosyncratic examples, annotation choices and present an elaborate corpus along with agreement statistics. We see this work as a valuable resource and a stepping stone for research in Gujarati Computational Linguistics.

Keywords: low-resource languages, universal dependencies, Gujarati

1. Introduction
The Universal Dependencies (UD) project (Nivre et al., 2016; de Marneffe et al., 2021) offers cross-linguistically consistent annotations for dependency treebanks, part-of-speech, and morphological features. The ever-expanding language base under the UD umbrella ensures that similar language patterns can be dealt with consistently when working with a new language. Further, language-specific features are brought to the fore for discussion. As a result, UD becomes the most fundamental of resources to be developed for a particular language.

Gujarati is an Indo-Aryan language originating from the western Indian state of Gujarat. The language is widely spoken by over 56 million speakers (Eberhard et al., 2022) and is one of the 22 languages with official status in India. Yet, the Gujarati Computational Linguistics community is still in its infancy. Joshi et al. (2020) classify Gujarati in the “Scraping-Bys” category (category 1) in their taxonomy indicating a scant availability of labeled datasets. Basic resources such as part-of-speech taggers, and named entity recognizers are not readily available. Hence, a dependency treebank in such a language can have a wide-reaching impact.

On the other hand, the UD community has already produced a handful of treebanks in various Indo-Aryan languages. As a result, we are equipped with resources in related languages like Marathi (Ravishankar, 2017), Hindi (Bhat et al., 2017; Zeman et al., 2017), and Punjabi (Arora, 2022). Such resources are of value while constructing a sample Gujarati treebank.

The benefits of building a sample Gujarati treebank are four-fold:

a) It presents as a valuable resource for the development of linguistic tools and resources in a low-resource language, i.e., Gujarati.

b) Gujarati uses a unique eponymous script that is not yet represented in the UD project. This can be especially valuable for future researchers interested in building resources for lesser-resourced languages such as Kutchi, and Bhill that also use the Gujarati script.1

c) It ensures annotation paradigms in similar contexts are adhered to and helps point out any discrepancies in existing treebanks.

d) We can point out some new idiosyncratic phenomena that might be Gujarati-specific, or missed by earlier works.

The above-mentioned reasons motivate us to propose a sample dependency treebank for Gujarati: GujTB.2 In the subsequent sections, we explain the selected corpora, statistics and highlight some interesting discussion points encountered.

2. The Dataset
In this section, we provide details of the annotated corpora and the annotation process.

Corpora. We investigated available corpora that include Gujarati text such as IndicCorp (Kakwani et al., 2020) and Samanantar (Ramesh et al., 2022). However, we observe that these datasets majorly contain news and other formal

1https://www.omniglot.com/writing/languages.htm
2Code & Data available at: https://github.com/UniversalDependencies/UD_Gujarati-GujTB
texts. Hence, we annotate a total of 187 sentences taken from diverse sources like Samanantar (news), UD Cairo (short),\textsuperscript{3} Gujarati translations (from Mehta and Srikumar, 2023) of the French novella – *Le Petit Prince* (fiction) (The Little Prince, de Saint-Exupéry, 1943), and a Gujarati grammar book (grammar) (Raimond, 2004).

**Annotation Process and Agreement.** Two of the paper authors\textsuperscript{4} annotated this dataset. The annotations were created separately, and followed by an initial correction phase to fix any obvious errors. A hundred-sentence subset of annotations was considered for the inter-annotator agreement (IAA) study.\textsuperscript{5} The IAA for the part-of-speech (POS) tags is 99.87 (Cohen’s $\kappa$). The head selection agreement is 99.44% and the relation agreement on the heads that matched is 99.88 (Cohen’s $\kappa$). The head selection agreement is the proportion of dependents assigned the same head by both annotators (similar to the unlabeled attachment score).

**Dataset Statistics.** The dataset statistics by genre are given in Table 1. The distribution of POS tags in the corpus is given in Table 2. Furthermore, we provide the statistics regarding dependency relations in Table 3. Notably, our dataset is a representative set of all possible relations in Gujarati.

<table>
<thead>
<tr>
<th>Genre</th>
<th>Sentences</th>
<th>Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>news</td>
<td>93</td>
<td>1159</td>
</tr>
<tr>
<td>short</td>
<td>20</td>
<td>178</td>
</tr>
<tr>
<td>fiction</td>
<td>40</td>
<td>331</td>
</tr>
<tr>
<td>grammar</td>
<td>34</td>
<td>217</td>
</tr>
<tr>
<td>Total</td>
<td>187</td>
<td>1885</td>
</tr>
</tbody>
</table>

Table 1: Data statistics by genre for GujTB.

### 3. Syntactic Relations

In this section, we discuss the many interesting dependency choices. While a large volume of dependency choices such as subjects, object, and light/serial verb constructions follow existing Indo-Aryan literature (Bhat et al., 2017; Ravishankar, 2017; Ojha and Zeman, 2020; Arora, 2022), our goal is to highlight the more subjective cases.

**Interrogative/Question particles.** The treatment of interrogative or question particles has largely varied in the UD literature.\textsuperscript{6} We follow the preceding Indo-Aryan treebanks in assigning question particles with the respective dependency and POS tags as what would be assigned for a valid answer substitution. However, in cases where an obvious substitution is not viable (e.g., Yes/No questions) as shown in Example 1, we find that an aux relation fits the best.

\[ 1 \]

\[ \text{Do you want to go ?} \]

\[ 'Do you want to go ?' \]
Non-projectivity. Bhat et al. (2017, pp.23) discuss non-projectivity in Hindi. Gujarati allows non-projective trees in a similar spirit. Partial free word order as shown in Example 2 can give rise to overlapping dependency edges.

\[
\begin{array}{c}
\text{akasmāta māmle CBIE ċārjśīṭa kari} \\
\text{accident topic.LOC CBI.ERG chargesheet did} \\
\end{array}
\]

‘CBI made a chargesheet about the accident’

Head-final conjunctions. UD guidelines necessitate that the head of a conjunctive phrase be the first conjunct. However, Gujarati carries case inflections and post-positional attachments on the final conjunct which mediate semantic relations between the governor and the conjunctive phrase (see Example 3). This may lead to unwarranted non-projectivity as shown in Example 4.

Note that, in Example 4, the English translation fails to mark plurality on the verb "won" while in Gujarati "jītyā" has a plural inflection. As a result, the entire conjunctive phrase, not individual proper nouns (Peter or Mary), has to be the subject. At first sight, the non-projectivity in this example may seem avoidable by annotating promoted subject "pīṭara" as root, and attaching "rajata" to "pīṭara" as orphan, with the second clause attached as conj to the first clause. However, this would cause the plural verb to agree with a singular subject which is not the head of the coordinated structure. Similar issues also arise due to fixed head-initial coordination rule in UD for other head-final languages (Çöltekin, 2015; Kanayama et al., 2018; Tyers et al., 2017; Han et al., 2020). Hence, an argument can be made to mark the final conjunct as the head of the conjunctive phrase. However, we follow the UD guidelines and mark the first conjunct to be the head of the phrase.

Polarity/emphatic markers within serial verb constructions. Gujarati supports verb-verb constructions where the second verb is, usually, semantically bleached. Owing to the existence of partial free-word ordering discussed before, we observe that serial verb constructions are often separated by polarity or emphatic particles as seen in Example 5. To the best of our knowledge, this case is idiosyncratic to Gujarati. However, note that the treatment of these particles does not change.

Ideophonic verbs. In Gujarati, repetitions of a word can occur in two cases: discursive repetitions (બોલ બોલ ["tell tell"], જા જા ["go go"]) and onomatopoeias (ધમ ધમ ["dham dham"], the sound of Indian drums). Example 6 presents a case of onomatopoeias. Szubert et al. (2021) introduced parataxis:repeat for expressing adjectival repetitions in child-directed speech. Sulubacak et al. (2016) use compound:redup for replicated words. In our case, onomatopoeias are used to imitate different sounds that express actions and act as verbal repetitions. Hence, we suggest using compound:svc. To indicate the ideophonic nature of the verb, we mark the feature VerbType=Ideo.

7As noted in https://github.com/UniversalDependencies/docs/issues/842
Absence of clausal subjects. We find that clausal subjects do not exist in Gujarati. We substantiate this argument using an English example, “What she said is likable.”: i) A perfect translation of this sentence does not exist in Gujarati. A close translation is given in Example 7. Note that a coreferential pronominal ते [te, that] is added to construct a grammatically sound sentence. ii) Secondly, the presence of a dative nominal construction with experiencer semantics is permitted. Such constructions are considered grammatical subjects (Arora, 2022) which makes clausal subjects impossible. iii) Finally, the mandatory coreferential pronominal mediates the relation between the governor and the would-be subject clause.

Challenging Construction. Example 8 depicts a case where arguments can be made for multiple possible annotations: i) Assigning det:predet to ए [e] and det ज [je] with पुस्तक [pustaka] as their head ii) One may argue a change in order between “जे” and “पुस्तक”, where “जे” would act as a subordinating conjunction. However, we contend a semantic difference between this sentence and the one presented in Example 8. We lean towards the first annotation.

Quoter and Quotation. We encounter a screenplay dialog-style quotation that is yet to be resolved (see Example 9). Recent guidelines recommend ccomp over parataxis for reported

8This is not a Gujarati-specific issue. Moreover, we have opened a discussion regarding this point:
speech. We believe this to be a much more pervasive (and not a Gujarati-specific) issue; applicable, perhaps, when UD is extended to plays.

\[(9) \quad \text{parataxis / ccomp}
\]

I play football : Mark
l’play football : Mark’

4. Tokenization and Part of Speech

Splitting Genitive Markers. Certain nominals (and, in some instances, verbs) in Gujarati are inflected for case. It is unclear if these suffixes should be separated from their heads. This is a known issue that has been raised in Ravishankar (2017). They choose to split genitive markers to be consistent with Hindi. We follow the same rule with the added incentive to separate out layer III postpositions that pair postpositions with preceding genitive markers (Masica, 1993).

The Case for Determiners. According to Gujarati grammars (Tisdall, 1892; Doctor, 2004), demonstrative pronouns like એ [e], તે [te], પેલું [pelum], etc. behave differently when attached to a nominal, versus when used independently. When occurring independently, we treat them as pronouns. Tisdall (1892) argues to treat them as adjectives when used with nominals (e.g., એ કૂતરો ‘that dog’). Gujarati grammar does not discuss determiners as such. However, we see this usage closer to the UD definition of determiners and hence use the same.

Modal auxiliaries. There are several verbs that can be compounded with other verbs, nouns, or adjectives to form verb compounds. While most of these are semantically bleached, Gujarati identifies a fixed set of verbs to act as modal auxiliaries (Doctor, 2004). This fixed set includes verbs like ‘ગય [ga, go], છૂંડ [ava, come], રહે [rahe, stay]’ (temporal), ‘કર [kara, do], લાગ [lāga, feel]’ (compulsion), and ‘પડ [pada, fell], જોઈ [joī, want]’ (obligation). We mark these fixed set of verbs as auxiliaries while the rest are marked as regular verbs.

5. Conclusion and Future Work

We present the first dependency treebank in the Gujarati language and script. We provided detailed dataset statistics and discussed interesting examples and decisions. In a low-resourced language like Gujarati, we see this sample treebank as an enabler for future computational linguistics research. In the future, we aim to increase the size of the annotated corpora to help contribute a dependency parser. Furthermore, we also intend to provide annotations for the morphological features of Gujarati.

6. Ethics Statement

The dataset presented in this work is a voluntary annotation effort between the two authors of this paper. While the annotators speak different dialects of Gujarati, we are aware that our corpus might not contain diverse dialectical varieties.

Acknowledgements

We thank Dr. Atul Kr. Ojha and Aryaman Arora for their useful insights. We thank the anonymous reviewers for their valuable feedback. We highly appreciate the feedback received at the UniDive (CA21167) General Meeting 2023 where we presented an extended abstract version of this work. The format of the paper is inspired by Arora (2022).

7. Bibliographical References


Overcoming Early Saturation on Low-Resource Languages in Multilingual Dependency Parsing

Jiannan Mao†∗, Chenchen Ding‡, Hour Kaing‡, Hideki Tanaka‡, Masao Utiyama‡, Tadahiro Matsumoto†
†Gifu University, Gifu, Japan
‡National Institute of Information and Communications Technology, Kyoto, Japan
†{mao, tad}@mat.info.gifu-u.ac.jp
‡{chenchen.ding, hour_kaing, hideki.tanaka, mutiyama}@nict.go.jp

Abstract
UDify (Kondratyuk and Straka, 2019) is a multilingual and multi-task parser fine-tuned on mBERT that achieves remarkable performance in high-resource languages. However, the performance saturates early and decreases gradually in low-resource languages as training proceeds. This work applies a data augmentation method and conducts experiments on seven few-shot and four zero-shot languages. The unlabeled attachment scores were improved on the zero-shot languages dependency parsing tasks, with the average score rising from 67.1% to 68.7%. Meanwhile, dependency parsing tasks for high-resource languages and other tasks were hardly affected. Experimental results indicate the data augmentation method is effective for low-resource languages in a multilingual dependency parsing.

Keywords: Parsing, Multilinguality, Low Resource Languages, Unsupervised Learning

1. Introduction

A dependency parser can be efficiently trained when large treebanks are available (Dozat and Manning, 2017; Qi et al., 2020). For low-resource languages with no (zero-shot) or limited (few-shot) treebanks, multilingual modeling has emerged as an efficient solution, where cross-lingual information is leveraged to compensate for the lack of data. Scholivet et al. (2019); Üstün et al. (2022) have demonstrated that the performance on multilingual tasks can be boosted by pairing languages with similarities. Multilingualism also reduces the expense when training multiple models for a group of languages (Johnson et al., 2017; Aharoni et al., 2019; Cai et al., 2021; Muennighoff et al., 2023).

UDify (Kondratyuk and Straka, 2019) is a multi-task network fine-tuned on multilingual BERT (mBERT) (Devlin et al., 2019) pre-trained embeddings. It is capable of producing annotations for any treebank from Universal Dependencies (UD) (Zeman et al., 2018). UDify exhibits strong and consistent performance across all 124 UD treebanks for 75 languages and multiple tasks such as lemmatization, part-of-speech (POS), and dependency parsing. However, an issue not yet paid enough attention in several related studies is the substantial discrepancy found in the performance of these methods in low-resource language learning scenarios, even when almost the identical training strategies, datasets, models, and evaluation methods were used in Choudhary (2021), Üstün et al. (2022), Effland and Collins (2023).

To address and investigate this issue, the work of Mao et al. (2023) conducts an experimental exploration into the low-resource case phenomenon by observing changes during model training. They adopted the data augmentation strategy, which leverages the original UDify for parsing raw sentences in single low-resource language to obtain initial probabilities. This is followed by the application of unsupervised learning to train these probabilities. Using the trained probabilities to create artificially structured dependency data and merging them into UDify’s training set enables UDify to be trained on a more extensive dataset.

In this work, we conducted comprehensive experiments on low-resource languages using data augmentation methods, expanded (for few-shot languages) and created (for zero-shot languages) artificial treebanks for the seven few-shot and four zero-shot languages.
zero-shot languages. By combining these artificial treebanks with the UD treebanks and using the UDify framework, we trained a multilingual parser. As a result, increases in the unlabeled attachment score (UAS) for zero-shot languages were observed, with the average value increasing from 67.1% to 68.7%; in the most-improved case, the UAS rocketed from 78.4% to 88.0%. Similarly, the few-shot languages experienced a UAS increase of 0.2%. In contrast, the UAS for other languages and evaluation scores for other tasks did not show significant changes, which suggests that the overall robustness of multilingual and multi-task processing is retained.

2. Background

2.1. UDify

The UDify model jointly predicts lemmas, POS tags, morphological features, and dependency structures. The pre-trained mBERT model\(^1\) is used in the UDify model for cross-lingual learning without additional tags to distinguish the languages. In addition, a strategy similar to ELMo (Peters et al., 2018) is adopted, where a weighted sum of the outputs of all layers is computed as follows and fed to a task-specific classifier:

\[
e_{\text{task}} = \sum_i m_{\text{BERT}}_{ij}.
\]

Here, \(e_{\text{task}}\) denotes the contextual output embeddings for tasks such as the dependency parse. In addition, \(m_{\text{BERT}}_{ij}\) denotes the \(m_{\text{BERT}}\) representation for layer \(i\) at token position \(j\).

In the task involving dependency structures, mBERT’s subword tokenization process inputs words into multiple subword units. However, only the embeddings \(e_{\text{task}}\) of the first subword unit are used, serving as input to the graph-based bi-affine attention classifier (Dozat and Manning, 2017). The resulting outputs are combined using bi-affine attention to produce a probability distribution of the arc-head for each word. Finally, the dependency tree is decoded using the Chu–Liu/Edmonds algorithm (Chu, 1965; Edmonds et al., 1967).

2.2. Unsupervised Dependency Learning

Adhering to the properties of dependency syntax (Robinson, 1970), a general unsupervised algorithm for projective N-gram dependency learning (Unsupervised-Dep) was described in Ding and Yamamoto (2013, 2014). This method constructs the best dependency tree with a dynamic programming method using a CYK style chart and is based on the complete-link and complete-sequence non-constituent concepts. However, considering the time complexity of this approach for arbitrary N-gram dependency learning, which may not be ideal for practical applications, we chose to focus in this study on the case of the bi-gram.

When considering the bi-gram, the directionality of a pair of words is set by the dependency relation, with \((w_{i} \rightarrow w_{j})\) indicating a rightward relation and \((w_{i} \leftarrow w_{j})\) indicating a leftward one. The bi-gram unsupervised learning update probabilities \(P(w_{i} \rightarrow w_{j}) \) and \(P(w_{i} \leftarrow w_{j})\) are calculated using the Inside–Outside algorithm (Lari and Young, 1990). Finally, the Viterbi algorithm (Forney, 1973) is employed to determine the tree construction in the calculated Inside portion with the maximum probability, thus generating the optimal structure.

3. Investigation

3.1. UDify with Data Augmentation

In the work of Mao et al. (2023), a data augmentation based on Unsupervised-Dep is provided. Due to Unsupervised-Dep has a high time complexity of \(O(n^3)\), making the common practice in the original methods, which start training from a random probability, somewhat inefficient. To circumvent this, the parsing results from UDify were utilized to initialize the probabilities. Despite the potential decrease in UDify’s accuracy on low-resource languages during its training, the final results consistently outperform those from other parsing models (Qi et al., 2018; Tran and Bisazza, 2019), providing a solid foundation for the proposed initialization approach.

The process starts with the raw corpus, Data, input into the trained UDify by the original UD treebank, to generate the dependency arc-heads, represented as DEP\(_{arc}\), and POS, lemmas, etc., denoted as Others. Statistical computations on DEP\(_{arc}\) generate initial probabilities \(P(w_i \rightarrow w_j)\) and \(P(w_i \leftarrow w_j)\), serving as input for Unsupervised-Dep alongside Data.

Following several iterations of training through Unsupervised-Dep, the re-estimated \(P(w_i \rightarrow w_j)\) and \(P(w_i \leftarrow w_j)\) emerge. They become the parameters for the Viterbi algorithm to determine the optimal dependency arc-head as given by

\[
\text{DEP}_{arc} = \text{Viterbi}(x, P(w_i \rightarrow w_j), P(w_i \leftarrow w_j)),
\]

where \(\text{DEP}_{arc}^*\) is the tree with the highest probability for a sentence \(x\) from Data.

Finally, \(\text{DEP}_{arc}^*\) is merged with Others, ultimately generating artificial data. The artificial data are then combined with the existing UD treebanks for the subsequent UDify training.

3.2. On Few- and Zero-Shot Languages

During the training of UDify, the dependency structures for zero-shot languages are learned through

\(^1\)github.com/google-research/bert/multilingual.md
transfer learning. Compared to high-resource languages, an early saturation in the accuracy of dependency parsing is observed across all zero-shot languages during the learning process. The peak performance is typically reached around the 12th training epoch, as illustrated in Figure 2. Mao et al. (2023) applied data augmentation to individual zero-shot language, effectively addressing this issue.

![Figure 2: Change in the UAS(\%) of low-resource languages during UDify(our) training.](image)

However, when applying Unsupervised-Dep data augmentation to multiple zero-shot languages, the effectiveness of this approach has not been explored due to the impact of the amount of data generated on parser performance. Especially considering that this approach may generate large amounts of artificial data, its practical application in this context needs to be evaluated.

Moreover, the training of multilingual parser reveals that few-shot languages are similarly affected by the volume of training data. This highlights the critical need for effective data augmentation methods to improve the parsing performance of models like UDify. We aim to employ Unsupervised-Dep for multiple languages to explore its potential in mitigating early saturation in zero-shot languages and improving parsing accuracy in few-shot languages within a multilingual context.

4. Experiments

4.1. Dataset

The raw data of seven few-shot and four zero-shot languages that are most often tokenized using spaces were collected from El-Kishky et al. (2020); Fan et al. (2021); Schwenk et al. (2021) to create our selected low-resource language set for the implementation of Unsupervised-Dep. The data in the experiment are summarized in Table 1 and referred to as OPUS-mult in subsequent sections.

For comparison with the UDify and to illustrate our motivation, our parser experiments employed the UD Treebank v2.3 used by UDify. During training, following McDonald et al. (2011), we merged training sets, randomized the sentence order each epoch, and fed the network diverse batches of original and artificial data from multiple languages.

<table>
<thead>
<tr>
<th>language(code)</th>
<th>#sent.(len.)</th>
<th>#train</th>
<th>#test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Armenian(hy)</td>
<td>2.4(8.2)</td>
<td>560</td>
<td>470</td>
</tr>
<tr>
<td>Belarusian(be)</td>
<td>2.0(9.0)</td>
<td>260</td>
<td>68</td>
</tr>
<tr>
<td>Hungarian(hu)</td>
<td>134.1(5.3)</td>
<td>910</td>
<td>449</td>
</tr>
<tr>
<td>Kazakh(kk)</td>
<td>1.7(8.2)</td>
<td>31</td>
<td>1047</td>
</tr>
<tr>
<td>Lithuanian(lt)</td>
<td>236.7(5.6)</td>
<td>153</td>
<td>55</td>
</tr>
<tr>
<td>Marathi(mr)</td>
<td>1.5(10.0)</td>
<td>373</td>
<td>47</td>
</tr>
<tr>
<td>Tamil(ta)</td>
<td>13.7(7.7)</td>
<td>400</td>
<td>120</td>
</tr>
<tr>
<td>Breton(br)</td>
<td>18.2(9.5)</td>
<td>0</td>
<td>888</td>
</tr>
<tr>
<td>Faroese(fo)</td>
<td>1.3(8.1)</td>
<td>0</td>
<td>1,208</td>
</tr>
<tr>
<td>Tagalog(tl)</td>
<td>150.0(16.2)</td>
<td>0</td>
<td>55</td>
</tr>
<tr>
<td>Yoruba(yo)</td>
<td>9.7(8.1)</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 1: Raw data collected from various corpora. Above: few-shot languages; below: zero-shot languages. #sent.(len.) denotes the raw sentences in unsupervised learning (in thousands), with the numbers in parentheses indicating the average length. #train and #test are the sentence counts in UD v2.3 treebank’s training and test sets, respectively.

4.2. Setup

To minimize the impact of experimental environment variations on the result of Popel and Bojar (2018) in the comparisons, we followed the parameter settings from Kondratyuk and Straka (2019) and re-implemented the model as UDify(our). Additionally, to expedite the training process, we employed Horovod (Sergeev and Balso, 2018) to implement parallel computation.

At the beginning of training on Unsupervised-Dep, we used the UDify(model) to parse each language present in the OPUS-mult dataset. The statistical results derived from the parsing outcomes of each language were adopted as its initial probabilities, which were continuously re-estimated throughout the unsupervised learning process. After the 10th training iteration, we employed the newly estimated probabilities to parse the sentences from OPUS-mult.

To assess the impact of augmenting training data for multiple low-resource languages on parsing accuracy, we designed and conducted several experiments. In the Unsupervised-Dep data augmentation experiments, we randomly selected 300 sentences for each language from OPUS-mult, processed them using Unsupervised-Dep, and integrated them into the UD treebanks to form the training dataset. The model trained from this dataset is referred to as Unsup. Inspired by the work of Rybak and Wróblewska (2018), we conducted a comparative experiment using a data augmentation method dubbed Self. In this approach, we used the same raw sentences train Unsup model and directly applied the parsing results obtained from the UDify(org) model. These results were merged with the original training set to train the Self model.
Table 2: UAS(%) for few- and zero-shot languages obtained using different methods. The last two columns display the combined test set results for few- (Few) and for zero-shot (Zero) languages. We denote the treebank names using language codes; both the low-resource languages have only one treebank in UD v2.3. The UDify(org) result was reported in Kondratyuk and Straka (2019).

<table>
<thead>
<tr>
<th>Method</th>
<th>hy</th>
<th>be</th>
<th>hu</th>
<th>kk</th>
<th>lt</th>
<th>mr</th>
<th>ta</th>
<th>br</th>
<th>fo</th>
<th>tl</th>
<th>yo</th>
<th>Few</th>
<th>Zero</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDify(org)</td>
<td>85.6</td>
<td>91.8</td>
<td>89.7</td>
<td>74.8</td>
<td>79.1</td>
<td>79.4</td>
<td>79.3</td>
<td>63.5</td>
<td>67.2</td>
<td>64.0</td>
<td>37.6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>UDify(our)</td>
<td>86.1</td>
<td>92.1</td>
<td>89.8</td>
<td>76.0</td>
<td>79.4</td>
<td>74.3</td>
<td>80.8</td>
<td>69.2</td>
<td>72.0</td>
<td>78.4</td>
<td>39.4</td>
<td>84.0</td>
<td>67.1</td>
</tr>
<tr>
<td>Self</td>
<td>85.9</td>
<td>92.5</td>
<td>89.6</td>
<td>76.2</td>
<td>79.2</td>
<td>74.8</td>
<td>81.2</td>
<td>69.8</td>
<td>72.5</td>
<td>85.3</td>
<td>38.8</td>
<td>84.0</td>
<td>67.6</td>
</tr>
<tr>
<td>Unsup</td>
<td>86.3</td>
<td>92.4</td>
<td>90.0</td>
<td>76.2</td>
<td>79.5</td>
<td>74.0</td>
<td>80.5</td>
<td>72.7</td>
<td>71.9</td>
<td>88.0</td>
<td>39.6</td>
<td>84.2</td>
<td>68.7</td>
</tr>
</tbody>
</table>

Table 3: UD scores on selected zero-shot and other languages obtained by different methods. Rest(%) refers to the average score of UPOS, UFeats, Lemma, and LAS in the UD scores.

### 4.3. Result and Discussion

A comparison with the experimental findings from Kondratyuk and Straka (2019) confirms the successful re-implementation of UDify(our), as illustrated in Table 2, and reveals that our replicated model surpasses those in related work (Choudhary, 2021; Üstün et al., 2022; Mao et al., 2023). Although no method produced a noticeable improvement for the few-shot languages, the results in this table indicate a significant improvement in UDify’s ability to parse the dependency arc-head accuracy for zero-shot languages at the end of the training with the Unsupervised-Dep data augmentation method. This is reflected in the results for the combined test set, where the UAS increased to 68.7%. Taking Breton from the zero-shot languages as an example, we illustrate the changes in UAS during the training process under different methods in Figure 1. The figure reveals that the inclusion of data generated through Unsupervised-Dep significantly mitigates the reduction in UAS accuracy for zero-shot languages over the course of the training, thereby improving the result.

The UAS of almost every zero-shot language improved when artificial data via Unsupervised-Dep were included. To our knowledge, this is the state-of-the-art result for Tagalog. The Tagalog-TRG treebank is quite small, encompassing only 55 sentences with an average sentence length of 4.2 words in UD v2.3. In contrast, we have gathered 150k Tagalog raw sentences with an average length of 16.2 words. We believe that the quality and quantity of raw sentences used for training Unsupervised-Dep have a crucial impact on the performance of the multilingual parser.

To further enhance UDify’s dependency parsing accuracy in low-resource languages, we attempted to increase the number of sentences generated by Unsupervised-Dep data augmentation to 50k, which we refer to as Unsup+. In the result of Unsup+, the UAS of the selected zero-shot languages in the test set saw further improvement, reaching 69.3%. We depict the changes in UAS for Breton during the Unsup+ training process in Figure 1.

Given UDify’s standing as a multilingual and multi-task parser, assessing the impact of our proposed methods on other languages and tasks is essential. To further scrutinize the variations between the UAS results of UDify(org) and Unsup, we carried out tests on all treebanks. As shown in Figure 3, the results indicate that Unsup effectively enhanced the UAS of zero-shot languages when artificial data were created using Unsupervised-Dep, especially for Breton and Tagalog. Meanwhile, its impact on the parsing precision of dependency structures in other languages is negligible.

For a comprehensive comparison, the UD scores of the zero-shot and other languages have been compiled in Table 3. Given that UDify must balance the loss produced by multiple decoders during training and the work of Rybak and Wróblewska (2018), these variations in evaluation metrics are considered reasonable. Broadly, our method has not had a negative impact on other languages and tasks, maintaining their performance levels.

Considering all results, we argue that creating training data for multiple low-resource languages using Unsupervised-Dep is both essential and effective in multilingual modeling contexts.
5. Conclusion and Future Work
This study highlights the issue of early saturation in parsing accuracy for UDify across multiple low-resource languages. To address this challenge, we implemented data augmentation for several low-resource languages through unsupervised learning. The experimental results demonstrated the effectiveness of data augmentation method in enhancing the parsing performance of multilingual parsers for low-resource languages.

Despite the limitations posed by training speed and the quality and quantity of raw data on our experiments, two possibilities remain: (1) Generating more data for zero-shot languages could lead to positive improvements. (2) The quality and quantity of raw data play a crucial role in the effectiveness of unsupervised data augmentation methods, thereby affecting the performance of multilingual parsers.

In future work, our research aims to explore additional influencing factors and considerations to further enhance multilingual parsing performance in low-resource language scenarios. Moreover, we plan to conduct research and exploration on low-resource languages using the latest UD treebanks.

6. Bibliographical References


Alexander Sergeev and Mike Del Balso. 2018. Horovod: Fast and easy distributed deep
learning in TensorFlow.  


Part-of-Speech Tagging for Northern Kurdish

Peshmerge Morad¹ Sina Ahmadi² Lorenzo Gatti³
¹University of Twente ²University of Zurich ³University of Twente
¹p.morad@hotmail.com ²sina.ahmadi@uzh.ch ³l.gatti@utwente.nl

Abstract

In the growing domain of natural language processing, low-resourced languages like Northern Kurdish remain largely unexplored due to the lack of resources needed to be part of this growth. In particular, the tasks of part-of-speech tagging and tokenization for Northern Kurdish are still insufficiently addressed. In this study, we aim to bridge this gap by evaluating a range of statistical, neural, and fine-tuned-based models specifically tailored for Northern Kurdish. Leveraging limited but valuable datasets, including the Universal Dependency Kurmanji treebank and a novel manually annotated and tokenized gold-standard dataset consisting of 136 sentences (2,937 tokens). We evaluate several POS tagging models and report that the fine-tuned transformer-based model outperforms others, achieving an accuracy of 0.87 and a macro-averaged F1 score of 0.77. Data and models are publicly available under an open license at https://github.com/peshmerge/northern-kurdish-pos-tagging

Keywords: Part-of-Speech tagging, morphosyntactic analysis, Northern Kurdish, low-resource NLP

1. Introduction

Automatic part-of-speech (POS) tagging or grammatical tagging is the process of assigning POS tags to each word/token in a given text. POS tagging is essentially a disambiguation task because words naturally are ambiguous and can have more than one correct tag depending on the context and their position in the sentence. POS tagging serves many purposes in natural language processing (NLP) applications, and it is traditionally considered a building block for other tasks such as named entity recognition (Ma and Liu, 2021), information extraction (Luan et al., 2017), spelling correction (Nagata et al., 2018), text classification (Pranckevičius and Marcinkevičius, 2016), natural language generation (Li et al., 2019), and machine translation (Hlaing et al., 2022).

Just as part-of-speech tagging serves as a precursor for tasks like syntactic parsing, tokenization is a crucial task in NLP and a prerequisite for POS tagging. Tokenization is segmenting the input text into smaller, distinct units termed tokens. These tokens can encompass compound words, single words, sub-words, symbols, or other significant elements. At its most fundamental level, tokenization separates tokens using whitespace as a delimiter (Mittkov, 2022, p. 549).

Unlike high-resourced languages (HRLs) like English and French, for which POS tagging and tokenization have been extensively addressed, low-resourced languages (LRLs) like Kurdish lack sufficient tools and resources (Ahmadi, 2020a). Although Northern Kurdish is included in Universal Dependencies (UD) (Nivre et al., 2020) (using the ‘Kurmanji’ label since version 2.1) based on Gökirmak and Tyers (2017)’s treebank, hence serving as a benchmark, achieving high-accuracy POS tagging for LRLs may require a greater emphasis on linguistic insights as observed in other languages (Manning, 2011). Our literature review indicates that there is room for effective and open-source contributions to Kurdish POS tagging.

In this paper, we report on the progress we have made in addressing the task of POS tagging for Northern Kurdish. More specifically, we revisit the UD Kurmanji treebank (Gökirmak and Tyers, 2017) by reannotating tokens that belong to specific word classes and introducing a different annotation scheme with more fine-grained linguistic features of Northern Kurdish. Secondly, we create a manually tokenized and annotated gold-standard dataset for Northern Kurdish with a total of 136 sentences and 2,937 tokens. To that end, we deploy an annotation scheme different from that of UD Kurmanji that aims for a more fine-grained representation of linguistic features of Northern Kurdish, notably noun phrases containing Izafe (also spelled Ezafe) acting as a relativizer and linker. Thirdly, we evaluate the effect of different POS techniques along with the annotation schemes. Finally, we implement different POS tagging models and introduce a state-of-the-art transformer-based POS tagger for Northern Kurdish.

The rest of the paper is organized as follows. In section 2, we provide an overview of the Kurdish language and its dialects, focusing on Northern Kurdish. Section 3 presents a comprehensive review of related work and state-of-the-art studies on POS tagging for LRLs in general, with a specific focus on Northern Kurdish. We then detail the annotation schemes for the training and testing datasets in section 4. In section 5, we discuss the process of collecting and annotating testing data, as well as augmenting the training data. Additionally, we provide a detailed explanation of the tokenization and POS tagging methods.
quently, section 6 presents our evaluation results, accompanied by an in-depth analysis. Finally, our conclusions are presented in section 7.

2. Kurdish Language

The Kurdish language belongs to the Northwestern Iranian branch within the Indo-European languages family, spoken by more than 30 million people. The Kurdish language (ISO 639-3 code kur) is divided into many dialects (with corresponding ISO 639-3 languages codes): Northern Kurdish or Kurmanji (kmr), Central Kurdish or Sorani (ckb), Southern Kurdish (sdh), and Laki (ldk) and is closely related to Zaza-Gorani languages (Ahmadi et al., 2019). Northern Kurdish is widely spoken in Syria and Turkey but also in the Kurdistan Region of Iraq, Iran, Armenia and among the Kurdish diaspora. It is written using Kurdified Latin-based and Arabic-based scripts. The Latin-based script is widely known as the Hawar alphabet introduced by Jeladet Ali Bedirkhan in 1932.

Northern Kurdish has a subject–object–verb word order and specifies grammatical gender (feminine and masculine). The noun in its absolute state and without any suffixes represents the generic and definite senses of the noun, and it marks four cases, namely nominative, oblique, izafe, and vocative. In addition, it has a split-ergative alignment in the past tense with transitive verbs. Furthermore, the passive voice (conjugated in all persons, moods, and tenses) is constructed using the verb hatin ‘to come’ and dan ‘to give’ plus the infinitive.

Both the oblique and the izafe case (construct case) are essential in Northern Kurdish for indicating the roles of the nouns and the pronouns in a sentence. Nouns, proper nouns, personal pronouns, and demonstrative adjectives, in both cases, undergo a form change as in “komputerar min” (my computer) where ‘a’ is an izafe linking ‘komputer’ (computer) to ‘min’ (my). They are either completely altered, or the case markers are added to the end of the noun and proper nouns. Those markers, shown in Table 1, are unstressed markers that reveal the gender and number of nouns. In this study, our introduced annotation scheme, discussed in Section 5.1, particularly revolves around addressing and segmenting the oblique and izafe case markers in our datasets.

Nonetheless, izafe case markers differ from oblique case markers in the fact that they can also appear as separate particles serving the same purpose within definite nouns; this phenomenon is referred to as construct extender (Thackston, 2006) because it allows extending the izafe case by adding adjectives or nouns to the first izafe case.

<table>
<thead>
<tr>
<th></th>
<th>Oblique</th>
<th>Izafe</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG. F.</td>
<td>-ê</td>
<td>-êkê</td>
</tr>
<tr>
<td>SG. M.</td>
<td>-ë</td>
<td>-êkê</td>
</tr>
<tr>
<td>Pl.</td>
<td>-ê</td>
<td>-êkê</td>
</tr>
</tbody>
</table>

Table 1: Case markers based on the number, gender, and definiteness of the noun in Northern Kurdish. If the noun ends in a vowel, the case markers will be preceded by a -y.

3. Related Work

The task of POS tagging has been addressed using various methods. Rule-based techniques (Brill, 1992; Karlsson, 1990) were the first methods applied. Decision Trees have also been employed for the task (Schmid, 1994). Furthermore, hidden Markov models (HMMs) and conditional random fields (CRFs) have been widely used and proved to be effective for this task (Schmid and Laws, 2008; Pradhan and Yajnik, 2023; Yousif, 2019; Stratos et al., 2016; Silfverberg et al., 2014).

Additionally, deep learning based approaches like recurrent neural networks and (Bi)LSTMs have shown to be powerful in capturing temporal dependencies when performing POS tagging (Wang et al., 2015; Qi et al., 2020; Horsmann and Zesch, 2017). Those are often combined with other techniques such as convolutional neural networks, HMMs, and CRFs (Shao et al., 2017; Plank et al., 2016; Ma and Hovy, 2016; Maimaiti et al., 2017).

In recent years, the rise of transformer-based architectures introduced by Vaswani et al. (2017) has led to the development of large language models (LLMs) such as GPT2 (Radford et al., 2019), BERT (Kenton and Toutanova, 2019) and RoBERTa (Liu et al., 2019). These models have greatly influenced NLP in various fields. However, despite being trained on multiple languages, they don’t always perform better than single-language models, especially in less-resourced languages, for tasks like POS tagging (Conneau et al., 2020). Nonetheless, they can adapt and improve their performance when fine-tuned (Maimaiti et al., 2021).

For Kurdish, Walther et al. (2010) presents the first dedicated work on POS tagging for Northern Kurdish, where a morphological lexicon (KurLex) and a POS tagger were created. The authors report an 85.7% precision, however on a small annotated corpus of 13 sentences. Although Gökirmak and Tyers (2017)’s treebank for Northern Kurdish is available on UD and has been used in various consecutive studies in multilingual training setups as in Qi et al., 2020 (BILSTM) and Nguyen et al., 2021 (transformer-based fine-tuning) inter alia, there is still no tool or fine-grained dataset in-
indicating the existing gap in the literature (Ahmadi, 2020a).

4. Annotation Schemes

4.1. UD Kurmanji Scheme

The UD Kurmanji treebank (Gökrmak and Tyers, 2017) is a treebank for Northern Kurdish that contains morpho-syntactic information such as POS tags and some morphological features. The data in the treebank is drawn from fiction and encyclopedic data in roughly equal measure. It consists of the Kurdish translation of The Adventure of the Speckled Band story and sentences from the Northern Kurdish Wikipedia. UD Kurmanji contains 10,189 tokens and has been annotated following the UD annotation scheme (Nivre et al., 2020), meaning it does not allow multi-word expressions, and it instructs to undetected contractions. In addition, the case markers, shown in Table 1, within nouns are not segmented. Moreover, the construct extenders in the treebank are tagged as ADP. For example, the noun phrase ‘Beşa Felsefeyê’ (department of philosophy) is tagged as NOUN and NOUN, respectively, while having Izafe and oblique case markers in both nouns.

4.2. Our Scheme

We propose a different, fine-grained annotation scheme taking into account all case and indefinite noun markers. In addition, we address multi-word prepositions such as ‘lê’ (from, analogous to au/aux in French), adverbs, and compound verbs and tag them as single tokens. It is worth mentioning that the UD annotation scheme (Nivre et al., 2020) serves as a basis for our scheme.

Case Markers and Determiners One of the main differences between our scheme and the UD Kurmanji scheme is how we segment the nouns and their attached indefinite, oblique, and Izafe case markers. We use the POS tags from the UD tagset (Petrov et al., 2012). While we use DET for indefinite and oblique case markers, we introduce a new POS tag named IZAFE for the Izafe case markers. For example, the noun phrase Beşa Felsefeyê (department of philosophy) is split into four tokens Beş, a, Felsefe yê and respectively tagged as NOUN, IZAFE, NOUN and DET.

Multi-word Expressions In UD Kurmanji, the tag X is assigned to nouns that are part of the compound verbs; in our case, we tag those nouns either as a NOUN or all together with the verbs they belong to as a multiword expression VERB. For instance, in UD Kurmanji, the compound verb ‘pêşkêş dikin’ (presenting) is split into two tokens: pêşkêş and dikin and tagged X and VERB, respectively. Within our annotation scheme, we tag it as VERB.

Regarding compound prepositions, we annotate the compound preposition ‘il ser’ (on/upon) as ADP, while in UD Kurmanji, it is separated into two tokens ‘il’ (in/at) and ‘ser’ (onto) where both are tagged as ADP. In addition, compound adverbs such as ‘bi tenê’ (only) are also separated into two tokens ‘bi’ (with) and ‘tenê’ (alone), both are annotated as ADP. However, we treat it as a multi-word token, and we annotate it as ADV.

Moreover, the verb to be in Northern Kurdish ‘bûn’ (to be) is always annotated as AUX in UD Kurmanji treebank, while we tag it as a VERB unless it appears as a light verb. In addition, the particles dê and dê are used for forming the future tense in Northern Kurdish and are tagged as AUX in UD Kurmanji. However, we tag those particles as PART because they are not auxiliary verbs.

Furthermore, the tokens ‘il’ (also/too) and ‘her’ (every) are annotated as PART and either as DET or ADV in the UD Kurmanji, respectively. We annotate the former as ADV and the latter as PRON.

5. Methodology

5.1. Data Collection and Annotation

We collect 136 (2,937 tokens) sentences written in Northern Kurdish from multiple news websites. The first 100 sentences are taken from the unannotated Pewan corpus (Esmaili et al., 2013). The remaining 36 sentences are taken from three Kurdish news websites, mainly Kurdistan241, Xwebun2, and Hawar News3. We annotated those sentences according to our annotation scheme introduced in section 4.2. We call this collection the “gold-standard dataset”, and we use it as a test set to evaluate our POS tagging models. Figure 1 demonstrates the statistics of this dataset.

Similar to the UD Kurmanji treebank, for each given sentence in our gold-standard dataset, we provide: 1) the raw (untokenized) sentence where tokens are delimited by whitespaces and the case markers are not split-off, and 2) a list of tokens with corresponding POS tags where the case markers are segmented and annotated.

The availability of the untokenized sentence, along with the list of the tokens, enables us to evaluate various tokenization methods. The untokenized sentence can be fed to any tokenizer, and its output can be compared against the list of tokens we already have, which we consider as gold tokens.

1https://www.kurdistan24.net/kmr
2https://xwebun1.org
3https://hawarnews.com/kr

72
5.2. Data Augmentation

We augment the UD Kurmanji treebank by splitting the case and indefinite markers from the tokens they are attached to. Thus introducing new tokens. For example, we split ‘hevali’ (a male friend) into three separate tokens, each with its corresponding POS tag: heval as NOUN, ek (indefinite noun marker) as DET, and finally i as IZAFE. In addition, we re-tag independent Izafe markers (construct extender) as IZAFE instead of ADP. Finally, we reverse the splitting of the contracted prepositions (jê, lê, pê, tê) in the treebank.

Our approach for augmenting the UD Kurmanji treebank bears a close resemblance to the research described by (Seddah et al., 2023). The authors made significant steps in addressing tokenization issues to ensure consistency in the NArabizi treebank annotations (Farah et al., 2020), the user-generated content variety of Arabic Algerian, which is known for its frequent usage of code-switching. For instance, they carefully segmented specific classes of words, such as determiners in noun phrases.

As a result of this augmentation step, the number of tokens increased in the treebank (12,233 tokens). We refer to this augmented version as UD Kurmanji augmented, while we refer to the version with its initial annotation scheme as UD Kurmanji original.

5.3. Tokenization

In addition to the KLPT tokenizer, Ahmadi, 2020b provided multiple neural tokenization models trained (unsupervised) on Northern Kurdish raw corpora. We use three of those models: Unigram (Kudo, 2018), Byte-Pair Encoding (BPE) (Sennrich et al., 2016), and wordPiece (Schuster and Naksima, 2012) tokenizers.

Moreover, we use the NLTK tokenizer and a manual tokenization method. The manual tokenization, as the name suggests, is the process of manually tokenizing any given text. This method is mostly performed in pairs with the task of manually annotating tokens with the corresponding POS tags. Despite being very time-consuming, it is considered to have the best outcome because it is done by humans with good linguistic knowledge of the language. Therefore, the manually tokenized text can be considered the ground truth that can be used for evaluating other automatic tokenization methods.

5.4. POS Tagging

The task of POS tagging can be seen as a multi-class classification task where a model is trained on annotated data to enable it to classify each token in any given sequence of tokens. There are multiple approaches to tackle the task of POS tagging. Generally, those approaches can be grouped into four categories: rule-based, statistical, neural-based, and transformer-based fine-tuned (Jurafsky and Martin, 2009; Kanakaraddi and Nandyal, 2018).

Except for the work of Walther et al., 2010, there has been no dedicated work for the task of POS tagging for Northern Kurdish. Therefore, we propose seven supervised POS tagging models. The goal is to cover POS methods as much as possible to establish a baseline method and to examine the effectiveness of those methods. Those methods will be explained in the following subsections.

It is worth mentioning that we train all POS tagging models independently, once on the UD Kurmanji original and once on the UD Kurmanji augmented. We take this approach because we want to assess the impact of the annotation scheme on the models’ performance. Hence, the labels (augmented) and (original) within the models' names indicate the dataset used for training the model, either UD Kurmanji augmented or UD Kurmanji original.

5.4.1. Statistical-based Models

Our first model is a Unigram model from the NLTK Python package (Bird et al., 2009). This model assigns tags based on word frequency observed during training. It uses conditional frequency distributions to calculate the most likely tag for each given token. The model may encounter unfamiliar words in linguistically resource-limited settings like ours (out-of-vocabulary). Therefore, we specify the default POS tag as NOUN when it fails to determine a POS tag for a token. This is a common practice when establishing a baseline, and it is motivated by Bird et al. (2009).
In addition, we create HMM (Huang et al., 2001) and CRF (based on CRFsuite library (Okazaki, 2007)) models using the implementation available in the NLTK Python package. Finally, we create an ExtraTrees POS model using the implementation from Scikit-learn (Pedregosa et al., 2011).

5.4.2. Neural-based Models

Our first neural-based model is the Averaged Perceptron POS tagging model, similar to the Extra Trees model, which has the notion of feature engineering. However, here we do not define our own set of features, we use the standard features set defined by the NLTK Python package since we use their implementation.

In addition, we use the Flair Python package (Akbik et al., 2019) to create a BiLSTM model using a configurable BiLSTM architecture as originally proposed by Huang et al. (2015). For this model, we use pre-trained sub-word fastText embeddings (Grave et al., 2018) specifically pre-trained on Northern Kurdish data. FastText enables us to generate embeddings from character-level n-grams, thereby being better at capturing morphological nuances.

5.4.3. Transformer-based Fine-tuned Models

In contrast to the previous models, where each model was trained from scratch for our task, we fine-tune the pretrained multilingual XLM-RoBERTa model (Conneau et al., 2020) on the UD Kurmanji original and UD Kurmanji augmented. We utilize the ‘base’ version of XLM-RoBERTa because of its lower computational requirements, making it easier to fine-tune. The fine-tuning is performed using TransKit (Nguyen et al., 2021), which offers a relatively fast and straightforward approach for fine-tuning LLMs like XLM-RoBERTa, thanks to the utilization of Adapters (Pfeiffer et al., 2020). We refer to the fine-tuned POS model as Northern Kurdish XLM-RoBERTa (NK-XLMR).

The intrinsic evaluation directly measures the tokenization system’s capabilities by comparing it to similar systems. We follow the same approach of (Ahmadi, 2020b) by performing tokenization evaluation using the Bilingual Evaluation Understudy Score (BLEU).

Table 2 shows the BLEU scores of the tokenization methods we used in this study using the gold-standard dataset as testing data. We see that the BLEU scores for the KLPT tokenizer are the highest, outperforming other tokenizers by a great margin. In contrast to other tokenizers, the KLPT tokenizer is characterized by its extensive knowledge of Northern Kurdish, enabling it to correctly recognize case markers and handle multi-word expressions like compound verbs and compound prepositions.

<table>
<thead>
<tr>
<th>Tokenizer</th>
<th>BLEU-1</th>
<th>BLEU-2</th>
<th>BLEU-3</th>
<th>BLEU-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>KLPT</td>
<td>0.73</td>
<td>0.65</td>
<td>0.59</td>
<td>0.53</td>
</tr>
<tr>
<td>unigram</td>
<td>0.54</td>
<td>0.44</td>
<td>0.36</td>
<td>0.29</td>
</tr>
<tr>
<td>NLTK</td>
<td>0.50</td>
<td>0.41</td>
<td>0.33</td>
<td>0.25</td>
</tr>
<tr>
<td>BPE</td>
<td>0.50</td>
<td>0.39</td>
<td>0.31</td>
<td>0.24</td>
</tr>
<tr>
<td>wordPiece</td>
<td>0.45</td>
<td>0.36</td>
<td>0.28</td>
<td>0.21</td>
</tr>
</tbody>
</table>

Table 2: BLEU scores for all tokenization methods on the gold-standard dataset.

Within the extrinsic evaluation, we evaluate the tokenization system by measuring its impact on our whole NLP pipeline. In our case, the tokenization system’s quality greatly affects the POS tagger’s performance. Therefore, the tokenization correctness can also be determined by examining the F1 and accuracy scores of the POS tagger presented in section 6.2.

6.2. POS Tagging Performance

We present the evaluation results (accuracy and macro-averaged F1 score) of all POS tagging models. In order to make the comparison clearer, we divide the results based on the used training data (UD Kurmanji original and augmented). While table 4 provides a detailed comparison of all models trained on the UD Kurmanji augmented, table 3 demonstrates the results of the same POS model but trained on UD Kurmanji original.

By comparing the results in both tables and regardless of the tokenization method, we observe a performance increase among the models. This increase is the highest within the manual tokenization method and the lowest within the wordPiece tokenization method. This confirms the importance and the impact of the data augmentation we did on the UD Kurmanji original treebank for the task of POS tagging. In addition, it stipulates the impact the performance of the tokenization method has on
Table 3: The macro-averaged F1 scores and accuracy (Acc) of the POS tagging models trained on the UD Kurmanji original and evaluated on our gold-standard dataset.

<table>
<thead>
<tr>
<th>Model / Tokenizer</th>
<th>manual F1</th>
<th>KLPT F1</th>
<th>unigram F1</th>
<th>NLTK F1</th>
<th>BPE F1</th>
<th>wordPiece F1</th>
<th>Acc</th>
<th>F1</th>
<th>Acc</th>
<th>F1</th>
<th>Acc</th>
<th>F1</th>
<th>Acc</th>
<th>F1</th>
<th>Acc</th>
<th>F1</th>
<th>Acc</th>
<th>F1</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (Unigram)</td>
<td>0.4 0.51 0.35 0.41 0.37 0.32 0.36 0.34 0.36 0.32 0.34 0.31</td>
<td>0.37 0.46 0.33 0.36 0.35 0.32 0.34 0.33 0.34 0.32 0.34 0.31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HMM</td>
<td>0.41 0.52 0.37 0.42 0.38 0.33 0.37 0.34 0.38 0.33 0.34 0.32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExtraTrees</td>
<td>0.44 0.54 0.37 0.42 0.40 0.36 0.38 0.37 0.39 0.35 0.36 0.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AveragedPerceptron</td>
<td>0.42 0.51 0.40 0.41 0.45 0.35 0.43 0.36 0.44 0.34 0.42 0.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BiLSTM</td>
<td>0.46 0.54 0.41 0.44 0.42 0.36 0.40 0.37 0.40 0.35 0.35 0.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRF</td>
<td>0.57 0.62 0.46 0.47 0.47 0.38 0.45 0.39 0.45 0.37 0.40 0.35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NK-XLMR</td>
<td>0.57 0.62 0.46 0.47 0.47 0.38 0.45 0.39 0.45 0.37 0.40 0.35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: The macro-averaged F1 scores and accuracy (Acc) of the POS tagging models, trained on the UD Kurmanji augmented and evaluated on our gold-standard dataset.

<table>
<thead>
<tr>
<th>Model / Tokenizer</th>
<th>manual F1</th>
<th>KLPT F1</th>
<th>unigram F1</th>
<th>NLTK F1</th>
<th>BPE F1</th>
<th>wordPiece F1</th>
<th>Acc</th>
<th>F1</th>
<th>Acc</th>
<th>F1</th>
<th>Acc</th>
<th>F1</th>
<th>Acc</th>
<th>F1</th>
<th>Acc</th>
<th>F1</th>
<th>Acc</th>
<th>F1</th>
<th>Acc</th>
<th>F1</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (Unigram)</td>
<td>0.59 0.73 0.47 0.52 0.41 0.37 0.4 0.32 0.4 0.33 0.37 0.33</td>
<td>0.62 0.77 0.48 0.53 0.4 0.37 0.41 0.33 0.4 0.34 0.38 0.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HMM</td>
<td>0.61 0.79 0.49 0.56 0.43 0.4 0.41 0.36 0.41 0.36 0.37 0.34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExtraTrees</td>
<td>0.68 0.83 0.57 0.57 0.47 0.41 0.49 0.37 0.45 0.37 0.40 0.35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AveragedPerceptron</td>
<td>0.72 0.83 0.52 0.57 0.45 0.40 0.43 0.36 0.43 0.37 0.41 0.34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BiLSTM</td>
<td>0.74 0.84 0.55 0.59 0.48 0.42 0.48 0.39 0.46 0.38 0.42 0.35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRF</td>
<td>0.77 0.87 0.56 0.59 0.49 0.43 0.51 0.39 0.47 0.39 0.44 0.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

POS tagging for Northern Kurdish. While this performance increase is in part due to the different annotation scheme, which is explained in section 4.2, the introduction of this richer scheme improved the performance of the POS models on specific POS tags other than IZAFE and DET. A detailed analysis of this improvement is reported in section 6.3.

Further observation reveals that within the context of the training on UD Kurmanji augmented, both the BiLSTM and AveragedPerceptron models exhibit identical accuracy scores, although their macro-averaged F1 scores diverge slightly but remain comparable. Conversely, when utilizing the UD Kurmanji original, a similar trend of identical accuracy emerges between the AveragedPerceptron and the CRF models. Additionally, it is notable that the HMM model falls behind, even when compared to the baseline.

Moreover, the NK-XLMR model is our best model as it outperforms all other models. This was an expected performance, and it is in line with our finding in section 3 where we showed how LLMs achieve state-of-the-art results for multiple NLP tasks, including POS tagging.

However, comparing the scores of NK-XLMR and CRF models in Table 4, we observe very close performance between the two. The difference is very small, 0.03 for the macro-averaged F1 and the accuracy scores. This is a notable result, especially with regard to the computational resources required for fine-tuning XLM-RoBERTa and for training the CRF model from scratch for the task of POS tagging. Based on our experiments in this study, fine-tuning XLM-RoBERTa for POS tagging took notably longer than training the CRF for the same task.

6.3. Analysis

The presented results in the Tables 4 and 3 unambiguously demonstrate two trends in our results. First, training the POS models on the UD Kurmanji augmented undeniably results in higher accuracy and F1 scores when compared with the outcomes of POS models trained on the UD Kurmanji original. Second, the performance of POS models tends to decline as we transition away from the initial. Second, the performance of POS models ambiguously demonstrate two trends in our results. First, training the POS models on the UD Kurmanji original, a similar trend of identical accuracy emerges between the AveragedPerceptron and the CRF models. Additionally, it is notable that the HMM model falls behind, even when compared to the baseline.

Moreover, the NK-XLMR model is our best model as it outperforms all other models. This was an expected performance, and it is in line with our finding in section 3 where we showed how LLMs achieve state-of-the-art results for multiple NLP tasks, including POS tagging.

However, comparing the scores of NK-XLMR and CRF models in Table 4, we observe very close performance between the two. The difference is very small, 0.03 for the macro-averaged F1 and the accuracy scores. This is a notable result, especially with regard to the computational resources required for fine-tuning XLM-RoBERTa and for training the CRF model from scratch for the task of POS tagging. Based on our experiments in this study, fine-tuning XLM-RoBERTa for POS tagging took notably longer than training the CRF for the same task.

6.3. Analysis

The presented results in the Tables 4 and 3 unambiguously demonstrate two trends in our results. First, training the POS models on the UD Kurmanji augmented undeniably results in higher accuracy and F1 scores when compared with the outcomes of POS models trained on the UD Kurmanji original. Second, the performance of POS models tends to decline as we transition away from the initial. Second, the performance of POS models ambiguously demonstrate two trends in our results. First, training the POS models on the UD Kurmanji original, a similar trend of identical accuracy emerges between the AveragedPerceptron and the CRF models. Additionally, it is notable that the HMM model falls behind, even when compared to the baseline.

Moreover, the NK-XLMR model is our best model as it outperforms all other models. This was an expected performance, and it is in line with our finding in section 3 where we showed how LLMs achieve state-of-the-art results for multiple NLP tasks, including POS tagging.

However, comparing the scores of NK-XLMR and CRF models in Table 4, we observe very close performance between the two. The difference is very small, 0.03 for the macro-averaged F1 and the accuracy scores. This is a notable result, especially with regard to the computational resources required for fine-tuning XLM-RoBERTa and for training the CRF model from scratch for the task of POS tagging. Based on our experiments in this study, fine-tuning XLM-RoBERTa for POS tagging took notably longer than training the CRF for the same task.
Leyla Qasim wanted to make the Kurdish voice heard in the world.

Figure 2: Outputs of the CRF and NK-XLMR compared to the gold annotations for a sentence from the gold-standard dataset (Translation: ‘Leyla Qasim wanted to make the Kurdish voice heard in the world’).

Figure 3: Confusion matrices of NK-XLMR (augmented) and NK-XLMR (original) models. Although both models exhibit inadequacy in handling the PART and ADV tags, NOUN and PROPN benefit from data augmentation.

The UD Kurmanji augmented is characterized by the enhancements we have introduced and discussed in detail in section 5.2. The data augmentation affected tokens from the following POS tags: NOUN, PROPN, DET, and ADP, which are important elements in the Izafe and oblique cases in Northern Kurdish.

By comparing the confusion matrices, we observe that NOUN and PROPN benefit the most from the data augmentation, demonstrating 0.05 and 0.06 accuracy improvement, respectively, and the ADP and VERB to a lesser extent. In addition, we see that the tags DET and Izafe enjoy huge improvement when trained on the UD Kurmanji augmented. However, we cannot consider it reliable since the Izafe tag was not present in the UD Kurmanji original.

Nevertheless, it is evident that the NK-XLMR (original and augmented) exhibits a notable inadequacy in handling the PART and ADV tags. Examined outputs of NK-XLMR(augmented) and the error rates presented in section 6.3 and section 6.3 also verify this inadequacy. The tag PART has an error rate of 1.0, which means the model completely fails in recognizing tokens belonging to this tag correctly. We argue that this can be attributed to a misalignment in the annotation schemes between the UD Kurmanji and ours rather than a lim-
The main objective of this study was to address the task of POS tagging for Northern Kurdish by utilizing the currently available resources. On the one hand, our multifaceted approach for this study enabled us to establish a baseline POS tagger for Northern Kurdish using the Uni-gram(augmented) model with an accuracy of 0.73 and a macro-averaged F1 score of 0.59 evaluated on the gold-standard dataset. On the other hand, the CRF(augmented) model achieves the second-best performance with 0.84 and 0.74 for accuracy and macro-averaged F1 score, making it the best-performing model among statistical POS tagging models. In addition, the CRF model stands out because of its quick training time.

The transformer-based NK-XLMR (augmented) outperforms all other models with an accuracy of 0.87 and a macro-averaged F1 score of 0.77, thus setting a new state-of-the-art performance for the task of POS tagging in Northern Kurdish. Our results are particularly robust compared to the work of Walther et al., 2010, where their POS tagger for Northern Kurdish was evaluated on only 13 sentences. This comparison underscores the reliability of our findings, considering the granularity of linguistic features in our gold-standard dataset and the larger number of test sentences (136 sentences) we used for evaluation.

Moreover, we further explored the impact of tokenization methods on POS tagging accuracy by comparing their outcomes against the gold standard tokens in our dataset. While encountering difficulties with certain linguistic nuances, the KLPT tokenizer demonstrated notable proficiency in capturing Northern Kurdish linguistic traits.

Finally, we successfully demonstrated the effect of the various linguistic features of Northern Kurdish, such as the Izafe and oblique case markers and contracted prepositions on the task by evaluating both variants of the models (original and augmented). Our POS tagging models trained on the UD Kurmanji augmented showed improvements on NOUN, PROPN, VERB, and ADP POS tags.

Limitations While this study has made several contributions to the field of Kurdish NLP, several limitations should be noted. Firstly, we did not target the task of syntactic parsing. Secondly, we did not explore the employment of LLMs or POS models from other closely related languages like Persian or dialects like Central Kurdish. Furthermore, we did not examine the impact of our POS tagging models and annotation schemes on other downstream tasks like named entity recognition, sentiment analysis, or parsing.

7. Conclusions and Discussion

The main objective of this study was to address the task of POS tagging for Northern Kur-
shop for NLP open source software (NLP-OSS), pages 72–84.


Hailiang Li, YC Adele, Yang Liu, Du Tang, Zhibin Lei, and Wenye Li. 2019. An augmented transformer architecture for natural language generation tasks. In 2019 International Conference on Data Mining Workshops (ICDMW), pages 1–7. IEEE.


Christopher D Manning. 2011. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In International conference on intelligent text processing and computational linguistics, pages 171–189. Springer.


Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging


Diachronic Analysis of Multi-word Expression Functional Categories in Scientific English

Diego Alves, Stefania Degaetano-Ortlieb, Elena Schmidt, Elke Teich
Saarland University
Saarbrücken, Germany
diego.alves@uni-saarland.de, s.degaetano@mx.uni-saarland.de
elsc00001@stud.uni-saarland.de, e.teich@mx.uni-saarland.de

Abstract
We present a diachronic analysis of multi-word expressions (MWEs) in English based on the Royal Society Corpus, a dataset containing 300+ years of the scientific publications of the Royal Society of London. Specifically, we investigate the functions of MWEs, such as stance markers ("it is interesting") or discourse organizers ("in this section"), and their development over time. Our approach is multi-disciplinary: to detect MWEs we use Universal Dependencies, to classify them functionally we use an approach from register theory, and to assess their role in diachronic development we use an information-theoretic measure, relative entropy.

Keywords: multi-word expressions, universal dependencies, relative entropy, discourse functions, diachronic analysis

1. Introduction
In this paper, we analyze multi-word expressions (MWEs) and the functions they fulfill in scientific writing, inspecting diachronic changes from the mid 17th century to today. From a communicative perspective, MWEs contribute to language efficiency as they constitute highly predictable linguistic material with a clear processing advantage for language users. Their use in scientific writing is particularly interesting due to the high informational load encountered within the scientific domain, where MWEs can act as devices to smooth the informational load in the signal (Conklin and Schmitt, 2012).

There has been a long-standing tradition to identify and analyze MWEs in scientific text and academic writing more widely, most prominently in research on English for Academic Purposes (EAP) (cf. Oakey (2020)). We combine this approach considering the Academic Formula List (AFL) with a UD-based approach, were we use the dependency relation label fixed to identify further MWEs not included in the AFL list. As it has been shown that scientific writing becomes increasingly conventionalized over time (see e.g. Degaetano-Ortlieb and Teich (2019)), the fixed MWEs are particularly important for a diachronic analysis aimed at investigating communicative efficiency. In this study, we focus on the most frequent grammaticalized fixed expressions identified from the RSC combined with a set of formulaic expressions commonly used in the scientific domain that can be considered as MWEs due to the statistical criteria defined by Simpson-Vlach and Ellis (2010).

Moreover, we label each identified MWE with functional categories to assess (a) the functions MWEs have fulfilled in scientific writing across 300 years, and (b) whether there have been changes in their usage over time. We derive the functions stance expressions, discourse organizers, and referential expressions from extensive previous work based on Hallidayan register theory (Halliday and Matthiessen, 2014) and widely used by EAP researchers (Biber et al., 2004; Simpson-Vlach and Ellis, 2010; Liu, 2012). Finally, to assess change regarding MWEs, we employ a method from language modeling, relative entropy (Kullback-Leibler Divergence).

The remainder of the paper is organized as follows. In Section 2 we discuss related work on functional categories of MWEs. Sections 3 and 4 present our methods and results. We conclude with a summary of our findings and perspectives for future work (Section 5).

2. Related Work
There are numerous corpus-based accounts regarding the usage of MWEs in different registers, including the scientific one (e.g. Biber and Barbieri (2007); Hyland (2008); Liu (2012)), considering also their classification in terms of functions (see Biber et al. (2004); Simpson-Vlach and Ellis (2010) and Oakey (2020) for an overview). These studies are usually based on strategies for identifying formulaic, pre-fabricated, chunk-like and otherwise phraseological linguistic items considering frequency-based measures (such as MPI) derived from corpora (see work on lexical bundles (Biber and Barbieri, 2007; Hyland, 2008), academic formulas (Simpson-Vlach and Ellis, 2010), and multi-word constructions (Liu, 2012)).
Computational linguistic accounts usually focus on techniques to identify and describe patterns of co-occurrence of linguistic units (e.g. Evert (2005); Gries (2022)). To identify potential MWE candidates different measures are applied. Gries (2022) proposes a strategy based on eight different dimensions of information, while Simpson-Vlach and Ellis (2010) define a formula teaching worth (FTW) score based on frequency and mutual information. The identification of MWEs using machine-learning methods are typically based either on DiMESUM (Schneider et al., 2016) or PARSEME (Savary et al., 2015) corpora and the complexity of this task can be attested by the low F1-scores of the state-of-the-art tools (i.e., below 65 as presented by Tanner and Hoffman (2023)). PARSEME corpus divides MWEs into different categories, but they are based on structural properties, not on their functions. Moreover, these datasets are not composed of scientific texts, and thus not totally suitable to address our research question.

Although the study of MWEs is a very active field, both from a linguistic and a computational point of view, the diachronic development of MWEs and their functions remains under-researched. While Biber et al. (2004) and Simpson-Vlach and Ellis (2010) propose a classification of MWEs in terms of discourse functions, these categories have not been examined diachronically. Alves et al. (2024) presented a study concerning the development of MWEs association metrics in scientific English, however, MWE functions were not the main focus of the analysis. Consequently, there are hardly any ready-to-use methodological approaches. With our work, we intend to fill these gaps.

### 3. Data and Methods

#### 3.1. Data

As a data source, we use the Royal Society Corpus (RSC) 6.0, a diachronic corpus of scientific English covering the period from 1665 until 1996. This resource comprises 47,837 texts (295,895,749 tokens), mainly scientific articles covering a wide range of areas from mathematics to physical and biological sciences, and is based on the Philosophical Transactions and Proceedings of the Royal Society of London (Fischer et al., 2020). Table 1 shows a detailed overview of the distribution of texts and tokens over time.

<table>
<thead>
<tr>
<th>Period</th>
<th>Texts</th>
<th>Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>1665–1699</td>
<td>1,325</td>
<td>2,582,856</td>
</tr>
<tr>
<td>1700–1749</td>
<td>1,686</td>
<td>3,414,795</td>
</tr>
<tr>
<td>1750–1799</td>
<td>1,819</td>
<td>6,342,489</td>
</tr>
<tr>
<td>1800–1849</td>
<td>2,774</td>
<td>9,112,274</td>
</tr>
<tr>
<td>1850–1899</td>
<td>6,754</td>
<td>36,993,412</td>
</tr>
<tr>
<td>1900–1949</td>
<td>10,011</td>
<td>65,431,384</td>
</tr>
<tr>
<td>1950–1996</td>
<td>23,468</td>
<td>172,018,539</td>
</tr>
</tbody>
</table>

Table 1: Size of the Royal Society Corpus 6.0 over time

Fixed Multi-word Expressions The Universal Dependencies (UD) guidelines for morphosyntactic annotations (De Marneffe et al., 2021) encompass the relation label fixed for certain fixed grammaticalized expressions which tend to behave like function words (e.g. because of, in spite of, as well as) with distinct functions.

To extract the fixed MWEs, we parsed the RSC 6.0 using Stanza tool (Qi et al., 2020) with the combined model for the English language trained on different UD corpora (i.e., EWT, GUM, GUMReddit, PUD, and Pronouns). Using a Python script with the pyconll library, we identified and counted the fixed MWEs in the RSC texts per year. From the list of fixed MWEs, we identified the 100 most frequent ones and manually annotated

---

1https://fedora.clarin-d.uni-saarland.de/rsc_v6/

2https://universaldependencies.org/

3https://github.com/pyconll/pyconll

4A manual evaluation of 70 sentences (10 per 50-year period of the RSC) showed that the labelled attachment score of the parser is equal or higher than 85% for fixed MWEs in the different time periods.
them according to the taxonomy in Section 3.2. Since we consider only the fixed MWEs with high frequency in the RSC and conducted a manual evaluation of the identified expressions, we assume that the parsing errors have been minimized in this study.

### AFL Multi-word Expressions
The Academic Formulas List is an inventory of the most common formulaic sequences in academic English. It is composed of: a) a core list of 207 formulaic expressions found in written and spoken academic language (e.g. *in terms of* and *at the same time*; b) 200 expressions from written corpora (e.g. *on the other hand and it should be noted*); and c) 200 MWEs extracted from spoken academic English texts (e.g. *be able to* and *if you look at*) (Simpson-Vlach and Ellis, 2010). The AFL MWEs were identified by the authors with a special measure of usefulness called the formula teaching worth (FTW), which combines frequency and mutual information measures. Thus, the classification of the formulaic expressions from the AFL as MWEs is done due this statistical criterion.

### 3.3. MWE Functional Categories

We follow the taxonomy proposed by Biber et al. (2004), which captures the major functions of MWEs with three primary categories: (a) stance expressions, which express attitudes or assessments of certainty, framing other propositions; (b) discourse organizers that reflect relationships between parts of the discourse; and (c) referential expressions that refer to physical or abstract entities, or to the textual context, identifying a specific entity or pointing out to a specific attribute of it.

Table 2 presents a summarized version of the taxonomy established by Biber et al. (2004) (i.e., functions and types) together with the number of MWEs per type and examples observed in the RSC.

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
<th>MWEs</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stance</td>
<td>epistemic</td>
<td>84</td>
<td>it is important, according to</td>
</tr>
<tr>
<td></td>
<td>attitudinal/modality</td>
<td>24</td>
<td>we have to, needs to be</td>
</tr>
<tr>
<td></td>
<td>ability</td>
<td>34</td>
<td>can be found, it is possible to</td>
</tr>
<tr>
<td>Discourse</td>
<td>topic introduction/focus</td>
<td>31</td>
<td>in this article, for example in</td>
</tr>
<tr>
<td></td>
<td>topic elaboration/clarification</td>
<td>70</td>
<td>due to the fact, the reason for</td>
</tr>
<tr>
<td>Reference</td>
<td>identification/focus</td>
<td>61</td>
<td>such as the, as can be seen in</td>
</tr>
<tr>
<td></td>
<td>imprecision</td>
<td>3</td>
<td>and so on, and so forth</td>
</tr>
<tr>
<td></td>
<td>specification of attributes</td>
<td>177</td>
<td>a form of, on the basis of</td>
</tr>
<tr>
<td></td>
<td>time/place/text reference</td>
<td>57</td>
<td>at the end of, in between</td>
</tr>
</tbody>
</table>

Table 2: Functional categories and types (cf. Biber et al. (2004)).

Note that Simpson-Vlach and Ellis (2010) classified most of the AFL MWEs according to a taxonomy similar to the one proposed by Biber et al. (2004). We selected these categorised MWEs to be examined in this study, adjusting the taxonomy according to Table 2.

### 3.4. Modeling Change with Relative Entropy

To analyse the diachronic development of the different MWE functional categories, first, we examined the relative frequency per year.

To detect evolutionary trends, we applied relative entropy, specifically Kullback-Leibler Divergence (KLD; Kullback and Leibler (1951)), a method for comparing probability distributions measuring the number of additional bits needed to encode a given data set A when a (non-optimal) model based on a data set B is used for a set of elements X. In our case, A and B correspond to sub-sets of the RSC (e.g. time slices) and X, i.e. the ensemble of MWEs of each function.

\[
D_{KL}(A||B) = \sum_{x \in X} A(x) \log \left( \frac{A(x)}{B(x)} \right)
\]

KLD provides an indication of the degree of divergence between corpora and identifies the features that are primarily associated with a difference.

To detect periods of change using KLD given each functional category (stance, discourse, and reference), we adopt the methodology described in Degaetano-Ortlieb and Teich (2018). Basically, we compare 20-year windows of past and present language use sliding with a 5-year gap over the time line (e.g. t1=1665-1685, t2=1691-1711). By plotting the divergence for each comparison on the time line, we can inspect peaks or troughs which

---

5 The annotation was made by a linguistics student and verified by two specialists.

6 Discrepancies regarding vocabulary size are controlled by applying Jelinek-Mercer smoothing with lambda 0.05 (cf. Zhai and Lafferty (2004) and Fankhauser et al. (2014)).

7 Degaetano-Ortlieb and Teich (2018) make the code available at: https://stefaniadegaetano.com/code/
indicates a change. A peak indicates that the divergence of that feature increases, and is thus typical of the future 20 years in comparison to the past 20 years. In particular, we consider the pointwise KLD, i.e. the individual KLD of each feature (here: either functions or types), in order to determine a feature’s rise or decrease in typicality.

4. Results

4.1. Frequency-based Trends

Figure 1 presents the evolution of each main functional category per year by relative frequency (i.e., MWEs occurrence/no. of tokens of each period).

In general, all three functions present an increasing tendency across time until the beginning of the twentieth century. The usage of referential expressions (black line) has a considerable increase in the second half of the eighteenth century. Moreover, from 1925 on, while both discourse (blue) and reference MWEs (red) present a decreasing tendency, the use of stance expressions seems to steadily increase even though these expressions remain relatively low in frequency.

![Figure 1: Relative frequency for each function.](image)

4.2. Diachronic Trends by Divergence

While relative frequencies pinpoint the rise or decline of specific linguistic features over time, KLD provides a detailed quantification of the overall linguistic shift from one period to another, identifying even those changes that do not correspond to simple increases or decreases in usage frequency. Thus, KLD provides insights into the degree of linguistic change and allows to identify more subtle patterns of linguistic evolution that relative frequencies alone may not discern. Figure 2 presents the overall results per category for all the MWEs (AFL and fixed). We can observe that from the 17th to the beginning of 20th century, reference and discourse MWEs tend to behave in opposite directions, i.e. when reference becomes typical, discourse goes down in typicality and vice versa, while stance MWEs present less change. The scenario changes in the 20th century when the presence of stance expressions in the corpus becomes more typical.

To better understand these diachronic trends, we also applied KLD considering the types of each function (see Figure 3). The main trends observed for discourse and referential expressions are due to the function types ‘topic elaboration/clarification’ and ‘specification of attributes types’, respectively. While the topic elaboration/clarification function is used to signal further explication providing a clearer understanding or additional information related to the topic being discussed (e.g. *in order to*, *as a result*, *the reason for*), the specification of attributes function type serves as a way to provide framing information (e.g. *the way which*, *the level of*, *these two*), i.e. essentially specifying or detailing characteristics, qualities, or attributes of a subject. These trends may be influenced by a variety of factors. Historical and cultural contexts that value explicit reasoning may lead to a preference for elaborate discourse, while changes in academic standards and expectations could necessitate a more precise specification of attributes. The rise of particular disciplines and interdisciplinary research, along with technological advancements that shape information dissemination, could also play significant roles.

Considering the increase in divergence for stance expression in the more contemporary period, we can observe that the peak is indicated by three out of four types for stance expressions. By 1825, ability becomes more typical showing an increased distinctive use (e.g. *can be used/found/expressed*), followed by attitudinal expressions until almost 100 years later where they decrease in divergence around the 1930s, when epistemic expressions (e.g. *according to*, *at least*) become typical. Around that period, also identification and focus reference expressions (e.g. *there has been, can be seen*) increase in typicality as well as topic and introduc-

![Figure 2: KLD measures for each function.](image)

![Figure 3: Overall results per category for all the MWEs (AFL and fixed).](image)
tion discourse organizers (e.g. *first of all, in this paper we*). During that period, there is also a peak in time, place and textual reference (e.g. *as shown in, shown in figure*). Overall, there is a trend towards a more varied distinctive use of MWE function types towards the more contemporary period. These trends seem to signal a use of MWEs to be increasingly inclined to articulate evidence-based reasoning as shown by MWEs such as *according to or as shown in*. These expressions serve to direct the reader’s attention to evidence or examples that support the argument being made, which is a fundamental aspect of scholarly work.

5. Conclusion and Future Work

In this paper we have presented an analysis of MWEs in scientific writing, tracing the evolution of their functions over a span of three centuries. Our investigation reveals a dynamic landscape of MWE usage, marked by significant shifts in function that reflect changing priorities and practices within the scientific community over time. In the initial stages, we observed a competitive relationship between discourse and reference functions of MWEs. This competition underscores the evolving nature of scientific discourse, as authors sought to balance the needs for clarity and precision with the demands of argumentation and discourse structuring. Towards the recent 100 years, our findings indicate a diversification in MWE functions, with stance expressions taking on a leading role. The shift towards epistemic stance, reference of identification/focus, of place/time/textual and discourse organizers of topic and introduction seems to be a means of directing the reader’s attention to evidence-based information.

Combining the AFL list with a UD-based approach to identify MWEs not covered by the AFL, allowed us to capture a broader range of conventionalized expressions that contribute to the diachronic trend of increasing conventionalization in scientific writing. The application of relative entropy as a methodological tool has further enriched our understanding of change over time, offering a quantitative measure of the shifts in MWE usage.

The functional categorization of MWEs, grounded in Hallidayan register theory, provides a solid theoretical framework for our analysis of functions and types. A limitation of our study is the uneven distribution of data across periods, with more material from recent periods, which may skew perceptions of MWE functionality and its evolution over time. Also, the diachrony of our data might present gaps within the AFL list. In future work, we aim to expand our research in three ways: (1) increase the number of MWEs related to the different functions and compare the obtained results with analysis of other domains; (2) model MWEs at the paradigmatic level by word embeddings to further increase coverage of items; (3) apply probabilistic measures of processing (e.g. surprisal) to gain insights on processing effects of conventionalization of MWEs. Overall, we aim to work towards gaining further insights into the complex ways MWEs serve the communicative needs of scientific writers and compare their usage across scientific domains and other registers.

Acknowledgements

This research is funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 232722074 – SFB 1102.
6. Bibliographical References


Lexicons Gain the Upper Hand in Arabic MWE Identification

Najet Hadj Mohamed12, Agata Savary3, Cherifa Ben Khelil14, Jean-Yves Antoine15, Iskander Keskes2, Lamia Belguith Hadrich2

LIFAT - University of Tours1
MIRACL - University of Sfax 2
LISN - University of Paris-Saclay3
EFREI Research Lab - University of Paris Panthéon Assas4
LIFO - University of Orleans5

najet.hadjmohamed@etu.univ-tours.fr, agata.savary@universite-paris-saclay.fr,
cherifa.ben-khelil@efrei.fr, jean-yves.antoine@univ-tours.fr, iskandarkeskes@fsegs.usf.tn,
lamia.belguith@fsegs.usf.tn

Abstract

This paper highlights the importance of integrating MWE identification with the development of syntactic MWE lexicons. It suggests that lexicons with minimal morphosyntactic information can amplify current MWE-annotated datasets and refine identification strategies. To our knowledge, this work represents the first attempt to focus on both seen and unseen of VMWEs for Arabic. It also deals with the challenge of differentiating between literal and figurative interpretations of idiomatic expressions. The approach involves a dual-phase procedure: first projecting a VMWE lexicon onto a corpus to identify candidate occurrences, then disambiguating these occurrences to distinguish idiomatic from literal instances. Experiments outlined in the paper aim to assess the efficacy of this technique, utilizing a lexicon known as LEXAR and the “parseme-ar” corpus. The findings suggest that lexicon-driven strategies have the potential to refine MWE identification, particularly for unseen occurrences.

Keywords: Multiword Expressions, Idiomatic Expressions, Literal vs. Figurative Meanings, Lexicon Augmentation, Arabic Language

1. Introduction

Multiword Expressions (MWEs) are a subject of interest across various fields related to language studies. They are part of each language’s lexicon, distinct from literal words due to their non-compositional, preconstructed nature. Recently, the identification and analysis of MWEs have garnered significant attention in the field of Natural Language Processing (NLP), owing to their prevalence and nuanced semantic complexities. Despite considerable efforts in MWE identification, researchers have encountered challenges in addressing the issue of unseen MWE instances1 (Taslimipoor et al., 2020; Pasquer et al., 2020b; Yirmibesoglu and Guengör, 2020; Kurfali, 2020). Savary et al. (2019) assert that to make substantial progress in MWE identification, it is imperative for the research community to integrate the identification process with the development of syntactic MWE lexicons. They advocate for lexicons that provide minimal morphosyntactic information, augmenting existing MWE-annotated corpora. This approach, they argue, complements traditional corpus-based methods with MWEs that occur rarely or never in MWE-annotated corpora. In this paper, we align ourselves with the same perspective, emphasizing the critical role of MWE lexicons in advancing MWE identification methodologies for Arabic language.

MWEs assume a unique and challenging role within this domain due to their non-compositionality and their ability to take on a figurative or literal meanings. For instance, the degree of transparency varies from one idiom to another. Thus, the following idiom is rather transparent: مَسَلَى الْعَرَقُ السَّرِيعَ (salk al-ṭarīq al-sarī) | lit. ‘to take the fast road’) ‘to choose the easier way’, i.e. it is easy to recover the motivation behind the image of taking a fast road. Conversely, in كَسَرَ السَيْفَ (kasara al-saif | lit. ‘broke the sword’) ‘to triumph over an opponent or a difficult circumstance’, the motivation for the image is unclear. Moreover, transparency can depend on the particular speaker’s knowledge. For instance, the literal reading (e.g. وَضَعَيَداً عَلَى الْجُرْحَ (lit. ‘to touch the wound’) ‘to evoke someone’s weakness’ is understandable for most speakers, while understanding the origin of the following idiom calls for historic and cultural knowledge: بِرَاءَةُ الْذِّبَحِ مَنْ ذَمَّهُ (brāʿī al-ṣīlah mn dm abn ʿīqūb | lit. ‘to have the innocence of the wolf from the Jacob’s son blood’) ‘to be innocent’.2

1 No other verbal multi-word expression containing the exact same set of lemmas has been annotated at least once in the training corpus.

2 This idiom relates to the story of Jacob and his broth-
Significant research has been dedicated to detecting metaphors and understanding idiomatic expressions. Metaphors are deliberately constructed to convey figurative meanings, while idiomatic expressions can be interpreted either literally or figuratively, depending on the context of use (Shutova, 2010; Mason, 2004; Liu and Hwa, 2017). The accurate processing of idiomaticity within textual sequences is fundamental in NLP, given that idiomatic expressions constitute a significant aspect of linguistic communication. Attaining high performance in this task holds the potential to enhance various downstream applications, including sentiment analysis, information retrieval, and machine translation (Hashempour and Villavicencio, 2020; Mohamed et al., 2023). In this paper, our main focus is on identifying MWEs using an Arabic lexicon, with the goal of capturing unseen expressions more effectively and reducing the ambiguity of literal interpretations. Thus, we are also interested in the challenge of distinguishing between these two interpretations, which is complicated by the fact that idioms often do not follow easily identifiable linguistic patterns, especially for the Arabic language, given that it is characterized by a fairly flexible word order (Hadj Mohamed et al., 2022). While our research primarily focuses on Arabic, we have also tested our model for the binary disambiguation of Potential Idiomatic Expression (PIE) task (see Section 2 on English and German languages. The paper is organized as follows: Section 2 provides a thorough review of existing literature on MWE identification. Section 3 focuses on MWE identification in Arabic. Following that, Section 4 elaborates on our methodology for MWE identification in Arabic, emphasizing the integration of lexicons and the disambiguation process, while Section 5 details the data used in our experiments. Finally, in Section 6, we present and analyze our experimental results.

2. Related work

A considerable amount of research has focused on MWE-specific tasks. In this paper we are primarily concerned with **MWE identification**, which consists in automatically annotating MWE occurrences in running text (Constant et al., 2017). Most approaches to this task are supervised, i.e. trained on manually annotated datasets, such as STREUSLE (Schneider and Smith, 2015) or PARSEME (Savary et al., 2018). Shared tasks such as DISMUM (Schneider et al., 2016) and PARSEME (Ramisch et al., 2020) boosted the development of such tools. MWE identifiers are then trained and evaluated on these corpora. For instance, two approaches to MWE identification within a transition system were compared in (Al Saied et al., 2019): one based on a multilayer perceptron and the second on a linear SVM. Both approaches utilize only lemmas and morphosyntactic annotations from the corpus and were trained and tested on PARSEME Shared Task 1.1 data (Ramisch et al., 2018). The approach in (Kurfal, 2020) leverages feature-independent models with standard BERT embeddings. mBERT was also tested, but with lower results. An LSTM-CRF architecture combined with a rich set of features: word embedding, its POS tag, dependency relation, and its head word is proposed in (Yirmibesoglu and Gungor, 2020). The main focus of PARSEME Shared Task 1.2 was the detection of the unseen Verbal Multiword Expressions (VMWEs) which is more challenging compared to the identification of seen VMWEs (Ramisch et al., 2018). Several systems participated in the shared task, including MTLB-STRUCT (Taslimipoor et al., 2020), TRAVIS-mono and TRAVIS-multi developed by Kurfal (2020). Seen2Unseen developed by Pasquer et al. (2020a), ERMI by Yirmibesoglu and Gungor (2020) and others. Notably, the MTLB-STRUCT system, which leverages multilingual BERT fine-tuned for joint parsing and MWE identification, achieved the top cross-lingual macro-average in the open track for both the identification of VMWEs and the subtask of identifying unseen VMWEs.

Since unseen VMWEs prove critically hard to identify, a natural idea would be to leverage the advances of **MWE discovery**, which consists finding new MWEs (types) in text corpora, and storing them for future use in a lexicon (Constant et al., 2017). Very many different approaches were devised for this task in the past, based on statistical association measures (Evert, 2005), parsing data (Seretan et al., 2011), lexico-syntactic constraints (Broda et al., 2008), possibly combined with the use of neural network (Pecina, 2010), etc.

An alternative approach in addressing unseen data, and the scarceness of MWE-annotated corpora in general, is to use existing **MWE lexicons**, extracted for instance from classical human-readable dictionaries (Kanclerz and Piasecki, 2022) or Wiktionary (Muzny and Zettlemoyer, 2013), possibly with example sentences contained therein (Tedeschi et al., 2022). Such a lexicon can be straightforwardly projected on a corpus by form/lemma matching. Each resulting word co-occurrence is then considered as a **potential idiomatic expression** (PIE), in the sense that it can be true idiomatic occurrence of a MWE, or just a literal/coincidental co-occurrence of the MWE component words.

The task of **binary disambiguation of PIEs** has been addressed by a number of works. Sporleder...
and Li (2009) propose a generalized method utilizing cohesion graphs, hypothesizing that a PIE is used figuratively if its removal improves cohesion. Liu and Hwa (2018) introduce a "literal usage metric" quantifying the literalness of a PIE, computed as the average similarity between words in the sentence and a literal usage representation. Ehren et al. used a 2-layer LSTM network to get latent representations for the verbal idiom tokens. These were then used in a fully connected layer to predict the class using softmax. They used pretrained static and contextualized word embeddings as an input for their model. In recent years, several shared tasks have been organized to advance research in binary PIE disambiguation. Notably, the Multilingual Idiomaticity Detection and Sentence Embedding shared task (Madabushi et al., 2022) has gained attention. It comprises two subtasks: (a) binary disambiguation of PIEs, and (b) semantic text similarity detection, including sentences with and without MWEs.

3. Arabic and MWEs processing

The "Arabic language" includes Modern Standard Arabic (MSA) and diverse Arabic dialects. MSA is used in religious texts, poetry, and formal writing, while dialects are spoken in everyday conversation. In this section, we provide an overview of MSA's distinctive characteristics and review previous research on the automatic processing of MWEs in Arabic, with a specific focus on MSA rather than dialectal forms.

In MSA, capitalization is absent, and the usage of punctuation marks is infrequent in contemporary Arabic texts. Additionally, this language commonly features long, complex sentences with right-to-left writing, often resulting in paragraphs that lack punctuation. Furthermore, as a Semitic language, Arabic exhibits a complex morphology. It uses concatenative morphology (agglutinated or compound words), where words are formed via a sequential concatenation process\(^3\). For example, the sentence ‘then they will write it’ is presented in Arabic as one word ‘فسكتونها’. Moreover, Arabic includes words that can be altered with diacritical marks, either above or below them, creating new words with distinct pronunciations and meanings, often similar to the original word. Consequently, texts lacking diacritical marks are prone to ambiguity.

In Arabic, as in German, the word order is flexible, allowing specific words in a sentence to be rearranged without altering its meaning. This adaptability is achieved through the language's use of case markers, particles, and other linguistic mechanisms to clarify word relationships, resulting in a more versatile syntax compared to languages with a more rigid word order. These unique features make Arabic a challenging language for NLP tasks.

Several studies and research have been conducted on Arabic Multiword Expressions (AMWEs). Attia (2006) explored AMWEs using a finite-state machinery and Lexical Functional Grammar (LFG). During processing, fixed and adjacent semi-fixed MWEs were scrutinized using lexical transducers, deconstructing one-word phrases into segments and integrating MWEs into spaced words. Syntactically flexible MWEs were handled by grammar rules as syntactically compositional but semantically non-compositional due to lexical selection rules. Attia et al. (2010) introduced a linguistic method based on regular expressions for extracting AMWEs from texts, with a specific focus on nominal AMWEs. Hawwari et al. (2014) compiled an AMWE list from 5,000 expressions extracted from dictionaries. (Al-Badrashiny et al., 2016) employed a paradigm detection method on the Arabic Treebank and Arabic Gigawords corpus, resulting in the autonomous extraction of 1,884 AMWEs, each displaying various forms due to morphological variations. Recently, as part of the PARSEME framework (Savary et al., 2023), Hadj Mohamed et al. (2022) manually constructed a corpus comprising 4,700 instances of Verbal AMWEs.

4. Method

Our ultimate goal is to address the task of identifying VMWEs in Arabic. However, within this paper, we specifically concentrate on the critical challenge of detecting unseen instances, which represents a significant frontier in the field. Our approach relies on a lexicon and minimizes noise by filtering out literal interpretations. In contrast to numerous existing methods for VMWE identification, we choose not to rely on a VMWE-annotated corpus, opting instead for a carefully curated VMWE list. This decision stems from the limited representation of MWEs with literal and figurative meanings in resources such as Arabic Wiktionary, leading us to manually extract VMWEs from an exhaustive paper dictionary. Given this VMWE lexicon, our methodology unfolds in two phases: the first is the identification of VMWE candidates, while the second involves the disambiguation of these candidate occurrences, as outlined by Algorithm (1). We start by aligning the VMWE lexicon with the test corpus to identify potential VMWE candidates within the text. This process involves comparing the lexicon entries with the content of the

\(^3\)Agglutination is the process, common in Arabic, of adjoining clitics from simple word forms to create more complex forms.
test corpus in order to detect instances where VMWEs may occur. Then, we apply a binary PIE disambiguation method to distinguish between idiomatic and literal instances among these candidates. VMWEs are identified from idiomatic occurrences, while literal instances are retained for further analysis as supplementary data.

The following sections provide more detailed descriptions of these two phases.

**Algorithm 1**: Procedure for extracting and filtering sentences containing MWEs from the corpus

```plaintext
1: procedure EXTRACTANDFILTER(C', L, model)
2:   literal ← []
3:   idiomatic ← []
4:   for mwe ∈ L do
5:     for sentence ∈ C do
6:       if mwe occurs in sentence then
7:         class ← PIEC(mwe, sentence)
8:         if classification is "idiomatic" then
9:             literal.append(sentence)
10:        else
11:            idiomatic.append(sentence)
12:       end if
13:     end for
14:   end for
15: end procedure
```

### 4.1. Identifying VMWE candidates

During this phase, VMWE candidates are identified based on the lemmas associated with each MWE in the lexicon. The use of multisets allows for the identification of candidates in any order, regardless of the syntactic dependency between them. For example, consider the first VMWE seen in the lexicon (L) in Figure 1: "وضع يده" (lit. ‘put hand+his’) ‘put one’s hand’.

In sentences (1) and (2) from the parses-sent corpus, the three lemmas "وضع" (‘to put’), "يد" (‘hand’), and "ه" (‘his’) are present, resulting in their extraction as VMWE candidates. However, sentence (2) contains no VMWEs but rather a coincidental occurrence. In contrast, the candidate identified from sentence (4) represents a literal occurrence for the third VMWE "طيار غرابه" (lit. ‘his crow flew off’) “to get old” in L. The choice of using a forward step of filtering is a matter of balance between precision and recall. The expected noise present in the identification phase results in good recall (R=0.79) but low precision (P=0.41). Addressing this challenge, the second filtering phase (4.2) aims to enhance precision. We achieve this through the implementation of subtask (A) of the SemEval shared task (Madabushi et al., 2022).

### 4.2. Disambiguating candidate VMWE occurrences

As previously stated, we proceed with our filtering phase by employing the same subtask (A) from the SemEval shared task. The aim here is to distinguish between the compositional (literal) and non-compositional (idiomatic) uses of PIE within a given context. This is different from the task of MWE extraction, which focuses on identifying VMWEs within a corpus. Namely, our method takes a set of sentences containing a target PIE as input. We handle the disambiguation of PIES in a manner similar to word sense disambiguation. Our fundamental assumption is that the context in which PIES are used literally and figuratively differs significantly enough to justify distinct contextual representations. Figure 2 outlines an overview of the architecture, which is built upon the contextual language model used in our experiments, namely BERT.

Firstly, we aim to leverage the semantic idiosyncrasy characteristic of idiomatic expressions, highlighting that the meanings of the components within idiomatic expressions are related to the context in which they appear. To achieve this, we start by tokenizing the input, which consists of the sequence S and the target PIE. Following this, contextualized embeddings are generated using BERT and produce a vector representation for both the expression (PIE) and its context (S). Then, we add a Bidirectional LSTM (BiLSTM) layer for each embedding sequence to extract initial features from the raw embeddings. This results in $h^{S} = \text{BiLSTM}(e^{S})$ and $h^{\text{PIE}} = \text{BiLSTM}(e^{\text{PIE}})$.

The attention flow layer integrates and combines information from both the context word sequence and the query word sequence (Seo et al., 2017). This process generates query-aware vector representations of the context words and propagates the word embeddings from the preceding layer. Similarly, in our specific task, the attention flow layer merges details from two embedding sequences that encode diverse types of information. We fused $h^{S}$ and $h^{\text{PIE}}$ into an attention layer to obtain an enhanced contextualized representations for both the sentence and the PIE. This results in a unified representation that integrates information from both the entire sentence and the PIE. Finally, we introduce a MaxPooling layer to reduce spatial dimensions in neural network architectures while preserving the most important features by selecting the maximum value from each feature map. Following this, the fused representation is passed through a series of Dense layers for classification.

The final output is produced by a sigmoid-
Figure 1: Overview of the method.

Figure 2: Overview of the PIEC model
activated Dense layer, providing a binary classification result (idiomatic or literal). Table 1 shows the hyper-parameters use with this architecture.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence Length</td>
<td>128</td>
</tr>
<tr>
<td>Training Batch Size</td>
<td>256</td>
</tr>
<tr>
<td>Epoch number</td>
<td>30</td>
</tr>
<tr>
<td>Learning Rate</td>
<td>0.00001</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
</tr>
</tbody>
</table>

Table 1: Model Training Parameters

5. Data
Assessing the efficacy of our MWE identification method necessitates both a VMWE lexicon and a corpus. As for the corpus, we used the “parseme-ar” corpus from PARSEME 1.3 (Hadj Mohamed et al., 2022; Savary et al., 2023), which contains 4,7000 VMWEs within 7,500 sentences extracted from PADT belonging to the UD collection (Hajic et al., 2009). In our experiments, our focus was on two categories of VMWEs outlined in the parseme-ar corpus: LVC (Light Verb Construction) and VID (Verbal Idiom). We excluded the IAV (In Inherently Adpositional Verb) category, as it is considered optional. Following this, we manually created a lexicon named LEXAR⁶, referenced as (L) in Figure 1. We meticulously extracted and compiled idiomatic expressions from “Contextual Dictionary of Idiomatic Expressions” by Elsini (1998). Following the PARSEME annotation guidelines⁵, we identified a total of 1504 Arabic VMWEs, and each expression in LEXAR underwent categorization by assigning a part-of-speech (POS) tag and determining its type as either LVC or VID. The annotation process, which took between 1-2 days and overlapped almost 70% of VMWEs with PARSEME-AR, ensured a comprehensive coverage of VMWEs in our corpus. We evaluated the performance of our idiomatic expression classifier, PIEC, by conducting evaluations with specialized datasets tailored to measure its accuracy in classifying sentences with idiomatic expressions. These evaluations encompassed datasets in Arabic, German, and English languages. Table 2 provides a summary of the data used to evaluate the secondary task. For Arabic, we trained the PIEC on a dataset included 34 idiomatic expressions. Each expression accompanied by sentences from the corpus of the shared task ConLL⁷, encompassing both idiomatic and literal meanings. The 34 expressions were crafted manually by two native Arabic speakers. For instances lacking literal examples, we used ChatGPT to generate them, followed by manual verification. The MAGPIE corpus (Haagsma et al., 2020) provided the English dataset. It offers a collection of 1,756 PIEs, each representing different syntactic patterns, along with their associated sentences, totaling 56,622 annotated data instances with an average of 32.24 instances per PIE. For German we used the COLF-VID dataset (CORpus of Literal and Figurative meanings of Verbal IDioms) (Ehren et al., 2020). It contains 6,985 sentences sourced from newspaper articles, with annotations for 34 German VID types. Each MWE in the dataset is tagged with one of four labels: IDIOMATIC, LITERAL, UNDECIDABLE, or BOTH.

6. Results
The main goal of this study is to identify VMWEs, with a particular emphasis on unseen instances. Accordingly, we employed evaluation metrics aligned with the criteria of the shared task (Savary et al., 2017): These metrics include MWE-based metrics, which encompass precision, recall, and F1 scores for accurately detecting entire VMWEs, as well as precision, recall, and F1 measures for all VMWEs, including those that are unseen (unseen MWE-based). In Table 3, we compare the performance of our approach against MTLB-STRUCT.

On the multilingual level, MTLB-STRUCT achieved an MWE-based F1 score of 34.24 on unseen VMWEs and a global MWE-based F1 score of 56.27. Note that these results were obtained by re-training MTLB-STRUCT on the parseme-ar without the IAV category. However, even with the improvement in scores generated by the AraBert-based model (F1= 0.62 on the dev), Arabic is still one of the languages with the lowest performance score for global MWE-based and unseen-based scores. Although the F1 scores for unseen MWEs are still not optimal, our approach outperforms MTLB-STRUCT in terms of MWE-based F1 score by 7% and for unseen MWEs by 9%. Among the 278 unseen VMWEs assessed, our approach detected 125, whereas MTLB-STRUCT identified 104 out of the total.

For our experiments on the binary disambiguation of PIEs task (Figure 2), we focused only on the IDIOMATIC and LITERAL labels. Table 4 presents the results of our experiments on the TEST set. As baseline, we used a conventional SVM (Support Vector Machine) with MUSE (Multilingual Unsupervised and Supervised Embeddings) (Conneau et al., 2018) features. Em-
Table 2: Literal and idiomatic occurrences of PIEs in Arabic (AR), German (DE) (we excluded both the types of BOTH and UNDECIDABLE, which accounts for the disparity in the count between literal and idiomatic expressions compared to the total) and English (EN)

<table>
<thead>
<tr>
<th>Lang</th>
<th>Literal</th>
<th>Figurative</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR-train</td>
<td>103</td>
<td>202</td>
<td>305</td>
</tr>
<tr>
<td>AR-dev</td>
<td>16</td>
<td>30</td>
<td>46</td>
</tr>
<tr>
<td>AR-test</td>
<td>29</td>
<td>57</td>
<td>86</td>
</tr>
<tr>
<td>COLF-VID-train</td>
<td>1,172</td>
<td>5,705</td>
<td>6,902</td>
</tr>
<tr>
<td>COLF-VID-dev</td>
<td>264</td>
<td>1,214</td>
<td>1,488</td>
</tr>
<tr>
<td>COLF-VID-test</td>
<td>265</td>
<td>1,238</td>
<td>1,511</td>
</tr>
<tr>
<td>MAGPIE-train</td>
<td>2,676</td>
<td>12,676</td>
<td>15,352</td>
</tr>
<tr>
<td>MAGPIE-dev</td>
<td>595</td>
<td>2719</td>
<td>3314</td>
</tr>
<tr>
<td>MAGPIE-test</td>
<td>635</td>
<td>3339</td>
<td>3974</td>
</tr>
</tbody>
</table>

7. Conclusion

This paper introduces a simple yet impactful strategy for improving the identification of VMWE through the integration of lexicons, with our lexicon named LEXAR. Specifically focusing on the Arabic language, we demonstrate that our approach outperformed neural architectures like MTLB-STRUCT. Additionally, our method effectively addresses the challenge of binary disambiguation by employing contextual embeddings, which differentiate between various uses of the same lexical units and assign appropriate representations. Although detecting unseen MWEs proves to be a challenging task in our experiments, we achieve promising results using lexicons, surpassing the previous state-of-the-art. Moreover, our proposed model for the binary disambiguation of PIEs task shows significant potential for extension to multiple languages, facilitated by multilingual contextual embeddings.

Acknowledgement

We would like to express our sincere gratitude to Rafael Ehren and Laura Kallmeyer for graciously accepting me (Najet Hadj Mohamed) to undertake a short-term scientific mission at Heinrich Heine University Düsseldorf. We especially thank Rafael Ehren for providing the preprocessed English data, which significantly contributed to the completion of this research. Additionally, we extend our appreciation to UniDive, the CA21167 COST Action[8]. Universality, diversity and idiosyncrasy in language technology for their support and funding, which facilitated this study.

8. Bibliographical References


Mohammed A. Attia. 2006. Accommodating multiword expressions in an Arabic LFG grammar. In

[8]https://www.cost.eu/actions/CA21167/
<table>
<thead>
<tr>
<th>Lang</th>
<th>Our approach</th>
<th>MTLB-STRUCT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MWE-based</td>
<td>unseen MWE-based</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>Arabic</td>
<td>64.87</td>
<td>61.91</td>
</tr>
</tbody>
</table>

Table 3: Comparing our approach performance with MTLB-STRUCT on MWE-based and unseen MWE-based metrics.

<table>
<thead>
<tr>
<th>Lang</th>
<th>SVM-MUSE</th>
<th>PIEC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Literal</td>
<td>Figurative</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>Arabic</td>
<td>0.40</td>
<td>0.50</td>
</tr>
<tr>
<td>English</td>
<td>0.81</td>
<td>0.26</td>
</tr>
<tr>
<td>German</td>
<td>0.79</td>
<td>0.41</td>
</tr>
</tbody>
</table>

Table 4: Comparing SVM-MUSE and PIEC performance across 3 languages in term of Precision (P), Recall (R), and F-measure (F1).

International Conference on Natural Language Processing (in Finland), pages 87–98. Springer.


Murathan Kurfali. 2020. Travis at parseme shared task 2020: How good is (m) bert at see-
ing the unseen? In International Conference on Computational Linguistics (COLING), Barcelona, Spain (Online), December 13, 2020, pages 136–141.


Pavel Pecina. 2010. Lexical association measures and collocation extraction. Language resources and evaluation, 44:137–158.


Revisiting VMWEs in Hindi: Annotating Layers of Predication

Kanishka Jain, Ashwini Vaidya
Department of Humanities and Social Sciences
Indian Institute of Technology, Delhi
{huz218481, avaidya}@iitd.ac.in

Abstract

Multiword expressions in languages like Hindi are both productive and challenging. Hindi not only uses a variety of verbal multiword expressions (VMWEs) but also employs different combinatorial strategies to create new types of multiword expressions. In this paper we are investigating two such strategies that are quite common in the language. Firstly, we describe that VMWEs in Hindi are not just lexical but also morphological. Causatives are formed morphologically in Hindi. Second, we examine Stacked VMWEs i.e. when at least two VMWEs occur together. We suggest that the existing PARSEME annotation framework can be extended to these two phenomena without changing the existing guidelines. We also propose rule-based heuristics using existing Universal Dependency annotations to automatically identify and annotate some of the VMWEs in the language. The goal of this paper is to refine the existing PARSEME corpus of Hindi for VMWEs while expanding its scope giving a more comprehensive picture of VMWEs in Hindi.

Keywords: Annotation, Stacked VMWE, Morphological Causative

1. Introduction

Verbal multiword expressions are linguistic constructions that involve multiple verbs or a combination of verb and other lexical item(s). These expressions combine to form new meanings (Baldwin and Kim, 2010). However, the non-compositional nature of these multiword expressions pose a challenge to any kind of natural language processing (NLP) task. Therefore, they have been part of multiple annotation efforts across languages.

The PARSEME shared task (Ramisch et al., 2020, 2018) is one such effort that aims to identify and annotate different types of VMWEs in multiple languages. We examine the Hindi corpus from the PARSEME shared task (Ramisch et al., 2020). In this paper, we have conducted a detailed survey of the corpus and identified some problems. A prominent issue that was prevalent across all annotation categories was missing annotations for a number of expressions. Another repeated issue that we observed is the annotation of modal constructions as multi-verbal constructions (MVCs) as both are structurally similar to each other. We address these and other issues in the existing corpus and refine the annotations to create a better quality dataset.

Multiword expressions in some languages are highly frequent. Hindi, for instance, in comparison to languages like English, is known to have a greater proportion of VMWEs compared to simple verbs (Vaidya et al., 2016). This productive usage of multiword expressions in the language has been captured in the PARSEME corpus edition 1.3 (Savary et al., 2023). But two additional and quite common phenomena need to be addressed. In Hindi, verbal complex allows for recursive combinations of light verb, multi-verb, and causative verbs. Sometimes all three can combine together. When two VMWEs appear together to create a single predicate then we refer to such predicate as Stacked VMWE. Further, VMWEs in Hindi are formed not only lexically (i.e. combining two or more lexical items) but also morphologically (i.e. combining two or more morphemes). In Hindi, morphological VMWEs occur as Causatives. Both, stacked and causative VMWEs are extensively used in the language but have not been explicitly annotated as such within existing annotation frameworks of the language.

The aim of this paper is twofold. First, to refine the existing corpus by addressing various issues and second, to extend its scope.

The paper is organized as follows. In Section 2 we describe different types of VMWEs found in Hindi. We also describe causatives and stacked VMWEs. Section 3 discusses the issues found in the annotations and how they have been addressed in the present study. Results and conclusion are presented in Section 4.

2. VMWEs in Hindi

2.1. PARSEME VMWEs

The PARSEME framework (Ramisch et al., 2020, 2018) has five categories of verbal multiword expressions (VMWEs) out of which three are tagged for Hindi i.e. Light Verb construction (LVC) as LVC.full and LVC.cause, Multi-Verb Construction (MVC), and Verbal Idiom (VID). The fundamental
difference among these categories lie in terms of their predication strategy. A VID has at least two elements combining – a main verb and its dependent which is not restricted to any one particular lexical category as shown in (1). On the contrary, LVC and MVC are formed with a preverbal element and a light verb. The only difference between the two categories is that the preverbal element in an LVC is noun whereas in case of an MVC it is a verb as shown in (2) and (3), respectively.

(1) bāṛī mehengai par lāgam
increasing.F price-hike,F on
lāgana zaruri he
put.INF important.F be,PRS
‘It is important to control the price-hike (or inflation).’

(2) lāṛke-ne gehnō-ki
boy.3.SG.M-ERG jewellery.PL.M-GEN.F
CORI ki
theft.F do,PST.M
‘The boy has stolen the jewellery.’

(3) lāṛke-ne pari
boy.3.SG.M-ERG book.SG.F read
li
take.PST.SG.F
‘The boy read the book (completely).’

Further, as mentioned above LVCs have been distinguished as LVC.full and LVC.cause. The difference is made in terms of the type of light verb used. If the light verb is ‘causative’ such that the subject is the cause of an event then it has been annotated as LVC.cause else as a LVC.full. An example is shown in (4). Compare it with its non-causative counterpart in (2). The subject lāṛka/‘boy’ is the cause of an event of theft in (4) but an agent in (2). The causative meaning is expressed by the /-va/ morpheme on the verb in (4).

(4) lāṛke-ne naukar-se
boy.3.SG.M-ERG servant.3.SG.M-INST
gehnō-ki cori
jewellery.PL.M-GEN.F theft.F
kār-va-yi
icaus-PST.PERF.SG.F
‘The boy made the servant steal the jewellery.’

In the existing PARSEME corpus of Hindi a total of 1034 VMWEs have been annotated out of 35430 tokens as shown in Table 1. Further, it is to be noted that the frequency of VMWEs when compared to other Indo-European languages is quite high. These number are compiled from PARSEME shared tasks 2020\textsuperscript{2} and 2018\textsuperscript{3}.

While the existing PARSEME framework covers all the prominent categories of VMWEs in Hindi, there are additional phenomena that are not present. The rest of the paper discusses two such phenomena – stacked VMWEs and causatives.

### 2.2. Morphological Causative

Causatives are common across natural languages. This is especially true for South-Asian languages like Hindi where any verb, theoretically, can undergo the morphological process and form causative. For instance, in (5b) the causative marker /-va/ attaches to the transitive verb /banana/ ‘build’ and forms causative /banvaṇa/. The causativization of the transitive verb in (5a) increases the valency from two to three.

(5) a. lāṛke-ne ghar
boy.3.SG.M-ERG house-3.M
banaya
build.PST.PERF.SG.M
‘The boy built a house.’

b. lāṛke-ne bacci-se
boy.3.SG.M-ERG girl.3.SG.F-INST
ghar	house.3.M
ban-va-ya
icaus-PST.PERF.SG.M
‘The boy made the girl build the house.’

Apart from causativizing a simple verb, the language also allows causativization of light verbs\textsuperscript{4} as shown in (4) where the light verb /ki/ ‘do’ is a causative.

Valency change is a property that is common to LVCs, MVCs and morphological causatives (Butt and King, 2006; Butt et al., 2008; Butt, 2010). For instance in (6a) simple verb /katna/ ‘cut’ has two argument positions – the servant and the tree. But in (6b) when katna/ combines with the light verb /dena/ ‘give’, forming an MVC, it has three argument positions. The new argument position for /lāṛka/ ‘boy’ is licensed by the light verb /dena/ (Butt, 2010).

\textsuperscript{2}http://multiword.sourceforge.net/mwelex2020

\textsuperscript{3}http://multiword.sourceforge.net/lawmwcxg2018

\textsuperscript{4}According to (Butt et al., 2008), Hindi also allows for causatives in MVC construction but we did not find examples of this in the current corpus.
This valency change is similar to causatives in example (5) where /-va/ morpheme combines with verb and license a new argument position for the causer ‘girl’. This provides evidence that morphological VMWEs are similar to lexical VMWEs in Hindi. Hence, we propose to include them in the PARSEME framework.

PARSEME’s existing annotation schema already annotates example like (4) as LVC.cause distinguishing them from their non-causative counterpart as in example (2) annotated as LVC.full. The addition of other causatives will then give a comprehensive picture of VMWEs in this language.

The examples discussed so far captures only one kind of causatives i.e. a causative formed by attaching /-va/ morpheme. They are also known as ‘indirect causatives’. However, Hindi also has direct causatives that are formed by causativization of intransitive verbs as exemplified in (7).

In (7a), the verb /jali/ ‘burn’ in intransitive whereas in (7b) the direct causative marker /-a/ is attached to the verb and forms the causative /jalanə/. Direct causatives, similar to indirect causatives, change the valency of the base verb from single argument place to two argument places. Therefore, direct causatives are also an example of morphologically formed multiword expressions.

In Hindi, direct causatives for some verbs are realized by a change in the phonological realization of the root of the verb as in (8) where the verb /dho/ ‘wash’ changes to causative /dha/o/.

These examples show that the system of morphological predication in the language is quite robust and complex. It is, therefore, essential to capture these various kinds of morphological multiword expressions to understand the representation of different types of VMWEs in Hindi. Hence, in this work we propose to annotate causatives using a morphological feature ‘Cause’ on verbs (see Section 3). The feature ‘Cause’ can effectively differentiates between the causative and non-causative forms of the verbs.

### 2.3. Recursive VMWEs

VMWEs in Hindi are not limited to combining two lexical items or morphological items but due to their recursive nature allow two or more VMWEs to stack describing a single event (Butt et al., 2003). An example is shown in (9) where an MVC is stacked on an LVC and results in a Stacked VMWE.

In (7a), the verb /jali/ ‘burn’ in intransitive whereas in (7b) the direct causative marker /-a/ is attached to the verb and forms the causative /jalanə/. Direct causatives, similar to indirect causatives, change the valency of the base verb from single argument place to two argument places. Therefore, direct causatives are also an example of morphologically formed multiword expressions.

In Hindi, direct causatives for some verbs are realized by a change in the phonological realization of the root of the verb as in (8) where the verb /dho/ ‘wash’ changes to causative /dha/o/.

These examples show that the system of morphological predication in the language is quite robust and complex. It is, therefore, essential to capture these various kinds of morphological multiword expressions to understand the representation of different types of VMWEs in Hindi. Hence, in this work we propose to annotate causatives using a morphological feature ‘Cause’ on verbs (see Section 3). The feature ‘Cause’ can effectively differentiates between the causative and non-causative forms of the verbs.

### 2.3. Recursive VMWEs

VMWEs in Hindi are not limited to combining two lexical items or morphological items but due to their recursive nature allow two or more VMWEs to stack describing a single event (Butt et al., 2003). An example is shown in (9) where an MVC is stacked on an LVC and results in a Stacked VMWE.

<table>
<thead>
<tr>
<th>Language</th>
<th>Tokens</th>
<th>VID</th>
<th>LVC.full</th>
<th>LVC.cause</th>
<th>MVC</th>
<th>Others</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>124203</td>
<td>139</td>
<td>244</td>
<td>43</td>
<td>4</td>
<td>402</td>
<td>832</td>
</tr>
<tr>
<td>French</td>
<td>525992</td>
<td>2156</td>
<td>1878</td>
<td>97</td>
<td>22</td>
<td>1501</td>
<td>5654</td>
</tr>
<tr>
<td>German</td>
<td>173562</td>
<td>1437</td>
<td>311</td>
<td>33</td>
<td>0</td>
<td>2260</td>
<td>4041</td>
</tr>
<tr>
<td>Hindi</td>
<td>35430</td>
<td>61</td>
<td>641</td>
<td>26</td>
<td>306</td>
<td>0</td>
<td>1034</td>
</tr>
<tr>
<td>Italian</td>
<td>430789</td>
<td>1484</td>
<td>734</td>
<td>174</td>
<td>33</td>
<td>1785</td>
<td>4210</td>
</tr>
</tbody>
</table>

Table 1: Number of VMWEs in different Indo-European languages including Hindi in PARSEME shared tasks.

In (7a), the verb /jali/ ‘burn’ in intransitive whereas in (7b) the direct causative marker /-a/ is attached to the verb and forms the causative /jalanə/. Direct causatives, similar to indirect causatives, change the valency of the base verb from single argument place to two argument places. Therefore, direct causatives are also an example of morphologically formed multiword expressions.

In Hindi, direct causatives for some verbs are realized by a change in the phonological realization of the root of the verb as in (8) where the verb /dho/ ‘wash’ changes to causative /dha/o/.

These examples show that the system of morphological predication in the language is quite robust and complex. It is, therefore, essential to capture these various kinds of morphological multiword expressions to understand the representation of different types of VMWEs in Hindi. Hence, in this work we propose to annotate causatives using a morphological feature ‘Cause’ on verbs (see Section 3). The feature ‘Cause’ can effectively differentiates between the causative and non-causative forms of the verbs.

2.3. Recursive VMWEs

VMWEs in Hindi are not limited to combining two lexical items or morphological items but due to their recursive nature allow two or more VMWEs to stack describing a single event (Butt et al., 2003). An example is shown in (9) where an MVC is stacked on an LVC and results in a Stacked VMWE.

In (7a), the verb /jali/ ‘burn’ in intransitive whereas in (7b) the direct causative marker /-a/ is attached to the verb and forms the causative /jalanə/. Direct causatives, similar to indirect causatives, change the valency of the base verb from single argument place to two argument places. Therefore, direct causatives are also an example of morphologically formed multiword expressions.

In Hindi, direct causatives for some verbs are realized by a change in the phonological realization of the root of the verb as in (8) where the verb /dho/ ‘wash’ changes to causative /dha/o/.

These examples show that the system of morphological predication in the language is quite robust and complex. It is, therefore, essential to capture these various kinds of morphological multiword expressions to understand the representation of different types of VMWEs in Hindi. Hence, in this work we propose to annotate causatives using a morphological feature ‘Cause’ on verbs (see Section 3). The feature ‘Cause’ can effectively differentiates between the causative and non-causative forms of the verbs.

2.3. Recursive VMWEs

VMWEs in Hindi are not limited to combining two lexical items or morphological items but due to their recursive nature allow two or more VMWEs to stack describing a single event (Butt et al., 2003). An example is shown in (9) where an MVC is stacked on an LVC and results in a Stacked VMWE.

In (7a), the verb /jali/ ‘burn’ in intransitive whereas in (7b) the direct causative marker /-a/ is attached to the verb and forms the causative /jalanə/. Direct causatives, similar to indirect causatives, change the valency of the base verb from single argument place to two argument places. Therefore, direct causatives are also an example of morphologically formed multiword expressions.

In Hindi, direct causatives for some verbs are realized by a change in the phonological realization of the root of the verb as in (8) where the verb /dho/ ‘wash’ changes to causative /dha/o/.

These examples show that the system of morphological predication in the language is quite robust and complex. It is, therefore, essential to capture these various kinds of morphological multiword expressions to understand the representation of different types of VMWEs in Hindi. Hence, in this work we propose to annotate causatives using a morphological feature ‘Cause’ on verbs (see Section 3). The feature ‘Cause’ can effectively differentiates between the causative and non-causative forms of the verbs.

2.3. Recursive VMWEs

VMWEs in Hindi are not limited to combining two lexical items or morphological items but due to their recursive nature allow two or more VMWEs to stack describing a single event (Butt et al., 2003). An example is shown in (9) where an MVC is stacked on an LVC and results in a Stacked VMWE.

In (7a), the verb /jali/ ‘burn’ in intransitive whereas in (7b) the direct causative marker /-a/ is attached to the verb and forms the causative /jalanə/. Direct causatives, similar to indirect causatives, change the valency of the base verb from single argument place to two argument places. Therefore, direct causatives are also an example of morphologically formed multiword expressions.

In Hindi, direct causatives for some verbs are realized by a change in the phonological realization of the root of the verb as in (8) where the verb /dho/ ‘wash’ changes to causative /dha/o/.

These examples show that the system of morphological predication in the language is quite robust and complex. It is, therefore, essential to capture these various kinds of morphological multiword expressions to understand the representation of different types of VMWEs in Hindi. Hence, in this work we propose to annotate causatives using a morphological feature ‘Cause’ on verbs (see Section 3). The feature ‘Cause’ can effectively differentiates between the causative and non-causative forms of the verbs.

2.3. Recursive VMWEs

VMWEs in Hindi are not limited to combining two lexical items or morphological items but due to their recursive nature allow two or more VMWEs to stack describing a single event (Butt et al., 2003). An example is shown in (9) where an MVC is stacked on an LVC and results in a Stacked VMWE.
In (9) there are three elements unlike the common pattern observed in LVCs and MVCs of predicting two elements. There is a noun /'do' as well as /'put'. The first or main verb can be in its base form or infinitive form whereas the second light verb is inflected for tense and aspect similar to MVC in the language.

Forming stacked VMWEs via recursion has not been implemented in an annotated corpus. Although PARSEME Hindi Corpus edition 1.3 does capture some of the stacked VMWEs as illustrated in Figure 1, it has not been discussed explicitly.

Figure 1: An example of LVC and MVC Stacked VMWEs in PARSEME Hindi Corpus edition 1.3. The noun /'do' and a light verb /'take'.

Further, recursivity in VMWEs can be seen at various levels thus resulting in layers of predication. In our example of LVC.cause in (4), the causative is stacked with an LVC forming an LVC.cause which can be further predicted with an MVC. The stacked VMWE in (10) thus shows stacking of three VMWEs – LVC+causative+MVC.

(10) lârke-ne naukar-se boy.3.SG-ERG servant.3.SG-INST gehnô-ki cori jewellery.PL.M-GEN.F theft.F kar-va dali do-IAUS.SG.M put.PST.F

‘The boy had the servant steal away the jewellery.’

The annotation of these layers of predication is shown in Figure (2).

Figure 2: An example of LVC, Causative, and MVC Stacked VMWEs in PARSEME Hindi Corpus edition 1.3. The noun /'sight' combines with the verb /'do', indirect causative marker /va/, and a light verb /'take'.

While VMWEs are formed via recursivity of existing multiword expressions, we do not intend to annotate them with a new label. Rather, we extract them using existing annotations which will be more efficient (see Section 3.2.3).

3. Enhancing the Annotations

The task of identifying multiword expressions is challenging and requires linguistic expertise. While the annotation guidelines developed as part of PARSEME shared task (Ramisch et al., 2020, 2018) standardizes the process of identification of VMWEs for many languages but there still exist various problems. In the following sections, we discuss some of the issues found in the PARSEME Hindi corpus edition 1.3 pertaining to existing annotation of VMWEs in Hindi and their refinement. We also discuss the annotations of morphological feature for causatives (Section 3.1) and representation of Stacked VMWEs (Section 3.2.3) in the existing annotation schema.

The PARSEME corpus of Hindi uses a treebank which is annotated using UD framework and therefore we could employ annotations for morphological description of tokens for automatic tagging of VMWEs.

3.1. Semi-Automated Annotation of morphological VMWEs

Beginning with causatives, we propose to add them as a morphological feature. If a verb is present in its causative form then we add ‘Cause=Yes’ as a boolean feature as illustrated in Figure (3). We note that Universal Dependencies guidelines have a similar feature ‘Voice=Cau’⁵. In a future version of our corpus, we plan to update this feature to be in accordance with UD guidelines.

(a) kârnâ- kârnana VERB VMINumber=Sing[VerbForm=Inf]Cause=Yes
(b) kârnâ- kârnana VERB VMINumber=Sing[Person=3]Cause=Yes

Figure 3: Feature structure for Hindi causative verb inflected for agreement /kârnâ/ in (a) and /kârnâ/ in (b) with the ‘cause’ morphological feature. Note that the lemma form for both the verbs is /kârnâ/.

The annotation process of causative verbs is semi-automatic as indirect causatives and one type of direct causative can be tagged using rule-based heuristics. The lemma form for /-va/ causatives have /-va/ attached however there are some discrepancies in the data therefore we have used a list of morphological endings with /-va/ morpheme varying only in terms of agreement features on the tokens to retrieve all indirect causative verbs.

The annotation of direct causatives was also challenging. Beginning with the /-va/ causatives, the UD framework does identify these causatives in their lemma. However, there are two issues in using them. First, as noted in case of indirect causative

⁵https://universaldependencies.org/u/feat/Voice.html
there are some inconsistencies with the identification of lemmas in the data. Second, Hindi also have other verbs ending with vowel /a/ like /ja/ 'go', /la/ 'get', and so on that are not causatives. Hence using only lemma leads to over-generation of tokens and to avoid that we have used multiple heuristics and manual checks while annotating the /-a/ causatives.

The second issue was with other type of direct causatives (c.f. example 8) where causative formation affects the phonological realization of the root and we get irregular forms. Since there is no particular pattern which can be exploited to identify these kind of verbs we have annotated them manually. A total of 269 causatives have been annotated – 165 automatically and 104 manually.

3.2. Automated Annotations of lexical VMWEs

Annotation of LVCs and MVCs was done in two stages, that is, automatic annotation using python scripts followed by manual adjudication. After annotating LVCs and MVCs we have extracted Stacked VMWEs.

3.2.1. LVCs

In this work, we aim to comprehensively annotate all the occurrences of VMWEs in the corpus. While examining the PARSEME corpus we observed that despite passing tests from the PARSEME guidelines a number of MVWEs were not annotated. Though it was true for all the categories, it was especially seen in case of LVCs (see Table 2 for comparison). Therefore, we used the dependency relation to find all the instances of LVCs in the corpus. Particularly, the ‘compound’ dependency relation that already identifies these noun+verb pairs have been used as in Figure 4.

```
cori  ki
steal.F do.PST.PRF.F
```

Figure 4: Compound dependency relation as tagged in UD framework for LVCs

All the missing LVCs were added to the existing corpus according to PARSEME guidelines. In order to distinguish between LVC.full and LVC.cause we use feature ‘cause’, annotated previously. For the purpose of this work, we have limited LVC.cause to only indirect causatives and have not included direct causatives.

We have also manually adjudicated the corpus using PARSEME tests for LVCs to remove any erroneous cases that have been annotated. Since, automatic annotations were dependent on UD dependency relation, we found few instances where nouns that were not abstract have been identified to be in compound relation with a verb as shown in (11)

```
(11) dʰən  li-ya
money.M take-PST.PERF.M
‘took money’
```

In (11), /dʰən/ ‘money’ is annotated for compound relation with verb /liya/ ‘take’. These were not annotated as LVCs.

<table>
<thead>
<tr>
<th>Data</th>
<th>LVC full</th>
<th>LVC cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARSEME</td>
<td>641</td>
<td>26</td>
</tr>
<tr>
<td>New</td>
<td>743</td>
<td>40</td>
</tr>
</tbody>
</table>

Table 2: Number of LVCs in existing PARSEME corpus and the new corpus.

3.2.2. MVCs

MVCs as discussed in Section 1 are formed by the combination of verb with a light verb. However, this pattern is confusable with other types of constructions in Hindi. For instance, both modal and passive constructions are superficially similar to MVCs. Modal verbs include examples like /pa/ ‘able’, and sak/ ‘can/may’ (example 12). /pa/ is ambiguous such that the same form occurs both as a simple verb meaning ‘to get’ and as a ability modal (Bhatt et al., 2011). As a simple verb, it can form a complex predicate and occur as a preverbal but it does not occur as a light verb. The current guidelines of PARSEME includes it as a light verb, however according to our current analysis the guidelines for Hindi needs to be updated to prevent confusion with modals.

For both MVCs and modals, the main verb appears in its base form while light verbs and modals are inflected for agreement features (Butt and Ramchand, 2005), as shown in (12).

```
(12) laɾka  kitab  poeta
boy.3.SG.M book.SG.F read
pa-ya can-PST.PERF.SG.M
‘The boy could read the book.’
```

Constructions like (12) will pass the PARSEME tests for tagging MVCs, however, semantically there is a difference between light verbs and modals. Light verbs contribute sub-event information as seen in (13), where light verb /diya/ ‘give’ contributes permissive meaning to the event (Butt,
1995). Modals, on the other hand, place an event into possible world semantics (Butt, 2010) (example (12)).

(13) larkhe-ne naukar-ko
boy.3.SG.M-ERG servant.3.SG.M-DAT
xat paqhi ne di-ya
letter.SG.M read.INF give-PST.PERF.SG.M
‘The boy let the servant read the letter.’

Similarly, verbs in passive constructions appear by combining any main verb with an auxiliary verb /ja/ ‘go’ as shown in (14). The /ja/ ‘go’ can participate in a number of constructions. It can be used as a simple verb with the meaning ‘to go’, as a light verb with the meaning ‘with force’ and also as an auxiliary when a sentence is passivized. On the surface, the passive resembles MVCs where two verbs are predicated and are incorrectly annotated as MVCs in the current PARSEME corpus of Hindi at several places.

(14) larkhe-se kitab
boy.3.SG.M-INST book.SG.F
pahi ga-yi
read.PST.SG.F go-PST.PERF.SG.F
‘The book was read by the boy.’

The main verb in passives, for example pahi ‘read’, in (14), is inflected for tense and aspect which violates the first test of PARSEME guidelines for MVCs that the first verb (V-dep) should be non-finite. Therefore, passives clearly are not a case of VMWEs in Hindi.

Annotating MVCs was a little challenging as there is no direct relation in UD framework that can identify these verb+verb constructions. Further, we have to avoid constructions like modals and passives to be falsely tagged. Therefore, we have applied a number of rules to identify MVCs.

We have first filtered verbs that were tagged as ‘VM’ (main verb) for their xpos and are followed by auxiliary verbs (tagged as VAUX). Since, VAUX in all of these annotations includes any verb that has not been annotated as the main verb of the sentence, we decided to use a list of commonly used auxiliaries in Hindi including copulas, progressive marker, modals, and /vala/ to filter any false positive MVC cases, thereby also resolving the issue of modal constructions being tagged as MVCs. We have also filtered main verbs for any tense, aspect, and agreement inflections resulting in verbs that are in their base or infinitive form to avoid tagging of passives.

MVCs have also been added to the existing annotations according to the guidelines. If it already exists then we do not make any changes. It was followed by manual adjudication of the data to remove any false positive cases.

On comparing with original numbers (c.f Table 1), the total number of MVCs has dropped to 269. The reason is the removal of modals and passives from the data.

3.2.3. Stacked VMWEs

In Section 2.3 we have mentioned that we are not introducing any new label for Stacked VMWEs. As discussed, Stacked VMWEs shows recursive use of different types of multiword expressions occurring as a single predicate. Therefore, they can be easily retrieved using existing annotations for LVCs, MVCs, and causatives. For instance, as illustrated in Figure 1, we can extract by looking for verbs that are annotated for both LVCs and MVCs. Table 3 shows the frequency of stacked VMWEs. Also, note that since PARSEME has not reported the numbers for Stacked VMWEs in their previous editions of the language we have kept it as null.

<table>
<thead>
<tr>
<th>Data</th>
<th>LVC.full</th>
<th>LVC.cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARSEME</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td>New</td>
<td>61</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3: Number of Stacked VMWEs in the existing PARSEME corpus as compared to the New corpus.

The above table also highlights the fact that stacking of one VMWE onto another increases the complexity of the predicates and therefore occurs less frequently when compared to other VMWEs. As we can see that there was only one instance of LVC+causative+MVC kind of expression.

3.3. Verbal Idioms

Multiword Expressions are known for their non-compositionality with VIDs being the most diverse category such that detection of VIDs by automatic means was challenging. There were two types of issues. First, when a VID was tagged with a different VMWE category. Second, when an expression from another VMWE category was annotated as VID. Therefore, we have annotated them manually using PARSEME guidelines (Ramisch et al., 2020). These led to changes in the overall numbers of VIDs. As we can see in Table 4 the numbers have increased after the reannotation of the data especially after identifying the misclassified VIDs.

4. Results and Conclusion

The main aim of this study was to enhance the existing PARSEME Hindi corpus by expanding its scope
Table 4: Number of VIDs in the existing PARSEME corpus as compared to the New corpus.

<table>
<thead>
<tr>
<th>Data</th>
<th>VID</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARSEME</td>
<td>61</td>
</tr>
<tr>
<td>New</td>
<td>74</td>
</tr>
</tbody>
</table>

Further, the results show that Hindi frequently employs VMWEs as shown in Figure 5. LVC.full are more common where as stacked VMWEs are rarer.

Both Stacked VMWEs as well as causatives are infrequent as compared to other VMWE categories in all types of Hindi corpora. Our survey of corpora from other genres e.g., the Hindi TimeBank (Goel et al., 2020) and the IIT Delhi Dialogue Corpus for Hindi (Pareek et al., 2023) shows that Stacked VMWEs and causatives are consistently used (although they are relatively infrequent). We believe it is important to include these categories in the annotation framework to have a complete picture of VMWEs in Hindi.

Another goal of this study was to refine the existing annotations. For this, we have conducted a survey and identified a number of issues in the corpus. We have added annotations for the missing cases across different categories of VMWEs and removing any erroneous cases. The refinement process involved a combination of an automatic and manual annotation followed by adjudication. In case of automatic annotations we have described a method using UD framework to annotate some of the categories.

5. References


Towards the semantic annotation of SR-ELEXIS corpus: Insights into Multiword Expressions and Named Entities

Cvetana Krstev©, Ranka Stanković©, Aleksandra Marković©, Teodora Mihajlov
Association for Language Resources and Technologies, Univ. of Belgrade, F. of Mining and Geology, Institute for the Serbian Language SASA, Association for Language Resources and Technologies

cvetana@jerteh.rs, ranka@rgf.bg.ac.rs, malexa39@gmail.com, teodoramihajlov@gmail.com

Abstract

This paper presents the work in progress on ELEXIS-sn corpus, the Serbian addition to the ELEXIS multilingual annotated corpus (Martelli et al., 2023), comprising semantic annotations and word sense repositories. The ELEXIS corpus has parallel annotations in ten European languages, serving as a cross-lingual benchmark for evaluating low and medium-resource European languages. The focus in this paper is on multiword expressions (MWEs) and named entities (NEs), their recognition in the ELEXIS-sn sentence set, and comparison with annotations in other languages. The first steps in building the Serbian sense inventory are discussed, and some results concerning MWEs and NEs are analysed. Once completed, the ELEXIS-sn corpus will be the first sense annotated corpus using the Serbian WordNet (SpWN). Finally, ideas to represent MWE lexicon entries as Linguistic Linked-Open Data (LLOD) and connect them with occurrences in the corpus are presented.

Keywords: multiword expression, named entity, word sense disambiguation, sense repository, LLOD

1. Introduction

Even in the current era of neural language models, there is a high demand for high-quality, openly accessible corpora that are annotated with senses, especially for training and evaluating semantically related NLP tasks, like word sense disambiguation (WSD) and natural language understanding (NLU) (Pedersen et al., 2023b). Despite many efforts in the field over the past decades, such corpora are still scarce for many languages with limited resources, including Serbian. This scarcity is caused not only by the lack of freely available sense inventories, which are necessary for these tasks, but also by the complexity and cost of compiling annotations since it requires substantial manpower, preferably from experienced linguists or lexicographers. For Serbian, the availability of curated dictionaries for such use is limited, and not even subsets for particular corpora annotation are available.

The paper is structured as follows: In Section 2 we give an account of related work, and continue by presenting in Section 3 the ELEXIS-WSD dataset, its extension with Serbian data and its basic annotation layers prior to the semantic annotation. Section 4 discusses the annotation of MWEs and NEs in the ELEXIS-WSD as well as its Serbian extension. The building of the sense inventory for Serbian and the role of MWEs and NEs in it are presented in Section 5. Possible ideas for publishing dictionaries of MWEs as LLOD and associating its entries with corresponding occurrences in the corpus are developed in Section 6. Finally, in Section 7 we conclude and discuss open questions, potential future research, and development.

2. Related work

A semantic concordance is a textual corpus and a lexicon, combined so that every substantive word in the text is linked to its appropriate sense in the lexicon (Miller et al., 1993). The popularity of SemCor (Landes et al., 1998), one of the initial sense-annotated English corpora based on the Princeton WordNet sense inventory (Fellbaum, 1998) inspired the NLP community to build sense-annotated corpora for many languages. Exploiting parallel texts in the creation of multilingual semantically annotated resources produced the MultiSemCor Corpus (Bentivogli and Planta, 2005). Another important multilingual sense annotated corpus is the Ontonotes (Weischedel et al., 2011), that uses the WordNet for sense annotations of the English part, whereas the Chinese and Arab parts base the sense annotations on various lexical sources.

The semiautomatic approaches to sense annotation were applied to overcome the scarcity of such data sets. The OneSec are sense-annotated corpora for word sense disambiguation in multiple languages and domains (Scarlini et al., 2020) that consist of Wikipedia texts containing between 1.2 and 8.8 M sense annotations of nouns per language.

The FrameNet project (Baker et al., 1998), based on the idea of describing lexical items through semantic frames, produces semantic frames (which contain information about the se-
mantic and syntactic valence of words). Target words are mostly nouns, adjectives, and verbs. Every frame and frame element is accompanied by a set of representative sets of manually annotated corpus attestations, and for every frame, the set of relations it enters is presented. Lexical databases based on the FrameNet principles were (or are being) built for several languages. The Salsa project (Burchardt et al., 2009) produced a German lexicon based on the FrameNet semantic frames and annotated a large German newswire corpus.

The English part of Ontonotes was annotated with verbal MWEs (Kato et al., 2018). The main outcome of the COST action PARSEME were unified annotation guidelines, and a corpus of over 5.4 million words and 62 thousand annotated VMWEs in 18 languages (Savary et al., 2018). Development was continued afterward with the inclusion of more languages and the enlargement of corpora for existing languages. The current edition of PARSEME corpus\(^1\) contains 26 languages, including Serbian (Savary et al., 2023). The expansion of MWE annotations to nominal and other MWEs is the task within COST action UNIDIVE\(^2\).

Named Entity Recognition (NER) enables the identification and classification of key information in text. The most frequently annotated classes are persons, locations, and organizations, but for a deeper text understanding identification of events, roles, time, measures, etc. is also necessary. In addition to that, named entity linking (NEL), also known as disambiguation, normalization, or entity resolution, involves aligning a textual mention of a named entity to an appropriate entry in a knowledge base, assigning a unique identity to mentioned entities.

### 3. The extension of ELEXIS-WSD

ELEXIS-WSD is a parallel sense-annotated corpus in which content words (nouns, adjectives, verbs, and adverbs) have been assigned senses for 10 languages: Bulgarian (bg), Danish (da), English (en), Spanish (es), Estonian (et), Hungarian (hu), Italian (it), Dutch (nl), Portuguese (pt), and Slovenian (sl).\(^3\) The list of sense inventories is based on WordNet for da (Pedersen et al., 2023a), en, it, nl. Wiktionary is used for es, and national digital dictionaries are used for bg, et, hu, pt, and sl (Federico et al., 2021).

In order to join this task and obtain the Serbian corpus as a part of the future edition of the sense repository being developed within WG2.T2 of the UniDive, the set of sentences from WikiMatrix\(^4\) in en was translated automatically (Google translation) into sr. We opted for the automatic translation in order to fasten the process, with the full awareness of the need to manually check the translation afterwards. This process was highly demanding, in terms of time and manpower, but it was an unavoidable step for getting high-quality dataset. A few (eight precisely) Serbian native speakers checked the Serbian sentence set thoroughly, in order to avoid literal or incorrect translation, and after that sentences were read carefully once again to resolve different issues: literally or incorrectly translated MWEs, unresolved references in the text (pronouns, e.g., in sr differ for gender, number, and case, and if the pronoun refers to an NP from the previous context, its reference had to be checked to choose the right morphological form); besides, it was necessary to check phonetic transcriptions of names (particularly personal ones), since in sr proper names are not written in the original form (the second reading and issue-resolving was done by two people). The process was time-consuming because of the very nature of the set—sentences are out of context, full of terms from different scientific areas, many of which are MWEs), their content is of encyclopedic sort, and often it was necessary to read the original document in English and/or some other language to understand the meaning and represent it correctly in sr.

After this process, the set was automatically tokenized, lemmatized, and POS-tagged (Stanković et al., 2020; Stanković et al., 2022). The outcomes of all these automatic procedures are being manually corrected. Results show that 2024 sentences in Serbian dataset have 25,478 word forms, content words tagged as: NOUN – 7,198 (diff. 2,413), PROPN – 1,552 (diff. 1,057), ADJ – 3,291 (diff. 1,256), VERB – 3,121 (diff. 913), ADV – 900 (diff. 287).\(^5\) Tasks that remain to be done include the annotation of MWEs and NEs (the first results are presented in the following section), the syntactic annotation, and linking with the sense repository (the first results are presented in Section 5).

### 4. MWEs and NEs in WSD

In this section, we are focusing on the annotation of MWEs and NEs in the ELEXIS-WSD and in its Serbian extension. As it will be shown, the number of MWEs and NEs annotated in 10 language sentence sets was not even, probably due to different resources used for their annotation.

---

1\(^{\text{https://gitlab.com/parseme/}}\)
2\(^{\text{https://unidive.lisn.upsaclay.fr/}}\)
3\(^{\text{https://www.clarin.si/repository/xmlui/handle/11356/1842}}\)
4\(^{\text{https://ai.meta.com/blog/wikimatrix/}}\)
5\(^{\text{This figures are not final since the annotation is presently being double-checked and harmonized with UD.}}\)
In order to compare MWEs and NEs occurring in the whole repository, MWEs and NEs in each of 10 languages were automatically translated into SR (as phrases, not word-to-word), and the number of the same translations obtained by translating MWEs from different languages was calculated. MWEs/NEs were automatically translated into Serbian in order to facilitate the comparison with MWEs/NEs retrieved in the Serbian set. Note that the translation of a MWE into sr need not be a MWE. For instance, *prime minister* (EN) → *premijer* (SR).

### 4.1. MWEs in ELEXIS-WSD

The number of annotated MWEs in the initial WSD repository is presented in Figure 1). The blue columns present the number of unique lemmas in the WSD, while the orange columns present the number of unique senses. This graphic shows that the numbers of MWEs in the WSD repository differ significantly between languages.

![Figure 1: Number of MWEs in the repository – a total of 1,710 MWEs in 10 languages](image)

Figure 2 shows that 1,412 different translations were obtained by translating a total of 1,710 MWEs. One international MWE appeared in 6 language sets, *lingua franca*. One of 14 MWEs translated from 4 languages into one sr term was *očekivano trajanje života* (SR): *life expectancy* (EN), *expectativa de vida* (PT), *pričakovana življenjska doba* (SL), *oodatav eluiga* (ET).

The automatic translation was not literal, as demonstrated by the example *srednja škola* (SR) ‘lit. middle school’ ↔ *visoka šola* (SL).

### 4.2. NEs in ELEXIS-WSD

The number of annotated NEs, without information about specific NE types, is presented in Figure 3 (blue columns present the number of unique lemmas, while orange columns present the number of unique senses).

![Figure 3: Number of NEs in the repository – a total of 606 NEs in 10 languages](image)

Named entities were not systematically annotated in all language datasets (for example, (SL) and (IT) sets have no NE annotated at all, while
some languages have many of them), resulting in 526 translations from a total of 606 NEs. The most frequent NE was Grčka, translated from four languages: Grækenland (DA), Grecia (ES), Kreeka (ET), Grécia (PT), followed by NEs translated from three languages, one of which is SAD: USA (EN), ZDA (SL), EUA (PT). Figure 4 presents the number of translations into Serbian.

![Figure 4: NEs translations into Serbian obtained by translating from 10 languages – no translation was obtained by translating from more than 4 languages](image)

4.3. Annotation of MWEs and NEs in the Serbian dataset

The pipeline for preparation and annotation of the Serbian set of 2,024 sentences is presented in Figure 5 – green color boxes and the closed locker symbol represent the finished tasks, pink color boxes and the open locker symbol designate the work in progress, mostly in the evaluation phase, while the pending tasks or tasks in their initial phase are represented by lilac boxes.

The Serbian set of 2,024 sentences was automatically annotated using four different resources and tools:

- The e-dictionary of non-verbal MWEs was used for the annotation of such MWEs. This dictionary was built on the same principles used for building the e-dictionary of simple words for Serbian. The inclusion of MWEs in this dictionary was based on several rather loose criteria: their appearance in some general, terminological or phraseological dictionary of Serbian as well as SrpWN, the frequency of their occurrence in corpora of Serbian, and the intuition of the resource author. The application of this resource to the Serbian sentence set resulted in 529 annotations (339 different) (Krstev et al., 2013). Among them were 351 (249) nominal MWEs, 133 (70) proper nouns, 44 (19) adverbial, and one adjectival.

- A system for the Named Entity Recognition (NER) based on e-dictionaries and rules annotated 2,006 occurrences of NEs (Krstev et al., 2014). Numbers of recognized NEs per class are presented in Table 1. Some multi-word named entities, particularly organization (ORG) and geopolitical names (TOP), are recognized both by dictionaries and the NER system.

- A system for the recognition of verbal MWEs based on e-dictionaries, rules, and the repertoire of VMWEs annotated in the Serbian part of the PARSEME Corpus Release 1.3 (Savary et al., 2023) annotated 230 occurrences of VMWEs (98 different), distribution by type: IRV – 174 (62), LVC.full – 35 (21), VID – 13 (10), and LVC.cause – 8 (5).

- A system for the recognition of adjectival and verbal similes is based on a set of more than 600 adjectival and more than 300 verbal similes. It can retrieve different variances of these similes, both in lexica and structure (Krstev et al., 2023). The previous research established that in literary texts an average of 2.2 adjectival similes can be expected per 10,000 words of a text; however, this system in the Serbian sentence set did not retrieve even a single one (Krstev, 2021).

The accuracy of MWE/NE recognition, recall, and precision will be determined during the next
step, when senses will be associated with simple-
and multi-word units. Previous evaluations of
used systems for the recognition of NEs and
MWEs (Krstev et al., 2013; Šandrih et al., 2019)
have shown that these systems prioritize precision
over recall, which means that in the later stages of
processing, through comparison with annotations
in datasets for other languages and manual eval-
uation, new entities will be annotated. It is to be
expected that the assignment of senses will reveal
some additional MWEs and NEs. This, in turn, will
enable the enhancement of used resources and
procedures.

<table>
<thead>
<tr>
<th>Tag</th>
<th>No</th>
<th>Tag</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERS</td>
<td>329</td>
<td>TIME</td>
<td>372</td>
</tr>
<tr>
<td>TOP</td>
<td>448</td>
<td>AMOUNT</td>
<td>169</td>
</tr>
<tr>
<td>ORG</td>
<td>126</td>
<td>MEASURE</td>
<td>62</td>
</tr>
<tr>
<td>DEMONYM</td>
<td>244</td>
<td>PERCENT</td>
<td>51</td>
</tr>
<tr>
<td>ROLE</td>
<td>175</td>
<td>MONEY</td>
<td>12</td>
</tr>
<tr>
<td>EVENT</td>
<td>18</td>
<td>Total</td>
<td>2,006</td>
</tr>
</tbody>
</table>

Table 1: Recognized NEs by classes.

4.4. The comparison of MWEs and NEs
across languages

Our initial comparison of MWEs and NEs an-
notated in the WSD repository and in the Serbian
sentence set (ELEXIS-sr) was based on their au-
tomatic translation to sr, as explained in Subsec-
tion 4. This was not ideal, since in several cases
the translation was not appropriate: e.g., the sr
highly polysemous verb dovesti was obtained as
a translation equivalent of two VMWEs from two
languages, appearing in two unrelated sentences:
dado lugar, ‘led to’ (es: 700) and tag med, ‘take
in’ (da: 148). Once the automatic translation was
checked, as actual equivalents of these VMWEs
in ELEXIS-sr appeared to be primiti (148) (‘take
in’ in ELEXIS-en), and dovesti (700) (‘led to’ in
ELEXIS-en), the translated verb itself.

On the other hand, in many cases automatic
matches were good: e.g., the sr translation
bruto domaci proizvod ‘gross domestic product’
was obtained from MWEs in four languages:
gross domestic product (en), produto interno bruto
(pt), bruto domaci proizvod (sl), sisemajandus
koguproduct (et), all occurring in the same sen-
tence – 1258. In the corresponding sentence in
ELEXIS-sr the translated term bruto domaci
proizvod was used and annotated as MWE. In all
mentioned languages these terms were also anno-
tated as MWEs (in ELEXIS-sl only its part is anno-
tated: domaci proizvod).

In other cases, the translation was good, it was
used in ELEXIS-sr, but it was not annotated in
it because it was missing in the used resources.

This was the case for prirodna selekcija, trans-
lated from natural selection (en), seleção natural
(pt), naravni izbor (sl), used in sentence 1560 in
ELEXIS-sr, but not annotated in it. This case of
missing annotations occurs in other languages as
well. E.g., equivalents for gross domestic product
are MWEs in bg, es, hu, nl, but yet are not anno-
tated. In it an acronym was used instead, and in
da a compound.

Having all this in mind, the overall results of the
comparison are as follows: out of 653 non-verbal
MWEs occurrences (384 lemmas) annotated in
ELEXIS-sr, 116 MWE lemmas occurred in at least
one language set in WSD; out of 228 VMWE oc-
currences (99 lemmas) annotated in ELEXIS-sr,
11 lemmas occurred in at least one language set;
only 93 NEs annotated in ELEXIS-sr were anno-
tated as MWE or PROPN in WSD (maybe due to
the poor lemmatization, automatic translation and
linking of proper names).

5. Sense repository

Since there is no freely available digital descrip-
tive dictionary of the Serbian language, the Ser-
bian sense repository will be based on the Ser-
bian WordNet SrpWN (Stanković et al., 2018).

ELEXIS sense repository for English, which is
also based on the Princeton WordNet (PWN), has
16,106 entries, each assigned with its internal iden-
tifier. Since the WordNet interlingual index is not
available in the ELEXIS-en sense repository, we
aligned PWN synsets with the ELEXIS-en sense
repository entries by comparing their definitions.
This process yielded 13,703 matches.

Subsequently, synsets from this subset were
aligned with the Serbian WN containing 25,322
synsets, which revealed that there were 5,997
matches. Finally, the subset missing from the list
of 13,703 synsets was compared with sentence
annotations in ELEXIS-en, which revealed that the
“urgent” first step is to fill the gap with 2,130
synsets. After the automatic translation of this list
of synonyms and their definitions from the PWN us-
ging Google API and OpenAI services, the obtained
list of Serbian candidates was expanded using sev-
eral other lexical resources compiled in previous
research. Postediting the list of synonym set can-
didates and their definitions is an ongoing activity.

The further analysis showed that from 437
MWEs annotated in ELEXIS-sr (see Subsection 4) – 339 non-verbal and 98 verbal – 171 (39%)
were found in the Serbian WordNet. Moreover,
some of them occur in 2 or more synsets. Table 2
gives the total number of senses and the number of
lemmas (literals) per MWE type. For instance,
komunikacioni sistem ‘communication system’ can
refer to a ‘(def.) system for communicating’ or to
Table 2: Annotated MWEs in ELEXIS-sr retrieved in SrpWN per type.

<table>
<thead>
<tr>
<th>Group</th>
<th>Type</th>
<th>Senses</th>
<th>Lemma</th>
</tr>
</thead>
<tbody>
<tr>
<td>MWE</td>
<td>NOUN</td>
<td>100</td>
<td>94</td>
</tr>
<tr>
<td>MWE</td>
<td>PNOUN</td>
<td>35</td>
<td>32</td>
</tr>
<tr>
<td>VMWE</td>
<td>IRV</td>
<td>80</td>
<td>42</td>
</tr>
<tr>
<td>VMWE</td>
<td>LVCfull</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>VMWE</td>
<td>VID</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

The following example gives an idea of how the lexical entry for MWE fast food can be represented in RDF along with its translations – in this case in Serbian brza hrana – and how links between senses and Wikidata entries are realized.

```
:le_fast_food
  a ontolex:LexicalEntry,
  ontolex:MultiwordExpression;
  ontolex:canonicalForm
    [ontolex:writtenRep
      "fast food"@en];
  lexinfo:partOfSpeech lexinfo:noun;
  ontolex:sense
    [ontolex:reference
      <https://www.wikidata.org/wiki/Q81799>];
  decomposed:constituent :cm_food;
  decomposed:constituent :cm_fast;
  rdf:_1 :le_fast; # lexical
  rdf:_2 :le_food. # entries
```

# component of cannonical form
:cm_food a ontolex:Component;
  decomposed:correspondsTo :le_food.

```
:le_brza_hrana a ontolex:LexicalEntry,
  ontolex:MultiwordExpression;
  ontolex:canonicalForm
    [ontolex:writtenRep
      "brza hrana"@sr];
```

# simplified naming
:tranSetEN-SR vartrans:trans
  fast_food-ensns-brza_hrana-srsns .
  :fast_food-ensns
    a ontolex:LexicalSense;
  ontolex:isSenseOf :le_brza_hrana .
  :brza_hrana-srsns
    a ontolex:LexicalSense;
  ontolex:isSenseOf :le_brza_hrana .

The OntoLex-FrAC\(^8\) vocabulary implements the lexicon-corpus interface (Barbu Mititelu et al., 2017) and DMLex\(^8\). Ontolex-lemon is widely used community standard for machine-readable lexical resources in the context of RDF, Linked Data, and Semantic Web technologies (McCrae et al., 2017). DMLex is a standard for structuring (human-oriented) dictionaries, which is published by LEXIDMA, a technical committee under OASIS, an organisation which oversees the production of open standards in the IT industry.

6. Linking MWEs and corpora

A holistic presentation of MWEs in lexicons and linking their entries with occurrences in a corpus is still an open question. We are considering the use of LLOD for interlinking MWE lexicon entries with their occurrences in corpora. Two options will be taken into account: Ontolex-lemon\(^7\) (with Lexicog module) and DMLex\(^8\). Ontolex-lemon is widely used community standard for machine-readable lexical resources in the context of RDF, Linked Data, and Semantic Web technologies (McCrae et al., 2017). DMLex is a standard for structuring (human-oriented) dictionaries, which is published by LEXIDMA, a technical committee under OASIS, an organisation which oversees the production of open standards in the IT industry.

The current draft version of the FrAC specification is found under https://github.com/ontolex/frequency-attestation-corpus-information/
2024) with corpus information to support corpus-driven lexicography and the inclusion of corpus evidence (attestations). A sentence number 823 in English: "It can be made at home or bought from fast food shops." and in Serbian "Može se napraviti kod kuće ili kupiti u prodavnicama brze hrane." illustrates this in the following example:

:le_fast_food
frac:attestation [frac:quotation "It can be made at home or bought from fast food shops."@en; frac:observedIn :EWS].

:le_brza_hrana
frac:attestation [frac:quotation "Može se napraviti kod kuće ili kupiti u prodavnicama brze hrane."@sr; frac:observedIn :EWS-ext].

The cross-lingual analysis of idiosyncratic constructions can be supported by publishing aligned and annotated corpus data as Linked Data employing community standards such as the NLP Interchange Format (NIF) (Hellmann et al., 2012) and CoNLL-RDF (Chiarcos and Fäth, 2017; Chiarcos and Glaser, 2020), a minimal NIF subset designed for compatibility with tab-separated formats used in NLP ("CoNLL"), Universal Dependencies ("CoNLL-U") and Parseme ("Parseme- TSV"). The sense repository should be probably published using Ontolex-lemon. The first ideas about leveraging Linked Data, NIF, and CONLL-U for Enhanced Annotation in Sentence Aligned Parallel Corpora are given in (Stanković et al., 2023).

7. Future Work

The development of the Serbian sentence set is a work in progress, as represented in Figure 5: translation and tokenization are done, POS tagging and lemmatization checking are in the final phase, and word sense inventory is being prepared. The syntactic annotation is still pending as well as the development of a LOD dictionary.

Our future research will give special attention to the annotation of MWEs and NEs in the ELEXIS program. On the one hand, we will coordinate our work with the other research groups, primarily groups dealing with ELEXIS, Parseme, UD and UniDive activities. On the other hand, having in mind that the meticulously prepared set ELEXIS will be used for various purposes, we plan to publish its various editions, e.g. annotating NEs using a large set of classes and sub-classes. One important research path will be the production of precise guidelines for distinguishing MWEs from NEs, as well as explicating differences in the notion of a MWE in the Serbian e-dictionaries, Parseme/UniDive and WordNet (e.g. MWEs pravi trenutak or pravi čas ‘time (a suitable moment)’ would probably not be considered a nominal MWE for Parseme/UniDive, that is, they would not pass the prescribed sequence of tests).

The future research goal is the comparative analyses of MWEs and NEs in ELEXIS multilingual set both from the linguistic and NLP point of view.

Acknowledgements

This research was supported by the Ministry of Science, the Republic of Serbia, #GRANT 451-03-65/2024-03/ 200126,#GRANT 451-03-66/2024-03/200174 and COST ACTION CA21167 - Universality, Diversity, and Idiosyncrasy in Language Technology (UniDive).

8. Bibliographical References

References


9. Language Resource References

To Leave No Stone Unturned: 
Annotating Verbal Idioms in the Parallel Meaning Bank

Rafael Ehren, Kilian Evang, Laura Kallmeyer
Heinrich Heine University Düsseldorf
Universitätsstr. 1, 40225 Düsseldorf, Germany
(rafael.ehren, kilian.evang, laura.kallmeyer)@hhu.de

Abstract

Idioms present many challenges to semantic annotation in a lexicalized framework, which leads to them being underrepresented or inadequately annotated in sembanks. In this work, we address this problem with respect to verbal idioms in the Parallel Meaning Bank (PMB), specifically in its German part, where only some idiomatic expressions have been annotated correctly. We first select candidate idiomatic expressions, then determine their idiomaticity status and whether they are decomposable or not, and then we annotate their semantics using WordNet senses and VerbNet semantic roles. Overall, inter-annotator agreement is very encouraging. A difficulty, however, is to choose the correct word sense. This is not surprising, given that English synsets are many and there is often no unique mapping from German idioms and words to them. Besides this, there are many subtle differences and interesting challenging cases. We discuss some of them in this paper.

Keywords: verbal idioms, semantic annotation

1. Introduction

Despite being one of the most discussed multiword expression (MWE) types, verbal idioms (VIDs) are surprisingly challenging to define. Actually, it seems to be easier to define them in terms of what they are not, as it is done by the PARSEME annotation guidelines (Ramisch et al., 2020)\(^1\). According to these guidelines, VIDs consist of a head verb and at least one lexicalized dependent which is neither a reflexive pronoun nor a particle. If the dependent is a verb or a noun, fine-grained tests need to be applied to discriminate the expression from multiverb expressions or light-verb constructions (LVCs). Another defining – and probably the most challenging – characteristic of an idiom is its non-compositionality, i.e. the meanings of its parts do not combine to form the meaning of the whole expression. However, since Nunberg et al. (1994), it is commonly acknowledged that there exists another dimension w.r.t. non-compositionality. We now make the distinction between decomposable and non-decomposable idioms. Both types are non-compositional, but for the former we can establish a mapping from its parts to their respective idiomatic meanings which in turn combine to form the meaning of the whole. Or, if we reverse the direction: We can decompose the idiomatic meaning and map these individual meanings to the components of the expressions.\(^2\) This, however, is not possible for non-decomposable idioms whose meanings do not allow for this kind of distribution over their parts. For illustration, consider the following two classic examples:

(1) After a long interrogation the spy spilled the beans.

(2) After a long illness, he finally kicked the bucket.

Example (1) shows an instance of the idiom spill the beans which means ‘to reveal a secret’. We consider this decomposable because the individual meanings can be mapped to the different components of the expression: ‘reveal’ to spill and ‘secret’ to beans. Such a mapping does not exist for ‘kick the bucket’ in example (2) because the idiomatic meaning ‘to die’ cannot be decomposed into individual meanings.

Because of this behavior, non-decomposable idioms are more challenging when it comes to semantic annotation (and consequently semantic parsing) than decomposable ones. For the latter, there exists a one-to-one mapping from words to concepts, but not for the former. This might be the reason why they are often ignored during semantic annotation and receive a literal treatment. Consider the following example from the English partition of the Parallel Meaning Bank (PMB):

\(^1\)https://parsemefr.lis-lab.fr/parseme-st-guidelines/1.2/?page=050_Cross-lingual_tests/030_Verbal_idioms__LB_VID_RB_

\(^2\)Nunberg et al. (1994) spoke of idiomatically combining expressions, which reflects the initial direction of the analysis (starting from its parts), but since then the terminology changed in order to favor the other direction (starting from the whole expression).
Discourse representation structure (DRS) for English PMB sentence 01/1871 *Are you pulling my leg?* (not gold).

The non-decomposable idiom *pull sb's leg* has the meaning ‘to tease sb’, but in the DRS above it is treated literally as *leg* is a discourse referent \((x_1)\) which it should not be. Thus, the DRS actually represents a leg pulling event which is not the desired analysis in this case.

The goal of this work is to improve the coverage of VIDs in the PMB, so that ultimately semantic parsers trained on its data can benefit from it. Furthermore, as a byproduct, we created a dataset of potentially idiomatic expressions (PIEs; Haagsma et al., 2020), since we also labeled instances of literal counterparts of VIDs. This will be further elaborated at the end of section 4.

The structure of the paper is as follows: First, we will discuss related work and the PMB. Then, we will detail the extraction of candidate sentences and the annotation process. Finally, we will present the results and discuss especially challenging cases before we draw our conclusions.

## 2. Related Work

Arguably the most well-known MWE corpora are the four editions (1.0–1.3) of the PARSEME corpus (Savary et al., 2015; Ramisch et al., 2018, 2020; Savary et al., 2023). What sets them apart from other corpora is their scope and homogeneity: The PARSEME corpora consist of a large number of datasets from different languages that were all annotated for verbal MWEs according to the same annotation guidelines. PARSEME corpora are not sense annotated, but these guidelines are highly relevant to us, too, as we used their definitions of the different verbal MWE types to decide which candidate expressions to annotate.

A corpus that contains semantic annotation of MWEs is the STREUSLE corpus (Schneider and Smith, 2015). It is a 55,000 words English web corpus consisting of reviews which were annotated for MWEs, but without restrictions to specific kinds of syntactic constructions. Furthermore, it distinguishes between *strong* and *weak* expressions, the former being opaque idioms (*shoot the breeze*) while the latter are more transparent collocations (*traffic light*). On top of that, they added a level of supersenses which are the top-level hypernyms in the WordNet taxonomy. There is no explicit mention of decomposable and non-decomposable idioms, but the aforementioned *strong* expressions receive a supersense as a unit while weak ones do not. So it is probable that non-decomposable expressions received the appropriate treatment w.r.t. to supersense tagging. However, since there were no guidelines to differentiate decomposable and non-decomposable idioms, it is not unlikely that some of the former were annotated as strong and thus erroneously received a holistic treatment.

Sembanks (corpora with deep meaning representations) treat idioms in different ways. Abstract Meaning Representations (AMR; Banerjee et al., 2013) and Uniform Meaning Representations (UMR; van Gysel et al., 2021) are not lexically anchored, so usually introduce a single concept node for an idiom consisting of several words (Bonn et al., 2023). On the other hand, sembanks with lexical anchoring need explicit mechanisms for dealing with cases where the word-concept mapping is not one-to-one, such as idioms. For HPSG, such mechanisms have been proposed, e.g. by Richter and Sailer (2014), but not, to our knowledge, applied in sembanks such as LinGO Redwoods (Oepen et al., 2002).

## 3. The PMB

The Parallel Meaning Bank (PMB; Abzianidze et al., 2017, 2020) is a partially parallel corpus of text in English, German, Italian, and Dutch, with semantic annotations. These include WordNet senses (Fellbaum, 1998) and VerbNet semantic roles (Kipper Schuler, 2005), among others. All semantic annotation layers are integrated into a meaning representation language based on Discourse Representation Theory (Kamp and Reyle, 1993) which places more emphasis than other frameworks such as AMR on precisely representing the scope of quantifiers as well as modal and logical operators. The semantic representations in this formalism are called Discourse Representation Structures (DRS).

The PMB is built using a dynamic annotation methodology (Oepen et al., 2002) based on a strongly lexicalized theory of the syntax-semantics interface. Statistical models produce an initial syntactic analysis of each sentence using Combinatory Categorial Grammar (CCG; Steedman, 2001) as well as an assignment of semantic tags, roles, senses, etc. to tokens. These annotation layers are corrected by human annotators by adding constraints called *bits of wisdom*. Bits of wisdom are stored in a database so they can be automatically reapplied to the output of the new versions of the statistical models in the future. The result is then fed into a rule-based component named Boxer which assigns a partial meaning representation (\(\lambda\)-DRS) to each token and then computes a DRS for the entire sentence. Automatically pre-
annotated documents are said to have ‘bronze’ status, documents with at least one bit of wisdom are ‘silver’, and documents marked as completely corrected by a human are ‘gold’.

While the syntax-based annotation methodology of the PMB helps ensure consistency, it is challenged by multiword expressions where the mapping between lexical meanings and tokens is not one-to-one. Some types of verbal multiword expressions are already handled adequately. For example, in the verb-particle construction (4) and in inherently reflexive verbs (5), the meaning is assigned to the head, and the other element is treated as semantically empty. Decomposable verbal idioms as in (6) are treated by assigning each component a suitable non-literal meaning. Of course, this is only true for documents that have already been annotated by humans; the automatic pre-annotation usually fails to pick correct non-literal senses, as shown for a German idiom in (7). Furthermore, not much attention has so far been given to light verb constructions and non-decomposable idioms. As a result, most sentences containing such constructions do not have a gold annotation in the PMB yet, but only an automatically generated (i.e., bronze status) and semantically inadequate annotation using a literal sense of each word. Examples of this are shown in (8) and (3).

The first step was to find potential candidates for the annotation, i.e. sentences that contained German VID instances. To this end, we collected VID types from the Redensarten-Index\(^3\) (transl. Proverb-Index), an electronic, privately maintained dictionary, which, contrary to the name, not only contains German proverbs but also an even larger number of idioms. At the time of this writing, the database comprises 15,661 entries. Since a lot of entries consist of several variants of the same expression, this number rises to 54,936 when counting every variant as a different type. After filtering out all the non-verbal expressions using parsing, 39,521 verbal ones remained.

After compiling a list of VID types, the next step was to find sentences in the PMB that contained instances of those VID types. We employed the parsing-based extraction method described in Haagsma (2020). This method only extracts sentences that contain the lemmata in the same dependency relations as the VID type, thus the focus of this approach is to increase precision by not extracting sentences that coincidentally comprise the same lemmata. Figure 1 shows two sentences that contain the tokens kicked, the and bucket, but only in (a) they have the desired dependency relations: ﬁsh between bucket and kick and obj between kick and bucket. In (b), the relation that holds between kick and bucket is oblique (for oblique) and accordingly the sentence would not be extracted, since it does not contain an instance of kick the bucket but only an accidental co-occurrence.

We employed UDPipe 2.12\(^4\) (Straka, 2018) to

In this work, we aim to improve the coverage of idioms in the PMB. This requires creating annotation guidelines that capture the semantics of such cases adequately while still fitting in with the lexicalized annotation framework of the PMB. It furthermore requires looking for idiom instances in the PMB and targeting them for annotation.

### 4. Extraction

The first step was to find potential candidates for the annotation, i.e. sentences that contained German VID instances. To this end, we collected VID types from the Redensarten-Index\(^3\) (transl. Proverb-Index), an electronic, privately maintained dictionary, which, contrary to the name, not only contains German proverbs but also an even larger number of idioms. At the time of this writing, the database comprises 15,661 entries. Since a lot of entries consist of several variants of the same expression, this number rises to 54,936 when counting every variant as a different type. After filtering out all the non-verbal expressions using parsing, 39,521 verbal ones remained.

After compiling a list of VID types, the next step was to find sentences in the PMB that contained instances of those VID types. We employed the parsing-based extraction method described in Haagsma (2020). This method only extracts sentences that contain the lemmata in the same dependency relations as the VID type, thus the focus of this approach is to increase precision by not extracting sentences that coincidentally comprise the same lemmata. Figure 1 shows two sentences that contain the tokens kicked, the and bucket, but only in (a) they have the desired dependency relations: ﬁsh between bucket and kick and obj between kick and bucket. In (b), the relation that holds between kick and bucket is oblique (for oblique) and accordingly the sentence would not be extracted, since it does not contain an instance of kick the bucket but only an accidental co-occurrence.

We employed UDPipe 2.12\(^4\) (Straka, 2018) to

\(^3\)https://www.redensarten-index.de/suche.php

\(^4\)More specifically, the German model german-gsd-

ud-2.12-230717

---

117
parse the gold, silver and bronze sentences of the German part of the PMB and subsequently used the method described above to extract sentences with VID candidates. This resulted in 6,187 sentences being extracted which were then prepared for annotation.

During this process not only instances of VID types were extracted, but also instances of their literal counterparts:

(9) Beth wurde von ihrem faulen Freund gefragt, ob sie **seine Hausaufgaben** für Geschichte machen würde.

Beth was asked by her lazy friend, whether she would do her homework for history.

In (9) we have an instance of **seine Hausaufgaben machen** (to do one's homework), but since it is the literal reading of this expression, we do not have an instance of the VID type (which means 'to prepare oneself'). These kind of literal instances are not relevant to the annotation of the PMB, but we decided to label them anyway in order to create a dataset of potentially idiomatic expressions (PIEs) as a byproduct. The term PIE encompasses both the literal and idiomatic meaning of an expression, thus we will use it from here on out when we talk about both at the same time.

### 5. Annotation

The annotation was conducted by three linguistically trained native speakers, with every sentence being annotated twice. Annotators were given text files where each instance to annotate came with a “form” with several questions they had to work through step by step (cf. Fig. 2).

In a first step, the guidelines were written and subsequently revised after a trial annotation of 50 sentences. However, due to the complex nature of the task, the guidelines kept on being revised multiple times throughout the whole process. To ensure consistency there was a subsequent correction step where every annotator revised their work once again. Weekly meetings with annotators were conducted throughout to discuss difficult cases and clarify the annotation guidelines.

The annotation consisted of several objectives:

1. Filter out false positives
2. Annotate the degree of idiomaticity
3. Judging the (non-)decomposability
4. Sense and role annotation

We will discuss these steps in more detail in the following.

Firstly, due to errors during the extraction and the fact that we did not filter the list of idiomatic expressions other than for verbal types, there was a large number of false positives, i.e. types of expressions not of interest to us. Our focus was exclusively on what can be considered verbal idioms (VIDs) or, in rarer instances, light-verb constructions according to the PARSEME annotation guidelines 1.2, so verb senses that are only considered “multiword” because they obligatorily occur with a certain function word were to be ignored. These include verb-particle constructions (VPCs, e.g. **jmdm. etwas antun** ‘do something to somebody’), and inherently adpositional verbs (IAVs, e.g. **zu jmdm. halten** ‘stand by sb.’). As we have seen in Section 3, VPCs are already handled satisfactorily in the PMB, and likewise IAVs, where the adposition is treated as part of the argument and does not contribute a sense on its own. Furthermore, proverbs were also not considered as these do not have free argument slots, contrary to idioms (e.g. *A watched pot never boils*).

In the next step, the annotators had to decide whether the PIE instance fell into one of the following categories: IDIOMATIC, PROBABLY IDIOMATIC, PROBABLY LITERAL, LITERAL OR BOTH. We gave the annotators the possibility to express uncertainty with the qualifier *probably* in order to account for the fact that some sentences did not have enough context to allow for maximum certainty regarding the reading - even if the annotator happened to be rather sure.

Manual filtering a list of 39,521 expressions would have been too time consuming.

For example, because a certain PIE type was known to have one predominant reading.

---

5Because they usually can be treated compositionally.

6Manually filtering a list of 39,521 expressions would have been too time consuming.

7For example, because a certain PIE type was known to have one predominant reading.
cases in which both readings (idiomatic and literal) are active at the same time.

After that, the goal was to judge the level of decomposability of the expression. Besides the obvious labels, decomposable and non-decomposable, the annotators could also choose the labels LVC, copula and mixed. The latter three categories will be discussed in the next section in greater detail.

Strictly speaking, the previous step was not really necessary, but served as a kind of priming for the last step: the semantic annotation of the idiom and its arguments. During this step, the annotators were supposed to choose the WordNet sense (Fellbaum, 1998) that most closely corresponded to the meaning of the idiom and add it to the sentence. In order to do this, the annotators had to decide on the level of decomposability anyway because the number of senses added to the VID depended on this. Consider the next two examples for illustration:

(10) Er_[Experiencer] schwimmt_[buck.v.02] gegen_[] den_[] Strom_[Stimulus]_[trend.n.01].
    He swims against the tide.
    ‘He bucks the trend.’

(11) Stecke_[despair.v.01]_[Experiencer] niet den Kopf_[] in den Sand_[]!
    Bury the head in the sand!
    ‘Don’t despair!’

Example (10) shows an instance of the VID gegen den Strom schwimmen (swim against the tide ⇒ ‘buck the trend’), which is decomposable as we can map the individual idiomatic meanings to the components: ‘buck’ → swim and ‘trend’ → tide. Consequently, the two WordNet senses buck.v.02 and trend.n.01 were added. The example furthermore shows that in addition to the senses we also added the semantic roles of the predicate’s arguments, in this case Experiencer and Stimulus. Annotators were instructed to use WordNet Search 3.18 for finding senses, and VerbAtlas (Di Fabio et al., 2019) for mapping them to VerbNet-style rolesets, but to prefer PMB-specific conventions when in doubt. As can be seen, the senses were added by suffixing an underscore followed by brackets to a component. If a component was annotated with a sense and a semantic role, the latter always preceded the former (first Stimulus then trend.n.01 in this case).

In example (11), on the other hand, we have an instance of the non-decomposable VID den Kopf in den Sand stecken (to put the head in the sand ⇒ ‘to despair’). It is non-decomposable as it is not possible to decompose the overall idiomatic meaning into individual meanings. For non-decomposable VIDs the WordNet sense (despair.v.01 in this case) was added to the verbal head of the expression, while the other brackets were left empty.

Apart from VIDs we also annotated for LVCs as they are also not handled in the desired manner in the PMB:

(12) Die Generation_[Theme] der Zeitzeugen geht_[end.v.01] zu_[] Ende_[] [...]
    The generation of contemporary witnesses goes to end [...]
    ‘The Generation of contemporary witnesses is ending.’

Example (12) contains an instance of the LVC zu Ende gehen (to go to end ⇒ ‘to end’). We consider this a special case of non-decomposability since no part of the meaning could ever be mapped to the semantically bleached verbal part. To ensure consistency we nevertheless add the sense

8http://wordnetweb.princeton.edu/perl/webwn
(end.v.01) to the verbal part of the expression. Please note that we did not annotate for expressions that according to the PARSEME annotation guidelines would be considered LVC.cause, i.e. the verb indicates the cause of the event (e.g. to grant rights or to provoke a reaction).

6. Annotation Results and Discussion

6.1. Inter-annotator agreement

For computing agreement, we excluded 341 sentences that had been discussed in annotation meetings, thus had not been annotated by two annotators independently. For simplicity, we also excluded 18 sentences that for various reasons did not have exactly 2 annotations and 7 sentences where one or both annotators detected more than one instance of the same idiom.

On the remaining 5,821 sentences, we classified annotators’ decisions both broadly into “idiom” or “not an idiom”, and more finely by, e.g. decomposability class or false positive class. On the coarse-grained comparison, annotators agreed in 3,448 cases that something is not an idiom and should thus not receive a detailed semantic annotation. In 1,945 cases they agreed it is an idiom. And in 428 cases they disagreed on this. Coarse-grained agreement is strong (Cohen’s $\kappa = .8433$).

On the fine-grained comparison, annotators agreed in 4,230 cases and disagreed in 1,591 cases, yielding a moderate $\kappa = .6311$. Table 1 shows how frequent each class is, looking only at instances where annotators agree. We can see that most instances extracted are false positives, in particular cases where the extracted structure is not an instance of the idiom type, as in Figure 1b. Among the instances unanimously classified as idioms, a large majority is annotated as non-decomposable.

Table 2 shows the ten most frequently disagreed upon classes. In many cases, annotators agree that the items are not relevant to our annotation goal, they just disagree on why (e.g., IAV vs. not an instance). In other cases, annotators came to different conclusions regarding decomposability. Finally, there are cases where one annotator annotated the item as a non-decomposable idiom whereas the other deemed it not an instance, an IAV, not a verbal PIE type, or literal.

For the sense and role annotation of items that both annotators classified as an idiom, we look at whether both annotators selected the same word as the syntactic head of the idiom (head selection), whether they assigned the selected head the same sense (head sense classification), and for each word in the sentence whether they marked it as the head of an argument that is part of the (decomposable) idiom (internal argument identification), or as an argument that is not part of the idiom (external argument identification). For unanimously identified internal arguments, we also look at role and sense classification, and for unanimously identified external arguments, at role classification. Table 3 shows the results, with strong agreement for head selection and argument identification, weak to moderate agreement for head sense classification, and moderate to strong agreement for argument role and sense classification scores.

6.2. Challenges to the annotation

In the following we will discuss some of the reasons that made the task quite challenging. As mentioned above, the guidelines were revised multiple times during the annotation process.

Decomposability One of these revisions consisted of adding another category w.r.t. decomposability. During the annotation it became clear that some expressions do not fit the binary distinction of decomposability presented above:

<table>
<thead>
<tr>
<th></th>
<th>frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>not a verbal PIE type</td>
<td>90</td>
</tr>
<tr>
<td>literal</td>
<td>121</td>
</tr>
<tr>
<td>decomposable, non-decomposable</td>
<td>181</td>
</tr>
<tr>
<td>non-decomposable, not an instance</td>
<td>136</td>
</tr>
<tr>
<td>LVC, non-decomposable</td>
<td>108</td>
</tr>
<tr>
<td>not a verbal PIE type, not an instance</td>
<td>91</td>
</tr>
<tr>
<td>IAV, non-decomposable</td>
<td>73</td>
</tr>
<tr>
<td>non-decomposable, not a verbal PIE type</td>
<td>43</td>
</tr>
<tr>
<td>IAV, literal</td>
<td>41</td>
</tr>
<tr>
<td>literal, non-decomposable</td>
<td>39</td>
</tr>
</tbody>
</table>

Table 2: Most frequent disagreements in PIE classification. Entries in bold are not only fine-grained but also coarse-grained disagreement.
whether to prioritize decomposition even when non-decomposable analysis would have been more convenient because a very suitable sense was available:

Another frequently discussed question was whether to prioritize decomposition even when non-decomposable analysis would have been more convenient because a very suitable sense was available:

Example (14) contains an instance of the VID jmdn. auf freien Fuß setzen (to set sb. on free foot ⇒ ‘to set sb. free’), so the WordNet sense set_free.v.01 would have been very fitting, but since we decided to prioritize the decomposition of the expression in such cases we opted for a decomposable analysis which seems less elegant.

Missing senses As one can imagine, it is not always straightforward to map a German idiom to an English WordNet sense. Sometimes there are two or more equally plausible possibilities, leading to spurious disagreement, e.g. dazzle.v.02 or stagger.v.04 for jmdm. den Atem rauben ‘to take sb.’s breath away’. In case of missing verbal synsets, we were often able to use a nominal, adjectival, or adverbal one instead, as in (15).

But sometimes we were hardly able to find any fitting sense at all.

For example, the expression nichts zu verlieren haben ‘to have nothing to lose’ means something along the lines of being desperate and prone to dangerous behavior, but we were not able to find a synset capturing this, as, e.g. desperate.a.03 seemed both too general and too specific, so we did not annotate (16), although in cases were we found a synset that was a bit too general but not too specific we usually accepted it, as in (17).

Some idioms have an emphatic meaning component not captured by the synset we assigned it, as in (18).

As a last resort when unable to find a roughly fitting synset, we would create a new one:

The expression jmdm. fällt die Decke auf den Kopf (the ceiling falls on sb’s head) alludes to the negative psychological effects someone can experience when confined to a small space for a long period of time. In English, the term cabin fever ex-
ists to describe this state, but it is not available in WordNet. And neither is any equivalent sense, so in such cases, we made a sense up which we suffixed with 00 (cabin_fever:n.00 in (19)).

**Collocations** Lastly, the status of collocations was discussed frequently. Although we were not aware of it during annotation, we find the distinction between *idioms of encoding* and *idioms of decoding* (Fillmore et al., 1988; Richter and Sailer, 2014) helpful. Idioms of decoding are idioms proper: a listener has to know the expression to understand it, e.g. *ins Gras beißen*, lit. ‘bite into the grass’, ‘kick the bucket’. Idioms of encoding require the speaker to know an expression to encode the meaning idiomatically, e.g. to know to say *Zähne putzen*, lit. ‘clean teeth’, ‘brush teeth’, and not *Zähne sauber machen*, lit. ‘make teeth clean’, although both encode the meaning compositionally and are understandable without having the expression in the mental lexicon. Mere idioms of encoding are sometimes called collocations, and were out of scope for this annotation project. But sometimes the difference is hard to tell.

(20) Endlich zeigte er sein wahres Gesicht.
Finally shows he his true face.
‘Finally he reveals his real personality.’

(21) Wir sollten das wohl unter vier Augen besprechen.
We should that probably among four eyes talk about.
‘We should probably discuss this in private.’

For example, in (20), one can argue that *sein wahres Gesicht zeigen* is an idiom of decoding because *Gesicht* with the sense *personality* is not often, perhaps never found outside of this expression, whereas *zeigen* with the sense *reveal* is quite common. Another example is shown in (21), where one can likewise argue that the adverbial phrase *unter vier Augen* in the sense *in private* usually only occurs with the verb *besprechen* or a small set of near-synonyms like *bereden, diskutieren*. We did not annotate these examples in the end and leave defining a sharper criterion for distinguishing idioms from collocations for future work.

7. **Conclusions and Future Work**

Idioms present many challenges to semantic annotation in a lexicalized framework, which leads to them being underrepresented or inadequately annotated in sembanks. In this work, we have carried out a targeted annotation of German idioms in the Parallel Meaning Bank by automatically detecting instances of potentially idiomatic expressions (PIEs) and annotating them for their idiomatic status, as well as their semantics, including WordNet senses and VerbNet semantic roles. Many automatically detected PIEs were false positives; of the rest, most received non-decomposable analyses, some decomposable ones, and some received special labels like MIXED, COPULA, or LVC. Inter-annotator agreement across the subtasks is very encouraging considering the complexity of the task, with the lowest score achieved for word sense disambiguation, unsurprising given that English synsets are many and there is often no unique mapping from German idioms and words to them. As our qualitative analysis of the results shows, there are also many subtle difficulties in classifying PIEs.

The next challenge will be to actually integrate the produced annotations into the PMB so as to get closer to a gold standard semantic annotation for sentences containing idioms. We are preparing a translation of the annotations into *bits of wisdom*, the format in which human annotator decisions are stored in the PMB and then inserted into the PMB’s dynamic annotation workflow. Assigning senses and roles is relatively straightforward; however, for non-decomposable idioms, we also have to make sure that the arguments get assigned λ-DRSs that do not contribute concepts, which will require adding some new rules to Boxer, the rule-based component computing meaning representations based on syntax and token-level annotations. The documents receiving the annotations will automatically receive silver status and have to be checked manually again to receive gold status. This will make the PMB a more comprehensive and challenging testbed for data-driven DRS parsers such as van Noord et al. (2020) or Shen and Evang (2022), whose ability to handle idioms future work will also address. Furthermore, an analogous annotation project is currently under way for English idioms in the PMB.

**Acknowledgments**

We would like to thank the anonymous reviewers for their feedback. We would also like to thank our annotators for their work. This work was carried out in the MWE-SemPrE project funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), project number 467699802.

8. **Bibliographical References**

Julia Bonn, Andrew Cowell, Jan Hajic, Alexis Palmer, Martha Palmer, James Pustejovsky,


9. Language Resource References


Universal Feature-based Morphological Trees

Federica Gamba, Abishek Stephen, Zdeněk Žabokrtský
Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics
Malostranské náměstí 25, Prague, Czech Republic
{gamba, stephen, zabokrtsky}@ufal.mff.cuni.cz

Abstract
The paper proposes a novel data representation inspired by Universal Dependencies (UD) syntactic trees, which are extended to capture the internal morphological structure of word forms. As a result, morphological segmentation is incorporated within the UD representation of syntactic dependencies. To derive the proposed data structure we leverage existing annotation of UD treebanks as well as available resources for segmentation, and we select 10 languages to work with in the presented case study. Additionally, statistical analysis reveals a robust correlation between morphs and sets of morphological features of words. We thus align the morphs to the observed feature inventories capturing the morphological meaning of morphs. Through the beneficial exploitation of cross-lingual correspondence of morphs, the proposed syntactic representation based on morphological segmentation proves to enhance the comparability of sentence structures across languages.

Keywords: Morphs, Universal Segmentations, Universal Dependencies

1. Introduction

Universal Dependencies (UD) (de Marneffe et al., 2021) is a framework for consistent annotation of natural language data across languages. The UD project develops cross-linguistically consistent treebanks to facilitate multilingual and cross-lingual parsing research from a typological perspective. However, the syntactic annotation proposed by UD, along with the standard tokenization often based on white-space, poses some challenges to actual comparability across languages, as different languages may adopt different strategies to express the same phenomenon. Consider, for instance, the English sentence *I will go through a forest*, translatable in Czech as *Půjdu lesem*.

![Figure 1: UD tree for the English sentence I will go through a forest.](image1.png)

![Figure 2: UD and morphological tree for the Czech sentence Půjdu lesem. Pů – a prefix expressing future tense, jd – the room morph for 'to go', u – a 1st pers. sg. conjugation ending, les – the root morph for 'forest', em – instr. sg. masc. declination ending.](image2.png)

These two equivalent sentences exhibit noticeable differences already in the token count, and their dissimilarity is reflected in their respective dependency tree structures. Nonetheless, a closer look at the sentences reveals that splitting word forms based on their morphological segmentation leads to a better mapping concerning isomorphy of trees and alignment of nodes, allowing for greater comparability. Notably, in this example, Czech encodes future tense through the prefix *pů*, whereas the ending *em* for instrumental case in *lesem* expresses movement through (Figure 1, 2). Similarly, at the surface level the German compound *Finanzkrise* ‘financial crisis’ does not correspond –

---

1. [https://universaldependencies.org/](https://universaldependencies.org/)
2. At least in the case of languages with the alphabetic writing system.
3. At the word level, we observe a 3:1, 3:1 node alignment; at the morph level, node alignment is 1:1, 1:1, 1:1, 1:1, 1:0 (article unexpressed in Czech), 1:1.
in terms of structure and token count – to its Czech counterpart finanční krize. However, if we segment the two members that led to the formation of the compound (Finanzen + Krise), we obtain a clear correspondence of the German and Czech forms. A syntactic representation based on morphological segmentation could thus enhance the cross-lingual comparability of languages that e.g. exhibit different amounts of inflection or productivity in compounding.

Additionally, what emerges from the observation of segmented morphs\(^4\) is that morphological features often tend to be associated to specific morphs. For instance, in the English word letters the morphs can be morphologically interpreted as an encoding for plurality. The morphological specification of a (syntactic) word form is encoded by a set of features in UD representing the lexical and grammatical properties. UD differentiates between lexical and inflectional features, where the former are an attribute of lemmas and the latter of word forms. This approach is convenient and productive in capturing the morphosyntactic functions of word forms, which fits the goal of UD, but it will not be incorrect to postulate that such lexical or grammatical functions can be encapsulated within morphs in a word form.

Thus, this study aims to propose a novel data representation, which exploits UD-like trees to represent simultaneously the UD-like syntactic sentence representation as well as the internal structure of word forms (hence taking the Item-and-Arrangement perspective on morphology (Bram, 2012)), which is merged within a single dependency tree. Using the inventory of universal morphological features in UD, we also investigate whether a strong correlation can be found between a given morph and a feature value, and then align the morphs to the observed feature that captures the lexical and grammatical functions of morphs. We thus propose a data structure that intertwines syntax and morphology with the goal of increasing comparability across languages.

The remainder of the paper is structured as follows. In Section 2 we present the related work, while Section 3 offers an overview of the resources that we employ for the present study. Section 4 details how such resources are exploited, focusing on the manipulation of treebank nodes and feature extraction, as well as discussing the strategy devised to comply with the UD schema. Section 5 shows the UD-like morphological trees that result from the present work, while Section 6 concludes the paper and outlines future research directions.

\(^4\)Due to the ambiguous usage of the term ‘morphemes’, we use the term ‘morphs’ henceforth based on Haspelmath (2020).

2. Related Work

The idea of representing the internal structure of words has been previously explored, especially for non-alphabetic languages such as Chinese. In these kinds of languages, the issue of delimiting word boundaries is far from trivial and requires alternative strategies to be inspected. For instance, Zhao (2009) investigates internal character dependencies inside a word as a result of the attempt to handle word boundaries by identifying character-level dependencies.

Li (2011) elaborates on this approach by suggesting to recover word structures in morphological analysis. One of the reasons for this lies in the observation that there exist many different annotation standards for Chinese word segmentation, which could even cause inconsistency in the same corpus.\(^5\) As we are working with alphabetical languages, their motivation for the work differs from ours. Additionally, we adopt dependency structures, while they work with constituency trees.

Concrete applications in the parsing of the approach in Li (2011) are described e.g. by Zhang et al. (2013), who annotate internal structures of words and then build a joint segmentation, part-of-speech (POS) tagging and phrase-structure parsing system. Zhang et al. (2014) integrate inter-word syntactic dependencies and intra-word dependencies, differentiating intra- and inter-word dependencies by the arc type to achieve results comparable to conventional resources.

In the case of languages with alphabetical writing systems, CELEX (Baayen et al., 1995) represents morphological word structure for Dutch, English, and German in the shape of a tree. Steiner (2017), e.g., exploits the resource in combination with GermaNet (Hamp and Feldweg, 1997). Morphological and compound information is extracted from the two resources respectively, and reused to build a so-called morphological treebank for German. However, such a morphological treebank consists of tree-shaped single tree-words only, without including any kind of syntactic information at a sentence level.

An example of integration of morphology and syntax is provided by the UD treebank for Beja (Kahane et al., 2021), a Cushitic language spoken in Sudan. In the treebank, a morph-based tokenization instead of a word-based one is adopted. All affixes are dependent on the stem and are assigned UD deprels corresponding to their functional role, with an additional :aff subtype (e.g., subject pronominal affixes are marked as nsubj:aff).

\(^5\)For instance, vice president could be considered as a single word or split into two words.
3. Exploited Resources

For the present study, we exploit the resources described hereafter. UniSegments, UniMorph, and SIGMORPHON data are selected to obtain the segmentation, which we employ to manipulate UD trees. The selection of the languages primarily stems from their availability across all resources.\

UniSegments UniSegments (Zabokrtský et al., 2022) is a collection of harmonized versions of selected resources relevant for segmentation, whose data have been converted to a common scheme. It comprises 17 existing data resources featuring information about segmentation in 32 languages. The level of granularity of information varies across the different resources. Some of them classify segments specifying whether they are either roots, prefixes/suffixes, inflectional endings, or zero morphemes; yet, despite using the same labels, they adopt different definitions of the classes. In the attempt to devise a truly shared schema, the creators of UniSegments chose to preserve the parts that require deep in-language expertise (e.g., lemmas), unify the information available in most resources (POS tags and, to some extent, segmentation), and keep as much of the language/resource-specific information as possible unchanged (Zabokrtský et al., 2022). This ensures a balance between the diverse levels of granularity observed in the resources but does not guarantee their full conformity. Inevitably, such discrepancies among the resources will be indirectly reflected in our data. At times, UniSegments includes more than one resource for the same language; in such cases, we select only one resource. We work with DeriNet (Vidra et al., 2021) for Czech, MorphoLex (Sánchez-Gutiérrez et al., 2018) for English, Demone (Hathout and Namer, 2014) for French, DerIVA-Tario (Talamo et al., 2016) for Italian, WordFormationLatin (Litta et al., 2016) for Latin, and MorphyNet (Batsuren et al., 2021) for Catalan, Finnish, German, Hungarian, and Portuguese.

UniMorph The Universal Morphology (UniMorph) (McCarthy et al., 2020) project aims at providing instantiated normalized morphological paradigms for hundreds of diverse world languages, provided in a shared morphological schema. As far as the languages we include in our work are concerned, morphological information is extracted from Wiktionary (e.g., for Finnish) or derived from existing morphological dictionaries which are publicly hosted on the LINDAT/CLARIAH-CZ repository (for English, French, German, Italian).7 Since information about vowel length is available for Latin data in UniMorph, data normalization is needed before undertaking the manipulation of nodes in treebanks.8

SIGMORPHON Some datasets were made available for the SIGMORPHON 2022 Shared Task on Morpheme Segmentation (Batsuren et al., 2022). We choose to exploit Czech gold annotated data, as the quality of the results could prove to be positively affected.

Universal Dependencies A brief introduction to UD is available in Section 1. For the languages under study, we select the following treebanks from version 2.12 (Zeman, 2023). Whenever a Parallel Universal Dependencies (PUD) treebank (Zeman et al., 2017) is available we include it, as the PUD collection can provide interesting insights in terms of parallel, cross-lingual comparison. Additionally, we also select PDT (Hajič et al., 2020) for Czech, GUM (Zeldes, 2017) for English, TDT (Pyysalo et al., 2015) for Finnish, GSD (McDonald et al., 2013) for French and German, ISDT (Bosco et al., 2013) for Italian, and Bosque (Rademaker et al., 2017) for Portuguese. We employ Ancora (Taulé et al., 2008) for Catalan, Szeged (Vincze et al., 2010; Vincze et al., 2017) for Hungarian, and ITTB (Passarotti, 2019) for Latin, for which no PUD treebank is available.

4. Workflow

We now describe the strategy designed to process the selected data and extract from it all the exploitable information. It mainly revolves around two main tasks: on the one hand, the manipulation of nodes in treebanks based on the segmentation contained in the selected sources (Subsection 4.1); on the other hand, the process of alignment between universal features and morphs (Subsection 4.2). As a result, we release a set of treebanks where morphological segmentation is incorporated within the UD representation of syntactic dependencies.9 How the morphs are integrated into the UD annotation is discussed in Subsection 4.3.

4.1. Manipulation of Treebank Nodes

As a first step, we convert the official UD treebanks to morphologically segmented treebanks, as described hereafter and illustrated in Figure 3.

To manipulate data we exploit Udapi (Popel et al., 2017), a framework providing an application programming interface for UD data. The code that performs the transformation is not language-specific,\

---

7With the only exception of SIGMORPHON.

8For instance, â and å are normalized as a.

9Both the code and the set of treebanks are openly available at https://github.com/fjambe/feature-based-morpho-trees/.
provided that resources featuring morphological information (e.g., about segmentation, derivation, inflection) are available. It takes as input the UD treebank to manipulate and outputs a version of it where morphological trees of segmented words are blended in UD tree-shaped sentence representation, within a well-formed CoNLL-U file.

By iterating over each node, we check whether information about morphological segmentation of the node form or lemma (as further explained later) is available in any of the exploited resources, i.e. UniSegments and UniMorph mainly, as well as SIGMORPHON gold data for Czech.\(^\text{10}\)

**Step 0: SIGMORPHON data.** In the case of Czech, we exploit SIGMORPHON manually annotated data as an additional resource. As a preliminary step in the workflow, for each form we first check whether it occurs in SIGMORPHON data; if it does, we split the form according to this segmentation. Since SIGMORPHON data only provides splitting, with no additional information about the resulting morphs, deciding which morph of the word should be considered the root is not straightforward. Thus, we decide to select as root the least frequent morph among those we identify within the word. Morph frequencies were calculated initially on the whole dataset. Whenever a form is found in SIGMORPHON, we then cease looking for possible additional segmentations, since forms in SIGMORPHON data are fully segmented. If, conversely, the word form is not retrieved at this stage, we continue with the procedure valid for all languages.

**Step 1: segmented lemma.** The first step consists of looking up the word lemma in UniSegments. If a match is found and a segmentation is available for the retrieved lemma,\(^\text{11}\) the information just retrieved is now stored, to be exploited subsequently to segment the node. For instance, the Czech word prokonzul ‘proconsul’ is found in UniSegments as well as provided with a segmentation (pro + konzul).

**Step 2: (un)inflected form, segmented lemma.** Afterward, we check if the node form corresponds to its lemma, i.e. if the token is not an inflected form. If this is the case, we proceed to split the form based on the segmentation retrieved in UniSegments, as illustrated by the prokonzul example. Conversely, if the form is inflected we postpone the splitting until we have gathered more information about the word ending. For this purpose, we begin by verifying whether the form is listed in UniMorph, which comprises a catalog of inflected forms. If this proves to be the circumstance, we combine the information from UniSegments with information about inflection retrieved from UniMorph. See e.g. the Catalan plural form culturals ‘cultural’, whose lemma is split in UniSegments as cultur + al, while UniMorph provides the morph -s for plural. If, conversely, no match is found in UniMorph, we design a strategy to obtain an approximation of the inflectional ending by comparing character by character the two strings (form and lemma) and extract as ending the last shared character and extending till the end of the word form. It is the case of the English verb form shortened, split as short + en in UniSegments, and for which we extract the ending -ed.

**Step 3: inflected form, unsegmented lemma.** If the node lemma is not found in UniSegments, we inspect whether the node form occurs in UniMorph only. If it does, we extract the information from UniMorph and proceed to segment at least the inflectional ending of the word, as in the case of the French travaillait ‘s/he worked’, third person singular form of the imperfect tense of the verb travailler ‘to work’. The form is segmented in UniMorph as travailler (lemma) + al (ending).

**Step 4: uninflected form, unsegmented lemma.** In case the word is not comprised in either UniSegments or in UniMorph, i.e. if the node lemma and the node form do not represent entries of either of the two resources respectively, we do not implement any morphological splitting of the node and we proceed to the next one. That is, for instance, what happens with the Latin form caelum ‘sky’, corresponding to nominative, accusative, and vocative singular. Since for Latin nouns the nominative singular form is chosen as lemma, the form is not split in UniMorph; given that it is not segmented in UniSegments either, no morphological splitting can be performed on such form.

Practically, in the CoNLL-U file we handle morphologically segmented words as UD multi-word tokens (MWTs). Yet, such a decision may generate ambiguity, as it could be complex to distinguish original MWTs from morphological MWTs,\(^\text{12}\) especially when they occur jointly (i.e., a MWT which we split further). Therefore, we decide to signal

\(^{10}\)At this moment, we search only for a single best segmentation for each node, without handling possible ambiguities. Considering multiple segmentations may turn out to be necessary, especially in heavily ambiguous languages such as Arabic; morphological lattices (More et al., 2018) could be then useful for representing sets of alternative segmentations.

\(^{11}\)Some of the lemmas included in UniSegments are not provided with a segmentation. See, for instance, Czech words růk ‘year’ or jazyk ‘language’, for which the only segment identified is the root, spanning over the whole word.

\(^{12}\)Within the expression ‘morphological MWT’ we intend to use ‘MWT’ only in the technical sense of the UD label.
Figure 3: Flowchart of the node manipulation process (US: UniSegments, UM: UniMorph).

morphology-based split elements of MWTs through the deprel subtype :morph (see Subsection 4.3).

Since we are proposing a novel data representation, we have no gold data to rely on to assess the quality of the output of our algorithm. In light of this, we created a random sample of 20 French words and segmented them manually, which resulted in identifying 56 morphs. Of the 56 morphs in this gold data, 8 (14%) were correctly identified by UniSegments alone, 18 (32%) by UniMorph alone, and 27 (48%) by our algorithm. Even though this sample is very small, it can be argued that combining the resources using our algorithm leads to a considerable improvement in the segmentation quality.

4.2. Feature Extraction

Additionally, by exploiting the statistical measures described hereafter we investigate whether and how morphs and UD feature sets align, to assess if specific feature inventories somehow capture the morphological meaning of morphs.

Similarly to what was done for node manipulation, we exploit the information contained in segmentation resources (in this case, UniSegments only) and in UD treebanks. Specifically, if a word form occurs in the treebank under study, and its lemma is also present in UniSegments, we segment it based on the segmentation provided by UniSegments. For example, in Catalan the word estacional ‘seasonal’ is present in the UD Catalan AnCora treebank and also in UniSegments, following which it is split as estacion and al. After having obtained the segmentations of the word forms from UniSegments, the UD feature set that is originally attributed to the word form is associated to the individual morphs the word form has been split into. For instance, the Hungarian word gyerek ‘children’ in the UD Hungarian-Szeged treebank has the feature set Number=Plur|Case=Nom. Based on the segmentation data for Hungarian in UniSegments, the word form is split as gyerek + ek; we assign the original feature set to both gyerek and ek. In the following step, the feature set is split into individual features and is assigned to the morphs. As a result, we now have two instances of gyerek, one with feature Number=Plur and the other with feature Case=Nom; the same applies to ek. In this manner, for every possible feature, we create an inventory of morphs to which the feature has been associated. For each feature-morph pair we calculate the joint frequency of locating a morph given a feature and the $\Delta P$ scores (Jenkins and Ward, 1965). According to Schneider (2020), $\Delta P$ is a measure of cue validity, i.e. it measures how strongly two events are linked. $\Delta P$ can be thus used to calculate collocation strength. Since it is a unidirectional dependency measure it can be decomposed in two distinct formulae, one for the forward-directed $\Delta P$ and the other for the backward counterpart. Using $\Delta P$, we obtain the measure of the strength of correspondence between a morph and a feature, and vice versa. It is reasonable to use such a unidirectional measure because the association of a morph and a feature is asymmetric. The $\Delta P$ scores are between -1 and 1.

$$\Delta P_{\text{forward}} = P(m|f) - P(m|\neg f) \quad (1)$$

$$\Delta P_{\text{backward}} = P(f|m) - P(f|\neg m) \quad (2)$$

In equations (1) and (2), $m$ stands for morph and $f$ stands for feature. $P(m|f)$ is the conditional probability of locating a morph given a feature among the other conditional probabilities in the equations. In Table 1, we present the $\Delta P$ forward and the $\Delta P$ backward scores for the morph ing in English given applied to the same files employed for manipulation of treebank nodes.

Specifically, among the available resources for French we selected Demonette.

All the steps described in this paragraph are not
different morphological features. We find that the morph *ing* has the strongest relation with the feature VerbForm=Ger. What this indicates is the fact that the VerbForm=Ger strongly correlates to the morph *ing* as indicated by $\Delta P$ forward; the $\Delta P$ backward scores show the potential feature attributes like Tense=Pres, VerbForm=Part as well as the highest ranked feature VerbForm=Ger. Hence by comparing the $\Delta P$ forward and backward scores some signals could be extracted for morph and feature correspondences. While for a well-resourced language like English, such findings are not surprising, interesting correspondences could emerge in the case of less described languages.

<table>
<thead>
<tr>
<th>Morph</th>
<th>Feature</th>
<th>$\Delta P$ forward</th>
<th>$\Delta P$ backward</th>
</tr>
</thead>
<tbody>
<tr>
<td>ing</td>
<td>Degree=Pos</td>
<td>-0.058</td>
<td>-0.118</td>
</tr>
<tr>
<td>ing</td>
<td>Number=Sing</td>
<td>-0.090</td>
<td>-0.287</td>
</tr>
<tr>
<td>ing</td>
<td>Number=Plur</td>
<td>-0.091</td>
<td>-0.251</td>
</tr>
<tr>
<td>ing</td>
<td>Mood=Ind</td>
<td>-0.096</td>
<td>-0.144</td>
</tr>
<tr>
<td>ing</td>
<td>Person=3</td>
<td>-0.094</td>
<td>-0.127</td>
</tr>
<tr>
<td>ing</td>
<td>Tense=Pres</td>
<td>0.139</td>
<td>0.148</td>
</tr>
<tr>
<td>ing</td>
<td>VerbForm=Fin</td>
<td>-0.037</td>
<td>-0.152</td>
</tr>
<tr>
<td>ing</td>
<td>VerbForm=Part</td>
<td>0.120</td>
<td>0.135</td>
</tr>
<tr>
<td>ing</td>
<td>VerbForm=Ger</td>
<td>0.966</td>
<td>0.710</td>
</tr>
<tr>
<td>ing</td>
<td>Polarity=Neg</td>
<td>-0.004</td>
<td>-0.001</td>
</tr>
</tbody>
</table>

Table 1: Probabilities of the morph *ing* in English.

In Table 2, we observe that the morph *ung* in German has the highest $\Delta P$ scores for the feature Gender=Fem. The association with other features is due to the co-occurrence with other morphs in a word form. For example, the feature set for the German word *Kleidung* ‘clothing’ is Case=Nom|Gender=Fem|Number=Sing. The observed co-occurrences with other features can be explained by the allocation of the original features among the morphs *kleid* and *ung*. This correlation indicates that morphs potentially can be attributed to morphological features in an empirical sense, and by using such collocation measures it is possible to extract some informative signals.

<table>
<thead>
<tr>
<th>Morph</th>
<th>Feature</th>
<th>$\Delta P$ forward</th>
<th>$\Delta P$ backward</th>
</tr>
</thead>
<tbody>
<tr>
<td>ung</td>
<td>Case=Nom</td>
<td>0.129</td>
<td>0.226</td>
</tr>
<tr>
<td>ung</td>
<td>Gender=Fem</td>
<td>0.467</td>
<td>0.798</td>
</tr>
<tr>
<td>ung</td>
<td>Number=Sing</td>
<td>0.267</td>
<td>0.549</td>
</tr>
<tr>
<td>ung</td>
<td>Case=Dat</td>
<td>0.230</td>
<td>0.389</td>
</tr>
<tr>
<td>ung</td>
<td>Case=Acc</td>
<td>0.246</td>
<td>0.364</td>
</tr>
<tr>
<td>ung</td>
<td>Gender=Masc</td>
<td>-0.175</td>
<td>-0.230</td>
</tr>
<tr>
<td>ung</td>
<td>Case=Gen</td>
<td>0.152</td>
<td>0.116</td>
</tr>
</tbody>
</table>

Table 2: Probabilities of the morph *ung* in German.

In the case of Hungarian (Table 3), the morph *ek* has the strongest affinity for the feature Number=Plur. But there are other morphs too in Hungarian which are responsible for carrying the feature Number=Plur, like *ok*, *ak*, *ei* and *ai*. In the case of German too, there are multiple morphs (Table 4) that mark for the feminine gender, like *keit*, *schaft*, *enz*, and so on. Our current unsupervised approach successfully captures all the morphs attributed to a given morphological feature; we however reiterate that this finding is purely empirical given the available data resource.

<table>
<thead>
<tr>
<th>Morph</th>
<th>Feature</th>
<th>$\Delta P$ forward</th>
<th>$\Delta P$ backward</th>
</tr>
</thead>
<tbody>
<tr>
<td>ek</td>
<td>Case=Nom</td>
<td>-0.006</td>
<td>-0.146</td>
</tr>
<tr>
<td>ek</td>
<td>Number=Sing</td>
<td>-0.033</td>
<td>-0.431</td>
</tr>
<tr>
<td>ek</td>
<td>Person=3</td>
<td>0.031</td>
<td>0.427</td>
</tr>
<tr>
<td>ek</td>
<td>Definite=Ind</td>
<td>0.026</td>
<td>0.328</td>
</tr>
<tr>
<td>ek</td>
<td>PronType=Ind</td>
<td>0.064</td>
<td>0.099</td>
</tr>
<tr>
<td>ek</td>
<td>Mood=Ind</td>
<td>0.030</td>
<td>0.340</td>
</tr>
<tr>
<td>ek</td>
<td>Tense=Pres</td>
<td>0.032</td>
<td>0.344</td>
</tr>
<tr>
<td>ek</td>
<td>VerbForm=Fin</td>
<td>0.028</td>
<td>0.333</td>
</tr>
<tr>
<td>ek</td>
<td>Voice=Act</td>
<td>0.028</td>
<td>0.333</td>
</tr>
<tr>
<td>ek</td>
<td>Number=Plur</td>
<td>0.163</td>
<td>0.531</td>
</tr>
</tbody>
</table>

Table 3: Probabilities of the morph *ek* in Hungarian.

From Table 5 and Table 6, we infer that the morphs *tunk* and *ok* both encode the features Number=Plur and Person=1 in Hungarian. In the case of verbs conjugated in first person plural like *voltunk* ‘we were’ and *tanultunk* ‘we studied’ the morph *tunk* has the feature set Number=Plur|Person=1, whereas the morph *ok* has the feature Number=Plur for nouns and Number=Plur|Person=2 for verbs (as in *tanultatok* ‘you all studied’), as well as the feature Person=1 (e.g. in *tanulok* ‘I study’).

<table>
<thead>
<tr>
<th>Morph</th>
<th>Feature</th>
<th>$\Delta P$ forward</th>
<th>$\Delta P$ backward</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunk</td>
<td>1</td>
<td>0.033</td>
<td>0.892</td>
</tr>
<tr>
<td>ok</td>
<td>7</td>
<td>0.232</td>
<td>0.852</td>
</tr>
<tr>
<td>ak</td>
<td>5</td>
<td>0.165</td>
<td>0.890</td>
</tr>
<tr>
<td>ek</td>
<td>5</td>
<td>0.163</td>
<td>0.531</td>
</tr>
<tr>
<td>ai</td>
<td>1</td>
<td>0.033</td>
<td>0.972</td>
</tr>
</tbody>
</table>

Table 5: Morphs for Number=Plur in Hungarian.

We do observe that a morph in Hungarian or any other language may take on multiple grammatical functions; we only cite these selected examples to highlight how polysemous morphs can be. Based on these feature sets extracted from UD it is possible to explore all the grammatical functions handled by the morphs across languages.

Based on the $\Delta P$ scores, we find that the morphological features more strongly associated with the Latin morph *us* are Case=Nom, Gender=Masc and Number=Sing (Table 7). The other features...
Morph f(morph,feature) $\Delta P$ forward $\Delta P$ backward
tunk 1 0.143 0.994
ok 1 0.136 0.119
om 1 0.141 0.328
tam 1 0.143 -0.130

table 6: Morphants for Person=1 in Hungarian.

<table>
<thead>
<tr>
<th>Morph</th>
<th>Feature</th>
<th>$\Delta P$ forward</th>
<th>$\Delta P$ backward</th>
</tr>
</thead>
<tbody>
<tr>
<td>us</td>
<td>Case=Nom</td>
<td>0.018</td>
<td>0.349</td>
</tr>
<tr>
<td>us</td>
<td>Case=Acc</td>
<td>-0.012</td>
<td>-0.247</td>
</tr>
<tr>
<td>us</td>
<td>Case=Dat</td>
<td>-0.013</td>
<td>-0.130</td>
</tr>
<tr>
<td>us</td>
<td>Degree=Comp</td>
<td>-0.012</td>
<td>-0.044</td>
</tr>
<tr>
<td>us</td>
<td>Gender=Masc</td>
<td>0.017</td>
<td>0.369</td>
</tr>
<tr>
<td>us</td>
<td>Gender=Fem</td>
<td>-0.018</td>
<td>-0.346</td>
</tr>
<tr>
<td>us</td>
<td>Gender=Neut</td>
<td>-0.013</td>
<td>-0.282</td>
</tr>
<tr>
<td>us</td>
<td>Number=Sing</td>
<td>0.014</td>
<td>0.159</td>
</tr>
<tr>
<td>us</td>
<td>Number=Plur</td>
<td>-0.018</td>
<td>-0.349</td>
</tr>
</tbody>
</table>

table 7: Probabilities of the morph us in Latin.

attributed to the morph us are potentially due to the feature values of the lexical root morph it happens to co-occur with. The $\Delta P$ backward scores indicate the morph us has a strong correspondence with the feature Gender=Masc.\footnote{However, this correlation comes purely from the data we have in hand. Theoretically, the morph us in Latin can equally express e.g., Case=Nom, Gender=Masc, and Number=Sing. Currently, we do not have a baseline to compare our empirical findings with theoretical facts.}

Given the observations, $\Delta P$ proves to be a strong unsupervised measure that extracts features associated with morphs, which potentially indicates that morphs do carry morphological features and in any case it would be reasonable to use this information to analyze word-internal structure in more detail.

4.3. Conforming to UD

When morphologically segmenting the nodes of a treebank, a natural question that arises concerns how to annotate morphs within UD. Specifically, when creating the morphological MWT we need to assign to its elements lemma, POS, morphological features, and deprel.

In many cases when segmentation is provided, UniSegments also comprises information about morphemes; namely, a word morph is possibly associated with its corresponding morpheme. For instance, the Latin verb auerto ‘to turn away’ is split as a + uerto, with the morph a associated to the morpheme a(b), which can indeed take both forms a and ab. When available, we adopt the provided morpheme as a lemma; otherwise, we set the morph lemma to be identical to its form. We assign the POS that the node originally has (i.e., before undergoing the segmentation) to the head of MWT, which should correspond to the stem of the word. All other tokens of the MWT, i.e. morphs, receive the POS tag X. Indeed, we decide not to tag them with labels describing their position with respect to the stem (e.g., prefix, suffix) or the morphological process they convey (e.g., inflection, derivation). By assigning the X UPOS tag, we try to be as compliant as possible to UD, although without affirming that we believe morphs to have a POS.

To annotate features, we exploit the feature-based alignment presented in Subsection 4.2. Specifically, for each of the morphological segments that we identify, we search for the features that are associated with them as a result of the alignment process. If any of those features can also be found in the original feature set of the token, we assign it to the morph and remove it from the set of features of the root, as we believe it to belong to the morph instead of the root.

When assigning deprels, we handle prefixes, root(s), and suffixes in a slightly different manner. Prefixes, extracted from UniSegments, are assigned nmod:morph if they are substantives (NOUN/PROPN), advmod:morph for all other POSs. If according to UniSegments the lemma presents just a single root, it inherits the deprel that the node originally had. If more than one is found, the second (and possibly more) is annotated as conj:morph. It is the case of compounds, for which the choice of conj is justified by the fact that we want all the lexical stems to be somehow on the same level. We are aware that parataxis is not the only possible relation between words constituting a compound (cf. Svoboda and Ševčíková 2024); however, we adopt this practical solution since the type of compound structure is not annotated in the exploited resources. As of now, we intend to use conj:morph only as a way to point out the co-existence of two lexical roots. In the case of suffixes, we try to approximately distinguish verbal and nominal inflection. Segmented morphs of verbs and auxiliaries are assigned aux:morph, while case:morph applies to nouns, adjectives, determiners, pronouns, adverbs, numerals, and extremely rare instances of adpositions. Whenever we are not able to reasonably assign either of the two deprels, we opt for dep:morph. As mentioned in the previous subsection, the :morph subtype allows to distinguish and retrieve all instances of morphological segmentations.

5. MorphoTrees

Figures 4(a), 4(b), and 4(c) display the same sentence, corresponding to English There are parallels to draw here between games and our everyday lives. The sentence, extracted from PUD treebanks, is shown also in Finnish and French and provides an example of how including the internal structure of
words into UD could provide interesting remarks. Indeed, parallel data available in PUD could be observed in an even more parallel perspective after morph splitting, as in different languages some features could be realized differently, but a similar approach could help align them. In Appendix A we also display the raw CoNLL-U representation of the sentences (Figures 7, 8, 9), in order for the features and the MWT-like strategy to be visible.

In the Finnish example in Figure 4(c), the word form jokapäiväisten ‘everyday ones’ is split as jokapäiväi and sten. Jokapäiväi gets the POS tag ADJ and the deprel amod and the morph sten gets the deprel case:morph as decided. In the English example in Figure 4(a), the word form games is split as game and s where the morph s gets the deprel case:morph. The compound everyday is split and day is attached as conj:morph to every. Similar splits can be also observed in the French example in Figure 4(b). Figures 5 and 6 show the integration of segmentation within non-PUD treebanks.

Everyday clearly shows a case where the two elements of the compound are attached paratactically according to our solution, whereas every is actually dependent on day within the structure of the compound. The example can also serve to highlight how the segmentation of the exploited resources, and hence its quality and level of granularity, is inherited in our data. For instance, in the verb établir the infinitive marker ir should be segmented, while it is not. Of course, this kind of choice also strongly depends on the adopted approach to morphological segmentation, which is far from being a solved problem yet. A similar observation would probably apply to Finnish as well, where some expected segmentations may be missing.
6. Conclusion and Future Work

In the paper, we presented the proposal of a novel data structure aiming at integrating the representation of the morphological internal structure of words into Universal Dependencies. Working on 10 languages as a case study, we first devised a prototype of a methodology to manipulate UD treebanks intending to include the morphological structure of words into the canonical UD-like sentence representation. Then, we investigated the alignment between morphs and feature sets, by calculating $\Delta P$ scores that indicate the strength of the relation between a morph and a feature, and proceeded to assign relevant morphological features to morphs. Both tasks exploited already existing resources to perform segmentation. Such an approach ties the quality of our data to that of the resources we employed, for which some limitations were observed (derived e.g. from conversion from different resources).

Overall, the work we presented does not intend to suggest a reorganization of Universal Dependencies towards the inclusion of internal, morphological word structure. Our goal is to provide a resource that integrates morphology and syntax, two linguistic layers often intertwining, and that can prove beneficial in enhancing comparability of languages that express comparable meaning through different grammatical strategies. The key factor for enhancing comparability lies in the cross-lingual correspondence of morphs.

In the future, we plan to improve the described workflow and expand the collection of morphological treebanks to more languages. Additionally, the extraction of the morphological trees from the sentence representation could be explored, towards their possible integration with DeriNet (Vidra et al., 2021). Moreover, in recent developments, morphological features are used to create multilingual morphological analyzers, for instance as presented by Pawar et al. (2023). We would like to carry forward our current research in that direction too by including a larger set of languages, as well as by including phenomena that we have neglected so far, such as non-concatenative morphology. We will find ways to estimate the quality of the resulting trees.

7. Acknowledgements

This work has been using data, tools and services provided by the LINDAT/CLARIAH-CZ Research Infrastructure (https://lindat.cz), supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2023062). The study was supported by the Charles University, project GA UK No. 104924 and project GA UK No. 101924; and partially supported by SVV project number 260 698. We would like to thank three anonymous reviewers for their very insightful feedback.

$^{20}$Most notably, different degrees of inflection.
8. Bibliographical References


Arya D. McCarthy, Christo Kirov, Matteo Grella, Amrit Nidhi, Patrick Xia, Kyle Gorman, Ekaterina Vylomova, Sabrina J. Mielke, Garrett Nicolai, Miikka Silverberg, Timofey Arkhangel-skiy, Nataly Krizhanovskv, Andrew Krizhanovskv, Elena Klyachko, Alexey Sorokin, John Mansfield, Valtis Ernštretis, Yuval Pinter, Cassandra L. Jacobs, Ryan Cotterell, Mans Hulden, and David


9. Language Resource References


136
A. Example sentences in ConLL-U format

Figure 7: CoNLL-U representation of the English sentence "There are parallels to draw here between games and our everyday lives (see also 4(a), 4(c), 4(b)). All three figures in the appendix allow us to better understand how morphological features have been treated. In the CoNLL-U files shown here the ninth and tenth fields have been removed, for reasons of space, as they are not strictly relevant to what is discussed in the present work.

Figure 8: CoNLL-U representation of the Finnish sentence.

Figure 9: CoNLL-U representation of the French sentence.
Combining Grammatical and Relational Approaches. A Hybrid Method for the Identification of Candidate Collocations from Corpora

Damiano Perri∗, Irene Fioravanti†, Osvaldo Gervasi∗, Stefania Spina†
∗University of Perugia
†University for Foreigners of Perugia
damiano.perri, osvaldo.gervasi@unipg.it
irene.fioravanti, stefania.spina@unistrapg.it

Abstract

We present an evaluation of three different methods for the automatic identification of candidate collocations in corpora, part of a research project focused on the development of a learner dictionary of Italian collocations. We compare the commonly used POS-based method and the syntactic dependency-based method with a hybrid method integrating both approaches. We conduct a statistical analysis on a sample corpus of written and spoken texts of different registers. Results show that the hybrid method can correctly detect more candidate collocations against a human annotated benchmark. The scores are particularly high in adjectival modifier relations. A hybrid approach to candidate collocation identification seems to lead to an improvement in the quality of results.

Keywords: Collocation, Automatic identification, Learner dictionary

1. Introduction

Multi-word expressions (henceforth, MWEs), defined as lexical units (collocations, idioms, lexical bundles, etc.) consisting of two or more words, have been the focus of extensive research in many areas including lexicography and NLP for several decades (Evert, 2004; Paquot, 2015; Spina, 2020). The creation of lexicographical combinatorial resources, such as dictionaries of collocations, explicitly targeted to learners of second languages (L2s), has been undertaken mainly for English (McIntosh et al., 2002); (Rundell, 2010), although general dictionaries of collocations not explicitly addressed to L2 learners exist for several languages, including English (Benson et al., 1986), and Italian (Urzì, 2009; Tiberii, 2012; Lo Cascio, 2013). The use of language corpora has significantly boosted research on MWEs and their lexicographic applications. This is particularly evident in the area of lexicography dedicated to MWEs, where the identification of typical word combinations hugely benefits from the use of vast collections of texts. These corpora allow to extract frequent naturally occurring lexical patterns, with the aid of NLP and statistical techniques for the analysis of word combinations (Hanks, 2012).

Two main tasks are involved in the extraction of MWEs from corpora (Seretan, 2011): the automatic identification of candidates, often according to specific a priori criteria on their grammatical and/or syntactic patterns, and the detection of phraseologically meaningful combinations (collocations, in this case), often based on frequency and/or statistical association measures, to filter out sequences of words without phraseological relevance. In this study, our focus is on the first task of automatically identifying candidate collocations in Italian corpora. We assume that the effectiveness of the subsequent stages in creating a learner dictionary of collocation strongly depends on how accurate this candidate identification proves to be. The more an automatic system based on NLP techniques can accurately identify word combinations that are potential collocations, the more accurate the data on their frequency. As a consequence, the association measures used to filter out non-collocations, all of which are, to varying degrees, dependent on frequency, can benefit from more reliable frequency values, resulting in increased accuracy.

The present study reports on an experiment aimed at proposing a hybrid approach to this task by comparing and evaluating the two most commonly used candidate detection methods - the POS-based method and the syntactic dependency-based method - with a third one resulting from the integration of the two previous approaches. For the first two methods, we adopt the denomination from (Castagnoli et al., 2016) and refer to the POS-based as the P-based approach and the dependency-based as the S-based approach, while we refer to the third integrated method as the Hybrid approach. Current collocation extraction approaches rely on linguistic pre-processing (e.g., POS-tagging or dependency parsing) of source corpora to better identify the candidates (Seretan, 2011). Previous research has shown that the P-based and S-based approaches have some limitations. The former re-
lies on an accurate and established NLP task such as POS-tagging. However, relying on positional POS patterns, it fails to capture the syntactic relations between word pairs or the marked sentence structures where the regular constituent order is reversed. For instance, a P-based approach would not detect the verb-direct object relation between *play* and *role* in Example 1 (the example is taken from Seretan, 2011, 59).

**Example 1.** *It is true, we must combat the menace of alcoholism in young people, and this text successfully highlights the role that families, teachers, producers and retailers must play in this area.*

On the contrary, this relation would probably be detected using an S-based approach that relies on parsed data and thus can identify the verb-direct object dependency. Another advantage of this approach is that it does not limit the distance between the two words constituting the candidate collocation, unlike the P-approach. However, parsing errors are a well-known shortcoming of this approach: error rates ranging from 7.85% to 9.7% of the total candidate collocations extracted were reported to be due to parsing errors by previous studies (Wu and Zhou, 2003; Lin, 1999). Despite the recent improvement in parsing accuracy, (Qi et al., 2020; Akbik et al., 2018) the parsing approach still has limitations in selecting candidate collocations as it provides little information on how words combine with each other and fail to distinguish frequent combinations and idiomatic ones with the same syntactic structure (Castagnoli et al., 2016).

This study aims to present a hybrid approach to detecting candidate collocation from corpora for lexicographic applications on a language different from English, i.e. Italian. The hypothesis we aim to validate is that this hybrid approach performs better in the candidate identification task. From an exploratory perspective, we also intend to investigate cases in which the hybrid method works better and identify cases where further improvements might be warranted.

### 2. Related work

In this section, we briefly survey the main methods and NLP techniques used to perform the specific task of detecting, or discovering (Constant et al., 2017) candidate collocations from corpora, regardless of the measures employed to identify the proper phraseological collocations, which represents a further step in the process of assembling the set of entries required by the lexicographic application.

Early NLP works addressing this task identified candidate collocations using frequent word sequences, regardless of their syntactic structure, and relied on n-gram methods to extract them from corpora (Choueka, 1988; Smadja, 1993). Later, this search "for needles in a haystack" (Choueka, 1988) more and more employed linguistically pre-processed corpora and lemmatised and POS-tagged data. This further step was especially suitable for handling morphological and syntactic variability typical of languages with richer morphology and more accessible word order (Evert, 2004). The P-approach is the first to become established, given the widespread availability of POS-tagged corpora in many languages. Many extraction systems relying on this approach involve an a priori selection of specific types of POS combinations (e.g. verb-noun, adjective-noun, etc.). Right from the start, a drastic improvement in the detection accuracy was found when a POS filter was applied (Breidt, 1993; Daille, 1994; Krenn, 2000; Ritz, 2006). These results were primarily reported for fixed and adjacent candidates, where even a simple linguistic analysis can capture basic grammatical patterns.

In later years, it has been suggested that the detection of candidate collocations can benefit from a finer linguistic analysis of texts. Seretan’s (2011) extensive study explored and evaluated the use of syntactic dependencies, as they can also capture discontinuous and syntactically flexible candidate collocations based on syntactic relations between words, improving the quality of the results. However, many systems relying on an S-approach aimed at MWE identification after parsing, so as to benefit from the previous syntactic analysis (Constant et al., 2017) reported high parsing error rates affecting the accuracy of the detection task. The issue of parsing accuracy is identified and evaluated by several studies (e.g. Orliac and Dillinger, 2003; Lü and Zhou, 2004). Lü and Zhou (2004) identified a parsing error rate >7%. Orliac and Dillinger (2003) also evaluated the most recurrent parsing errors and found that relative constructions were responsible for nearly half of the candidate collocations missed by their system.

Given all these reported limitations, it can be argued that the existing detection methods relying on an S-based approach are promising but have not yet been fully developed, due to issues related to parsing accuracy. There is, therefore, a general call for hybrid approaches to candidate collocation detection, combining the advantages of both P-based and S-based approaches while minimising their shortcomings. As Castagnoli et al. (2016) claimed, "the two methods seem to be highly complementary rather than competing with one another". Some attempts have been made to integrate the two approaches in recent years. Simkó et al. (2017) proposed a system using both POS-tagging and dependency parsing to identify single- and multi-token verbal MWEs in texts and reported the best results on the verb-particle constructions where their sys-
tem correctly identified around 60% of constructions, but only about 40% of other types. Shi and Lee (2020) proposed a joint method that combines scores from both POS-tagging and dependency parsing to extract headless MWEs. Their results showed that tagging is more accurate than parsing for identifying flat-structure MWEs. At the same time, the joint method leads to higher accuracy, and most of the gains derive from shared results between parsers and taggers.

3. Method

To validate our hypothesis and explore the performance of different systems in automatically detecting candidate collocations in Italian corpora, we designed our experiment to mimic the "natural" processes that will be employed in the final extraction of candidates to be included in a learner dictionary of Italian collocations. For instance, we did not pre-select target words or lemmas for the experiment. Instead, we considered all the word pairs produced in a text sample.

The only pre-selection we made was the syntactic relations of the candidate collocations. We opted to focus on syntactically-bound combinations, as the task of detecting candidate collocations is targeted to a lexicographic application. In the final dictionary entries, these collocations will be presented in accordance with their syntactic patterns. The choice was to investigate the two dependencies verb + direct object (Vdobj) and adjective modifier (amod) before and after a noun (both word orders are allowed in Italian). The choice is motivated by reasons of coverage and diversification. Firstly, previous research has shown that, among the eight syntactic structures most commonly forming collocations in Italian (verb + direct object, amod, noun + preposition + noun, noun + noun, verb + adjective, verb + adverb, noun + conjunction + noun, adjective + conjunction + adjective), the two that are considered in this study (Vdobj and amod) cover more than 50% of the total structures (Spina, 2016). Furthermore, while in both relations the order of the two components can be reversed, they have different features in terms of distance between their two components. In the Vdobj word combinations the distance between the two components can be even of several words (Example 2: there are five words between the verb mantenere 'keep' and the direct object promesse 'promises', and the two words are connected by a relative pronoun), while in the case of amod the two words are usually adjacent (Example 3) or near adjacent (Example 4).

Example 2. Non fare promesse che non riuscirai mai a mantenere!  
Don't make promises you will never keep!

Example 3. Elisa mi stava raccontando della sua brutta avventura  
_Elisa was telling me about her bad adventure_

Example 4. Questo è il momento più atteso della giornata  
_This is the most awaited moment of the day_

3.1. Sample texts

We randomly extracted eight texts from a reference corpus of Italian, the Perugia corpus (Spina, 2014; https://lt.eurac.edu/cqpweb/), of the total size of ca. 8,000 tokens, balanced across written (tokens = 4,000) and spoken (tokens = 4,000) registers. We included different text genres: two newspaper articles (a report and an editorial), two school essays and a tourism-related blog post for the written part, and transcriptions of a conference, of a political speech and of the dialogues of a television series for the spoken part. On the one hand, this diversification in registers and text genres allows us to perform a simulation close to the actual extraction of candidate collocations for all the combination types in the whole corpus. On the other hand, it enables us to evaluate the three approaches to this task for register variation, which could affect accuracy.

3.2. The three systems

We used the systems described below to compare three different methods for detecting and extracting candidate collocations from Italian corpora, whose output was compared with a benchmark of human annotation.

P-based approach  The sample texts were POS-tagged using TreeTagger (Schmid, 1994), trained with an ad hoc tagset based on a fine-grained set of 54 POS tags (Spina, 2014). Afterwards, the texts were searched via the Corpus Workbench (CWB) tool (Hardie, 2012) and the Corpus Query Processing (CQP) system by using three separate queries to detect the Vdobj relations and the two positional variants of the amod relations, with the adjective preceding or following the modified nouns. The three queries integrate POS tag sequences (the target ADJ, NOUN and VERB POS tags, as well as those that can potentially be inserted within the two constituents of the combinations, like articles, conjunctions or adverbs) and regex with lemmas to exclude (a list of the most frequent intransitive Italian verbs). The direct output of this regex-over-pos process represents the P-based approach, that was able to identify 549 candidate collocations.

S-based approach  In this approach, a candidate collocation consists of two syntactically related lexical
items. Therefore, the main criterion for detecting a candidate is the presence of a syntactic relation between the two items, in our case, the Vdobj and amod relations. In addition, to be identified as a valid candidate, each pair must satisfy more specific grammatical constraints. For instance, the words involved in the syntactic relations can only be nouns, adjectives or verbs. The sample texts were parsed using the framework of Universal Dependencies for treebank annotation (UD; de Marneffe et al., 2021) and the popular open-source library for advanced NLP in Python spaCy. Artificial intelligence is to date applied in many areas of science (Benedetti et al., 2020; Perri et al., 2022; Milani et al., 2021). The spaCy library is an example of the application of artificial intelligence to linguistic analysis. Since the simple parsing output does not yet represent the S-approach, the complete procedure details are described in section 3.3. The final number of candidate collocations identified by the S-based approach is 685.

**Hybrid approach** The hybrid approach results from merging the two previous approaches. It includes all the common candidates identified by both, as well as those only detected by the P-based approach and those only detected by the S-based approach. The Hybrid approach identified 748 candidate collocations.

### 3.3. Annotation

The output of the three systems was compared to a benchmark obtained by human evaluation. Two Italian trained linguists manually extracted all the Vdobj and amod combinations used in the eight sample texts. The two human annotators only adopted the criterion of the syntactic relations to extract the candidate collocations. Without calculating the inter-annotator agreement, any inter-annotator disagreements were resolved through negotiation until consensus was achieved for all forms. This annotation process resulted in a list of 610 candidate collocations, which served as a benchmark for the following steps.

### 3.4. Computational procedure

Three steps make up the computational process, allowing consistent and thorough data processing. The preliminary pre-processing of the texts was first carried out to enable homogeneous treatment of information. In the second step, the sentences were parsed using spaCy, and a set of rules was implemented to optimise the analysis. Finally, the results were statistically treated. Specifically, the results obtained through the S-approach were compared to those obtained through the P-approach and the Hybrid approach.

#### 3.4.1. The pre-processing of the input texts

The first step involved pre-processing the texts to standardise the input data format and remove any irrelevant elements for analysis. This process included inserting capital letters at the beginning of each sentence and full stops at the end. We removed all whitespace due to typing errors (e.g. double whitespace) or whitespace after the end of a sentence in order to ensure that all input is as clean and error-free as possible. The sentences were then extracted and inserted into a data structure. Each sentence was assigned to a row within a spreadsheet (CSV file), constituting the database for the following stages of the analysis. Having one sentence per line is crucial, as it ensures an easily repeatable analysis and prevents overloading the spaCy parser, which can operate with a limited amount of RAM without requiring excessive resources.

#### 3.4.2. The parsing of input phrases

The second phase of our work was devoted to sentence parsing using spaCy and the rules implemented in Python to recognize adjective modifier dependency (amod) and verb-direct object dependency (Vdobj).

The syntactic analyzer is a Python object obtained by importing the pre-trained spaCy library on the CPU-optimized Italian pipeline called it_core_news_lg\(^1\). The pre-training model occupies 541MB of written text (news and media). The pipeline provided by the model consists of tok2vec, morphologizer, tagger, parser, lemmatizer, attribute_ruler, ner. spaCy was trained with the UD Italian ISDT v2.8 (Italian Stanford Dependency Treebank; Attardi et al., 2015). There are various software libraries that can be used to perform the task of analysing the grammar of a sentence. We opted for spaCy since a version of its Italian language model was released very recently, on 1 Oct 2023\(^2\).

Each sentence in our corpus was analyzed word by word. Given a word, spaCy provides a list of output objects: DepRel, Form, Lemma, UPosTag, XPosTag, head.i.

- DepRel: indicates the syntactic dependence relationship of the word to the main word in the sentence.
- Form: represents the word’s surface form and how it appears in the text.

---

1. https://spaCy.io/models/it#it_core_news_lg
2. https://github.com/explosion/spacy-models/releases/tag/it_core_news_lg-3.7.0
Table 1: Comparison of the performance metrics of the three models across the entire dataset.

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Recall</th>
<th>Precision</th>
<th>F1 Score</th>
<th>Benchmark Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-based</td>
<td>0.70</td>
<td>0.79</td>
<td>0.87</td>
<td>0.83</td>
<td>78.90%</td>
</tr>
<tr>
<td>S-based</td>
<td>0.67</td>
<td>0.86</td>
<td>0.75</td>
<td>0.80</td>
<td>85.88%</td>
</tr>
<tr>
<td>Hybrid</td>
<td>0.67</td>
<td>0.90</td>
<td>0.73</td>
<td>0.80</td>
<td>90.20%</td>
</tr>
</tbody>
</table>

Table 2: Comparison of performance metrics of the three models concerning modifier adjectives.

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Recall</th>
<th>Precision</th>
<th>F1 Score</th>
<th>Benchmark Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-based</td>
<td>0.76</td>
<td>0.83</td>
<td>0.90</td>
<td>0.87</td>
<td>83.43%</td>
</tr>
<tr>
<td>S-based</td>
<td>0.68</td>
<td>0.88</td>
<td>0.75</td>
<td>0.81</td>
<td>88.25%</td>
</tr>
<tr>
<td>Hybrid</td>
<td>0.70</td>
<td>0.93</td>
<td>0.73</td>
<td>0.82</td>
<td>93.37%</td>
</tr>
</tbody>
</table>

Table 3: Comparison of performance metrics of the three models concerning verb-object combination.

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Recall</th>
<th>Precision</th>
<th>F1 Score</th>
<th>Benchmark Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-based</td>
<td>0.63</td>
<td>0.73</td>
<td>0.82</td>
<td>0.77</td>
<td>73.33%</td>
</tr>
<tr>
<td>S-based</td>
<td>0.66</td>
<td>0.83</td>
<td>0.76</td>
<td>0.79</td>
<td>82.96%</td>
</tr>
<tr>
<td>Hybrid</td>
<td>0.64</td>
<td>0.86</td>
<td>0.71</td>
<td>0.78</td>
<td>86.30%</td>
</tr>
</tbody>
</table>

Figure 1: Benchmark Match values per file related to the entire dataset (w=written, s=speech).

- **Lemma**: is the basic form of a word that appears in dictionaries.

- **UPosTag** (Universal Part of Speech Tag): indicates the grammatical category of the word according to the universal POS tag scheme.

- **XPosTag** (Extended Part of Speech Tag): provides an extended POS tag that can include additional information.

- **head.i** (Head index): indicates the index of the word to which the current word is directly connected as a child in the sentence tree structure.

This information alone is not sufficient to fully understand the sentence’s logical structure. Therefore, we identified several syntactic rules translated into Python functions to check the currently examined word and its head and determine whether it is part of an amode or Vdobj word combination. These rules were crucial in increasing the model’s accuracy and precision, by cross-using the values of the different linguistic information provided by the parsing output. Writing these rules is particularly complex, as Italian is a morphologically and syntactically rich language with relatively free word order. For this reason, we proceeded step by step by analyzing the results obtained from time to time.
and checking for incorrectly classified words to add rules, allowing the model to identify as many word combinations as possible. It is important to emphasize that the Python rules are specifically designed for the Italian language.

Some of the most important grammar rules that have been translated into Python code are now given. The first function recognizes a direct verbal object (Vdobj) with the obj relation with root as the dependency, while simultaneously verifying that the UPosTag of the root is VERB.

```python
if token.dep_ == "obj" and
token.head.dep_ == "ROOT"
and token.pos_ == "NOUN"
and token.head.pos_ == "VERB"
```

This rule is able to recognize the combination of words *hanno fama* in Example 5.

**Example 5.** Molto note per le proprietà minerali delle acque sono le sorgenti di nitrodi e di olmitello, le loro virtù terapeutiche *hanno fama* mondiale. 

*Well-known for the mineral properties of the waters are the nitrodi and holmitello springs, their therapeutic virtues are world-renowned.*

Conversely, the function below is designed to identify AMOD when the ‘amod’ relation exists, with ‘obj’ as the dependency, and the UPosTag of the ‘obj’ token is NOUN.

```python
if token.dep_ == "amod" and
token.head.dep_ == "obj"
and token.pos_ == "ADJ"
and token.head.pos_ == "NOUN"
```
The previous rule is able to recognize the word combination *straordinarie proprietà* in Example 6.

**Example 6.** Poi arrivarono i romani e scoprirono le *straordinarie proprietà* delle acque calde. Then the Romans came and discovered the extraordinary properties of hot water.

In total, we created 18 functions to help us in identifying amod and Vdobj syntactic patterns. These functions were subsequently added to a function array. Each word was parsed from the function array, and upon finding a match, the result was saved in our data structure.

```python
for token in line:
    for fun in functionsList:
        if fun(token):
            found="*"
```

At the end of this step, we obtained a data structure without duplicates of all word combinations categorized as amod or Vdobj, which was used as the input for the next step.

### 3.4.3. Statistical analysis of the model

The performance of the three approaches (P-based, S-based and Hybrid) was compared and evaluated through the usual measures of accuracy, precision, recall, F1 score. We defined in addition the benchmark match, which represents the percentage between the predictions generated by the model and the corresponding class labels in the benchmark file. It indicates how well the model aligns with the correct predictions established by the benchmark file, demonstrating its reliability and consistency against a validation dataset. The formula is \( b = \frac{TP}{TP + FN} \times 100\% \), where \( TP \) = True Positive, \( TN \) = True Negative, and \( FN \) = False Negative.

The Hybrid approach outperforms the P- and S-based approaches for the benchmark match and for recall. This better performance is observable across the entire dataset (Table 1), as well as for each of the syntactic relations taken individually (Tables 2 and 3). For the amod relation, the Hybrid approach reaches 93.37% of the benchmark match. This score can be regarded as highly positive in the context of candidate collocation identification. As expected, the P-based approach has better precision and worse recall, suggesting it has the lowest number of false positives but a reduced ability to identify positive instances. Conversely, the S-based approach shows low precision and high recall. It is worth noting that all the three methods have poorer results in detecting Vdobj relations compared to amod relations (Table 3), as in Vdobj relations the two words can be distant and in inverted order. However, the P-based approach is the one that has the most significant loss in benchmark match for Vdobj combinations (-10% compared to the amod relation).

In Figure 1, the benchmark match values related to the three approaches and the entire dataset are plotted as a function of the single sample files. Similar information is shown in Figure 2 about amod relation alone and in Figure 3 about Vdobj combinations alone. The figures allow for an evaluation of possible register influences on detection accuracy. The texts where the three approaches exhibit the most significant differences are two spoken texts, with a relatively formal register: the conference and the political speech, where the P-based approach has the worst results (Figure 1).

Overall, the Hybrid model validates our predictions and aligns more closely with the correct predictions established by the benchmark set, proving its reliability in complying with the gold standard of human annotation. The benefit of integrating the positional part-of-speech and syntactic information for candidate collocation extraction is thus confirmed.

### 4. Conclusions and future work

Focusing on the automatic identification of candidate collocations in Italian corpora for lexicographic purposes, this study reports on an experiment aimed at comparing and evaluating the two most commonly used candidate detection approaches - the P-based and the S-based approach - with a third hybrid method resulting from the integration of the two previous ones. The evaluation of this step is crucial in order to assess the quality of candidate collocations with respect to specific criteria: their grammatical well-formedness (Seretan, 2011). Our assumption was that this quality would benefit from the integration of robust regex-over-pos methods with syntax-based approaches, despite the challenges posed by parsing large amounts of text in a morphosyntactically rich language like Italian. Results show that the Hybrid approach outperforms the two other methods in benchmark match and recall values, confirming the validity of our assumptions. Further work is still needed to optimise the model as precision, accuracy and F1 score obtain higher values with a P-based approach. By implementing additional Python rules, e.g. negative rules (i.e. rules capable of removing false positives) we believe we can enhance the performance of the S-based approach by refining the predictive accuracy while reducing false positives. This, when combined with the outcomes of the P-based approach, is expected to result in an overall enhancement in the model’s performance.

Although the robustness of post-tagging can bal-
ance to some extent the lower accuracy of syntactic parsing, the rules applied in detecting syntactic relations after parsing need refinements to reduce errors resulting from false positives. One limitation of this experiment derives from using only two syntactic relations, whereas the final procedure for dictionary entry selection will need to consider a larger set of relations. However, the conclusion that can be drawn is that pursuing a hybrid approach to candidate collocation identification is worthwhile, as it leads to an improvement in the quality of results.

5. Acknowledgements

The research has been funded by the Italian Ministry of Research (MUR), PRIN: Research Projects of Major National Interest — Call 2022 - Prot. 2022HXZR5E. The title of the project is: DICI-A: A Learner Dictionary of Italian Collocations.

6. References


Multiword Expressions between the Corpus and the Lexicon: Universality, Idiosyncrasy and the Lexicon-Corpus Interface


1 Romanian Academy Research Institute for Artificial Intelligence, vergi@racai.ro
2 Institute for Language and Speech Processing, Athena Research Centre, voula@athenarc.gr
3 Heinrich Heine University Düsseldorf, evang@hhu.de
4 UFAL MFF, Charles University, zeman@ufal.mff.cuni.cz
5 Institute of Information and Communication Technologies, BAS, petya@bultreebank.org
6 Leiden University, c.p.a.tiberius@hum.leidenuniv.nl
7 Jožef Stefan Institute, simon.krek@ijs.si
8 Institute for Bulgarian Language, BAS, marks@athenarc.gr
9 University of Belgrade, ranka@rgf.rs
10 University of Augsburg, christian.chiarcos@uni-a.de

Abstract

We present ongoing work towards defining a lexicon-corpus interface to serve as a benchmark in the representation of multiword expressions (of various types – nominal, verbal, etc.) in dedicated lexica and the linking of these entries to their corpus occurrences. The final aim is the harnessing of such resources for the automatic identification of multiword expressions in a text. The involvement of several natural languages aims at the universality of a solution not centered on a particular language, and also accommodating idiosyncrasies. Challenges in the lexicographic description of multiword expressions are discussed, the current status of lexica dedicated to this linguistic phenomenon is outlined, as well as the solution we envisage for creating an ecosystem of interlinked lexica and corpora containing and, respectively, annotated with multiword expressions.

Keywords: multiword expression lexicon, corpus, proof-of-concept lexicon encoding

1. Introduction

In the last decade, the PARSEME COST Action (Savary et al., 2015) created the prerequisites for annotating corpora with multiword expressions (MWEs), mainly verbal ones. Consistent guidelines and an infrastructure for ensuring annotation consistency were developed, while the interaction among the members of the community was made possible by the COST Action and extended even beyond its duration. A corpus was created for 26 languages (Savary et al., 2023), in which verbal MWEs (VMWEs) were annotated according to the established guidelines. Meanwhile, a new COST Action, UniDive2, is gathering the community again, simultaneously increasing in size and allowing for the development of guidelines for annotating MWEs of other parts of speech, and eventually for further annotation of corpora with the new MWE types, as well as for increasing the number of languages represented in the corpus so far. At the same time, UniDive builds on Universal Dependencies (UD) (de Marneffe et al., 2021), which posits standardized guidelines for tokenization, lemmatization and morphosyntactic annotation in treebanks of languages.

Despite the abundance of large bodies of annotated corpora and large language models, systems still fail to adequately identify MWEs and thus the need for lexica that are specifically designed to handle MWEs within the context of Natural Language Processing (NLP) (Savary et al., 2019b). Within UniDive, Working Group 23 seeks to take this further and to schematize the steps needed towards creating an ecosystem in which annotated corpora and MWE lexica are linked together, intra- and interlingually and are used to facilitate MWE identification in a way that universality and idiosyncrasy are taken into account.

In this paper, we report on original (ongoing) work towards designing this lexicon-corpus interface. The paper is structured as follows: we first outline our goals and the challenges we need to face (Section 2); then, an overview of the current

---

1 https://parsmefr.lis-lab.fr/parseme-st-guidelines/1.2/?page=030_Categories_of_VMWEs
2 https://unidive.lisn.upsaclay.fr/doku.php?id=wg2:wg2
MWE dedicated lexica and the results of a survey aimed at better accounting for universality are presented (Section 3 and Section 4 respectively). The initial steps towards designing the lexicon-corpus interface, in a standardized manner with all its advantages are presented in Section 5. We outline the minimal requirements for encoding MWEs in computational lexica, with an eye to their interlinking with annotated corpora, in Section 6. Our conclusion is presented in Section 7.

2. Towards a lexicon-corpus interface: goals and challenges

For many decades, MWE-aware lexica have contributed a much larger set of MWEs than (annotated) corpora can do, as MWEs are rather rare in texts (Savary et al., 2019a), and to model their linguistic properties, namely, non-compositionality, lexical fixedness, discontinuity, potential modifiers of components, word order variation, etc. However, the representation of MWEs in hand-crafted lexica is far from homogeneous and even incomplete. At the same time, annotated corpora have been used as major operational tools for language modelling and the backbones of data-driven NLP methods. Yet, they seem inadequate when unseen MWEs are at stake, as these unseen ones may well be characterised by lexical combinations or syntactic structures that did not occur in annotated corpora and are thus hard to be identified automatically. Therefore, linking corpora and lexica would be beneficial for the robust MWE identification (Savary et al., 2019b).

As of now, corpora and lexica remain to a great extent disconnected, with a few exceptions (Odijk, 2013; Markantonatou et al., 2019; Autelli, 2020) in which examples are extracted from corpora and added to the lexicon to illustrate the use of the MWEs.

Our goal is to design a lexicon-corpus interface that leverages MWE identification cross-linguistically. Three are the major challenges: (a) the harmonisation of corpora and lexica by also accounting for universality and diversity, (b) the efficient encoding of MWEs of all grammatical categories cross-linguistically, and (c) the adoption of the appropriate mechanisms and tools for linking lexica and corpora. Our work has been organised along three axes:

i capturing universality via cross-language unification of lexical features,

ii designing a lexicon-corpus interface usable for several languages, and

iii proof-of-concept encoding of MWEs based on the outcomes of (i) and (ii).

3. MWEs in computational lexica: state-of-the-art

In order to overview the state-of-the-art in the development of computational lexica of MWEs, we collected information about resources in a structured and systematic way, focusing on those published since 2016, as those published before this year were included in the survey performed within the COST Action PARSEME (Losnegaard et al., 2016). We have retrieved information for 75 resources from the following sources: European Language Grid repository, using the keyword “expressions” in the category Lexical/Conceptual resources; the ACL Anthology, in which we also used a keyword search (multiword, idiom, phraseology, etc.); the Phraseology and Multiword Expressions book series published by Language Science Press, and Europhras conferences, which were manually examined.

The data was harmonised aiming at a uniform and comprehensive description of the identified resources. It was organized in the following sections: General information (general or dedicated lexicon, mono- or multi-lingual), Corpus (in cases where the resource is related to a corpus), Resource (size, owner, licensing), Lemma & Representations (whether the resource provides information about the “lemma” of the MWE and its morphosyntactic properties), Syntax (details about syntactic information about the MWEs), Semantics (whether the resource provides semantic information about the MWEs and of what type) and References (major publication(s) about the identified resource).

The general picture obtained so far shows that:

• 72% of the resources are aimed for NLP use.

• More than 40 languages and dialects are represented, mostly Indoeuropean ones.

• 70.7% of the resources are monolingual, 18.7% bilingual and 10.6% multilingual.

• Most datasets were acquired manually or semi-automatically (automatically collected and manually verified).

• Only 24% of the resources are linked to a corpus and 12% are linked to other resources. The resources are usually linked to small purpose-built corpora. Usage examples are sometimes collected from a large representative corpus (without linking to the corpus).

• With regards to the encoded information, 45% of the resources provide comprehensive description of MWEs (including morphological, syntactic and semantic information). Semantic information, in particular, is extremely diverse.
The survey on MWE lexica raises several significant questions related to handling universality and diversity. First, most resources assume that a MWE entry is the coupling of a “lemma” form with a meaning. The definition of the “lemma” form is an open issue (see also section 4). In addition, often MWEs have “lemma” variants due not to grammatical phenomena but, for instance, to mutually exclusive choices of functional words or to the optionality of articles, and still, all these forms correspond to one meaning. It has been up to each resource’s authors to decide which of these forms represents the MWE as its “lemma form” and how all these forms are related among them. As a result, different resources encode essentially the same MWE under different entries, as shown in Ex. 1 for Greek. Guidelines are needed even at this elementary level.

(1)

vazo (ti) thilia sto lemo kapiou
put (the) noose to.ADP.the neck someone.GEN
vazo (ti) thilia giro apo to lemo kapiou
put (the) noose around.ADV from.ADP the neck
someone.GEN
‘to force someone to be involved in an unpleasant situation’

Second, various resources encode a different set of morphosyntactic and semantic features, in some cases with different degree of granularity, which poses a problem for their combined use and mutual enrichment. Guidelines handling the diversity among languages, in terms of morphological and syntactic properties of MWEs would facilitate their uniform representation and boost their NLP applications.

4. Universality: on cracking hard nuts

The notion of “word” is central to UD, but it is hard to define it in the context of the various typologically diverse languages. Thus, as a starting point of comparison, the strategy proposed by Haspelmath (2023) is followed. According to Haspelmath, “A word is (i) a free morph, or (ii) a clitic, or (iii) a root or a compound possibly augmented by nonrequired affixes and augmented by required affixes if there are any.” He also defines all the terms that constitute this definition: a free morph, a clitic, roots of various kinds, a compound, required/nonrequired affixes. Even with this typologically friendly approach, there exist a number challenges in a cross-lingual context. The main ones are: demarcation of clitics (words) vs. affixes (non-words), analysis of the compounds, marking the places of contraction splits.

For better modeling of data on the word level, a survey was conducted with Haspelmath’s criteria. Responses for 43 languages were received. Based on that, a second version of the survey is being prepared that will allow for better comparison among language-specific properties. This new survey will target UD and non-UD languages and ask for examples of all of Haspelmath’s word types that occur in the language. For UD languages, it will also ask for divergences between Haspelmath words and treebank words.

Although lemmatization may seem a very straightforward process and a solved task, this is quite misleading, because there exists a number of problems both in the lemmatization of words and in that of MWEs. The guidelines from UD and PARSEME say relatively little about lemmatization from a linguistic point of view. The focus there has been predominantly on tokenization and morphosyntactic analysis before the application of various linguistic tests and proposed classifications. For example, the relation between a token and a word is discussed in Savary et al. (2018): a token coincides with a word, several tokens constitute a multiword and one multiword contains several tokens. In UD the following is said: “The LEMMA field should contain the canonical or base form of the word, which is the form typically found in dictionaries. If a language is agglutinative, this is typically the form with no inflectional affixes; in fusional languages, the lemma is usually the result of a language-particular convention. If the lemma is not available, an underscore (“_”) can be used to indicate its absence.”. It means that the majority of decisions are left within the hands of treebank providers. Also, the guidelines say that “Except perhaps in rare cases of suppletion, one form should be chosen as the lemma of a verb, noun, determiner, or pronoun paradigm”.

Various frameworks and annotation schemes apply different strategies to lemmatization and identify various issues. For example, Mambrini and Passarotti (2019) point to the following challenges in relation to Latin: the graphical representation, the spelling, the word ending, the representative paradigmatic slot, the homographic lemmas, the ambiguity in choosing the lemma, for example for participles that are hybrid forms and can be viewed either as verb forms by origin or as adjectives in some of their usages. The same holds for the deadjectival adverbs that can be viewed as part of the adjective paradigm or have their own lemmas. In (Mubarak, 2018) it is shown that the lemmatization task is quite complex for Arabic. The main linguistic problem is the mismatch between a word with a diacritic and its context (e.g. nouns and adjectives).

We outline only some of the challenges here. They refer to the issues of selecting the right form as a lemma, the existence of two options, the graphic representation varieties, the spelling specifics, the relation between inflection and derivation, the relation between orthographic words, their meaning
and their spelling. The presented examples below feature some frequent lemma assigning problems across annotation schemes – within a single language and among languages. The list is not exhaustive, but it reflects the situation in many other languages and frameworks. Since this task is work in progress, the plan is to study the lemmatization decisions in the various UD treebanks and in PARSEME corpora as being already very multilingual and as sources of integration of these two frameworks and data, and also beyond them – through investigating papers on different language families, as well as through questionnaires.

**Lemmatization challenges of some words and tokens**

- **Pronouns.** In some languages (like Bulgarian, Czech, Maltese) there are short and long forms of some pronouns (e.g. personal), or strong and weak ones (like in Greek and Italian). Thus, the following possibilities for lemmatization exist for the short 3rd person pronouns in Czech, for example: a) the lemma equals the wordform itself ([cs]: ho-3P.MASC.SG.ACC.SHORT ‘him’), b) the lemma goes to the long 3rd person form ([cs]: něho-3P.MASC.SG.ACC.SHORT ‘him’), c) the lemma goes to the nominative, masculine, 3rd person form ([cs]: on-3P.MASC.SG.NOM ‘he’), while in d) the lemma is the pronoun in 1st person, singular, nominative as the less marked form ([cs]: já-1P.SG.NOM ‘I’). Thus, different strategies can be applied with varying depth until reaching the lemma.

- **Doublets.** There are doublet verbs that share the same paradigm. For example, the same lemma verb with two different endings ([bg]: zna-m and zna-ya (lit. ‘know-1’) ‘to know’; or the same lemma adjective with two different variants ([bg]: sash’ti and sashti ‘same-M.SG’). Thus, one of the doublets might be selected as representative, but it is sometimes hard to make such a selection.

- **Numbers.** In text data, numbers can occur as words or as digits. Should both representations of the same number have the same lemma? And if so, then which one?

- **Negated words.** This problem relates also to graphic conventions. In some languages, the negation of a word is written together, for example – as a prefix. In Bulgarian, this holds for the nominals, in Czech this holds also for verbs, while in Romanian it holds for some nominals and for three out of the four non-finite forms of a verb (only for participle, supine and gerund, but not for infinitive). Should the lemma of the negated word be its positive counterpart (meaning that negation is treated rather like inflection than derivation)?

- **Diminutives.** Although the process of making diminutives is derivational, it is still not clear whether the lemma of the word should be the diminutive or the original word. According to the current UD guidelines, the lemma does not remove derivational morphology. If such a strategy is followed, the lemma should be the diminutive. However, if most of the diminutives are not part of the dictionary, then there might be problems during the next NLP processing tasks.

**Lemmatization challenges of some MWEs**

- **Compounding.** In many languages, a compound (traditionally a word with (at least) two roots) can be written differently: as two words, as one word or with a hyphen. Compare in Bulgarian the double spelling: biznes plan (two words) and biznesplan (one word), in English business plan (two words) and in German Businessplan (one word). A problem arises when trying to offer a uniform analysis of these compounds within a language and across languages.

- **(Quasi)reflexive verbs.** Even within one language family like the Slavic languages, the quasi-reflexive particle can be either a separate word ([bg]: smeyà se, [cs]: smát se ‘to laugh’) or part of the word ([uk]: smijatysja ‘to laugh’). The reflexive pronouns are part of the word also in some Romance languages ([es]: lavarse ‘to wash oneself’) and not in others ([ro]: se spăla ‘to wash oneself’), but in the non-reflexive meaning they lose this clitic (lavar ‘to wash something/someone’). The question is whether the lemma is defined within each language/language family on formal criteria, or there might be possibilities to create some cross-linguistic strategies.

5. **Linking MWE lexicon entries with their occurrences in corpora**

Publishing language resources as Linked Data enhances accessibility, interoperability, semantic enrichment, community collaboration, and the promotion of open science. These contribute to the advancement of linguistic research, language technology, and cross-disciplinary insights.

Analyzing unique language patterns across different languages can benefit from sharing aligned and annotated corpus data in a format that complies with community standards like the NLP Interchange
Format (NIF) (Hellmann et al., 2012, 2013) and CoNLL-RDF (Chiarcos and Fäth, 2017; Chiarcos and Glaser, 2020). CoNLL-RDF is a simplified version of NIF that aligns with tab-separated formats, such as CoNLL, CoNLL-U for Universal Dependencies, and Parseme-TSV for PARSEME.

Working towards the objective of designing a lexicon-corpus interface and prove its functionality, we will expand the existing ELEXIS-WSD Parallel Sense-Annotated Corpus (Martelli et al., 2023). Currently at version 1.1, it can be accessed from the CLARIN.SI repository (the current draft version of the FrAC specification is found under https://github.com/ontolex/frequency-attestation-corpus-information) and upgrading the annotation to enable linking MWE lexicon entries with their occurrences in the corpora.

Moreover, these resources should also be published as Linked Data (using NIF) to facilitate linking with the sense repository of the corpus. For the ELEXIS dictionary data, the OntoLex vocabulary, a widely used community standard for machine-readable lexical resources in the context of RDF, Link Data, and Semantic Web technologies (McCrae et al., 2017), will be considered, as it is currently the foundation for the majority of lexical data available on the web of data.

Apart from the core module Lemon with general data structures, OntoLex modules relevant to MWEs include the module for the internal structure and combinatory semantics of MWEs Decomp, and MWE morphology Morph module. The new module for Frequency, Attestations, and Corpus-based Information (FrAC) (Chiarcos et al., 2022a,b) supports linking lexica with corpora in many aspects of information relevant to the joint work with corpora and dictionaries. Lexicog (Bosque-Gil et al., 2019) is a module for lexicography that addresses structures and annotations commonly found in lexicography. It is designed to operate in combination with OntoLex for the representation of dictionaries and any other linguistic resource containing lexicographic data.

An attempt at leveraging Linked Data, NIF, and CoNLL-U for Enhanced Annotation in Sentence Aligned Parallel Corpora is reported in the literature and could be followed (Stanković et al., 2024).

6. Proof-of-concept lexical encoding of MWEs

Taking the above into consideration, a proof-of-concept lexical encoding of MWEs in NLP lexica, that also maintains the lexicon-corpus interface, should minimally abide by the following requirements:

• a definition of the notion of “word” that is as universal as possible,
• a shared understanding of MWEs that can be annotated in corpora and then linked with lexicon entries (both the MWE as a whole and its components), including all types of MWEs (not only nominal and verbal),
• centralised guidelines for lexicon encoding regarding, i.e., the notions of lemma, canonical form, lexical features, etc.,
• a uniform representation of the syntactic properties of MWEs, and
• tools and mechanisms for linking MWE entries with their occurrences in corpora.

7. Conclusion

In an effort to create an ecosystem of interlinked MWE-dedicated lexica and annotated corpora, with an eye to universality and accommodating the languages specificities, we have already painted the current landscape of this field and are striving to find solutions for cracking the hard nuts (syntactic word definition, word and MWE lemmatization, lexical features, etc.) and to create guidelines for MWE lexicographic description. Development of linguistic resources for various languages in a harmonized way and their interlinking using standardization methods can only lead to the progress of language technology, as well as serve as a model for low-resourced languages in their endeavour to catch up with domain’s evolution, speeding this process due to the benefits that Linked Data can offer (Bosque-Gil et al., 2022).

8. Acknowledgments

This paper is funded by the CA21167 COST Action UniDive, supported by COST (European Cooperation in Science and Technology).

9. Bibliographical References

University of Białystok Publishing House., Białystok.


10. Language Resource References


Agata Savary, Silvio Cordeiro, Timm Lichte, Carlos Ramisch, Uxoa Inurrieta, and Voula Giouli. 2019a. Literal occurrences of multiword expressions: Rare birds that cause a stir. 112:5–54.


Annotation of Multiword Expressions in the SUK 1.0 Training Corpus of Slovene: Lessons Learned and Future Steps

Jaka Čibej, Polona Gantar, Mija Bon
Faculty of Arts, University of Ljubljana
Aškerčeva cesta 2, 1000 Ljubljana, Slovenia
{jaka.cibej, apolonija.gantar, mija.bon}@ff.uni-lj.si

Abstract

Recent progress within the UniDive COST Action on the compilation of universal guidelines for the annotation of non-verbal multiword expressions (MWEs) has provided an opportunity to improve and expand the work previously done within the PARSEME COST Action on the annotation of verbal multiword expressions in the SUK 1.0 Training Corpus of Slovene. A segment of the training corpus had already been annotated with verbal MWEs during PARSEME. As a follow-up and part of the New Grammar of Modern Standard Slovene (NSSSS) project, the same segment was annotated with non-verbal MWEs, resulting in approximately 6,500 sentences annotated by at least three annotators (described in Gantar et al., 2019). Since then, the entire SUK 1.0 was also manually annotated with UD-part-of-speech tags. In this paper, we present an analysis of the MWE annotations exported from the corpus along with their part-of-speech structures through the lens of Universal Dependencies. We discuss the usefulness of the data in terms of potential insight for the further compilation and fine-tuning of guidelines particularly for non-verbal MWEs, and conclude with our plans for future work.

Keywords: multiword expressions, Universal Dependencies, Slovene

1. Introduction

Slovene was one of the languages involved in the PARSEME COST Action 1. As part of the activities, 11,411 sentences (approx. 41 %) of the ssj500k 2.1 Slovene Training Corpus (Krek et al., 2018)2 were annotated with verbal MWEs (Gantar et al., 2017) categorized according to the PARSEME guidelines and MWE-tests (Savary et al., 2018). Work on Slovene MWEs within the same corpus then continued after the conclusion of PARSEME within the national project titled New Grammar of Contemporary Standard Slovene: Sources and Methods 3, during which non-verbal MWE annotations were added to 6,500 sentences (a subset of the 11,411 sentences annotated within PARSEME). Non-verbal MWEs were annotated (the process is described in more detail in (Gantar et al., 2019)) according to a set of guidelines designed primarily from the point of view of inclusion of MWEs in dictionaries, while the categorization principles followed the definitions used in the compilation of Slovene Lexical Dictionary Database (Gantar and Krek, 2011) and the Digital Dictionary Database of Slovene (Kosem et al., 2021). However, the annotations have so far not been included in the SUK 1.0 corpus, pending an additional curation and resolution of crucial questions, mainly which of the annotated spans should be considered MWEs, particularly with regard to multiword combinations with varying levels of terminologicalness.

Recent advances within the UniDive COST Action 4, which among its tasks (specifically in Task 1.2) also includes the extension of the PARSEME verbal MWE annotation guidelines 5 with non-verbal MWEs, have provided an opportunity to continue the work already done on Slovene MWE annotations in the SUK 1.0 corpus within other projects, as well as to compare our own MWE-categorization with the one adopted within UniDive. At the time of writing this paper, the UniDive non-verbal MWE annotation guidelines contain no examples of Slovene MWEs, and a discussion is still underway. In addition to these examples, the lessons from the annotation of the SUK 1.0 corpus may provide a number of valuable insights during the initial phase of uni-

---


2. Since then, the ssj500k training corpus was extended with several other datasets and underwent a rebranding, now being called the SUK 1.0 Training Corpus of Slovene (Arhar Holdt et al., 2022). In this paper, we refer to it using the new name unless we specifically refer to an older version. The SUK 1.0 corpus consists mostly of newspaper texts, magazines, and internet texts, with a small percentage of fiction and non-fiction.


5. PARSEME Annotation guidelines 1.3 - https://parseme.fr/parseme-st-guidelines/1.3/
fying the PARSEME annotation scheme with Universal Dependencies (Savary et al., 2023). While the data only covers Slovene, its advantage is that several statistical calculations were made based on the annotations, including for example the scope of MWE annotation and length overlap, as well as inter-annotator agreement (each sentence was annotated by at least three annotators). In the paper, we discuss the annotated MWEs and focus predominantly on the points of disagreement and lessons learned that may prove useful for the compilation of MWE annotation guidelines within UniDive. The paper is structured as follows: we first provide a short overview of related work on MWEs (Section 2) and describe the data on annotated MWEs exported from the SUK 1.0 corpus (Section 3), then provide an analysis (Section 4). We conclude the paper with a discussion on the usefulness of the data within UniDive and a list of potential future steps in our work.

2. Related Work

MWEs still pose a problem for NLP tools such as machine translation systems, word sense disambiguation, or computational lexicography (e.g. MWE detection in corpora). A number of endeavors have been undertaken to provide training or evaluation datasets annotated with MWEs, both monolingual (Adali et al., 2016 for Turkish; Candido et al., 2020 for French; Kato et al., 2018 and Schneider et al., 2014 for English; Mohamed et al., 2022 for Arabic; Souza and Freitas, 2023 for Portuguese) and multilingual (Monti et al., 2015; Han et al., 2020; Savary et al., 2018).

So far, no Slovene manually annotated corpus includes comprehensive and systematic annotations of MWEs; aside from the already mentioned PARSEME verbal MWE annotations in the ssj500k 2.1 Training Corpus (Gantar et al., 2017) which also serves as the Slovene UD Treebank, a small dataset for the automatic detection of idiomatic expressions has also been made by Škvorc et al. (2022) in order to facilitate idiomatic expression extraction using contextual embeddings. There is also the Slovene subcorpus of the ELEXIS-WSD Parallel Sense-Annotated Corpus (Martelli et al., 2021); however, MWEs within the corpus have not been categorized and only their spans have been annotated, while the corpus itself was primarily compiled for word sense disambiguation focused on single word units.

The first step toward extending the SUK 1.0 corpus with comprehensive MWE annotations was made by (Gantar et al., 2019) by conducting an experimental annotation campaign to identify potential MWE candidates. We discuss the results in the following sections.

3. Data Description

The annotation process and the typology used to annotate MWEs in SUK 1.0 was described in detail by (Gantar et al., 2019), so we only provide a brief overview here. The main goal of the task was to annotate non-verbal multiword expressions according to a typology that defines two main subgroups of MWEs: (a) lexical units, which require an explanation (due to them being characterized by a certain degree of semantic non-compositionality), and (b) lexico-grammatical units, which are semantically relatively transparent (they complement or disambiguate the sense description of a headword (e.g. collocations) or they play a role of syntactic connectors or discourse organizers in language).

Multilingual lexical units are further divided into fixed expressions (which typically cover terminological expressions such as črna luknja ‘black hole’ in the sense of an astronomic phenomenon) and phraseological units (which typically express a metaphorical or pragmatic meaning, such as princ na bele konju ‘lit. ‘prince on a white horse’; ‘knight in shining armor’).

The annotators were thus tasked with annotating MWEs as either phraseological units (PU), fixed expressions (FE), or syntactic combinations (SC). It should be noted that this is a parallel categorization, so the existing verbal MWEs annotated within PARSEME were also assigned additional categories according to this system. In this paper, we focus on the annotated UD POS-structures and patterns, not the categorization according to our own typology; more detailed results of the categorization were already presented in Gantar et al.,

---

6 This categorization follows the organization of language data in the Digital Dictionary Database of Slovene (Kosem et al., 2021), where the main criterion to distinguish different types of MWEs depends on whether the MWE is a semantically independent or dependent unit.

7 In retrospect, it should be mentioned that the decision to explicitly categorize discourse organizers as lexico-grammatical units caused some disagreement during annotation; if a discourse organizer (such as v bistvu ‘in fact’, ‘actually’) requires a semantic explanation and plays a pragmatic role in the sentence that needs to be explained in a dictionary, it should be categorized as a phraseological unit (PU), which falls under lexical units.
The annotation resulted in a total of 15,727 MWE annotations in the first 6,500 sentences of the SUK 1.0 corpus. Each sentence was annotated by at least 3 annotators (see Gantar et al., 2019), so a potential MWE-candidate within an individual sentence has up to three annotations (depending on whether the annotator identified the span as a MWE). For instance, in the sentence below, two annotators identified one MWE candidate and each provided an annotation; one annotated v nasprotju (lit. ‘in contradiction’) while the other annotated v nasprotju s (lit. ‘in contradiction with’).

sl Toda [(v nasprotju) s] svojimi sorodniki sodijo kaneloni (cannello = cevka) šele slabih sto let k italijanski testeninski klasiki.

en But contrary to their relatives, cannelloni (cannello = tube) have been a part of the Italian pasta classics for less than one hundred years.

A total of 8,864 MWE candidates were annotated in the corpus, consisting of 6,385 different potential MWEs.8

Since the annotations were made, a section of the SUK 1.0 corpus was also manually annotated with UD-part-of-speech tags, UD dependency relations, and named entities (see Arhar Holdt et al., 2023); this includes the 6,500 sentences annotated with both verbal and non-verbal MWEs, which enables us to export MWE annotations along with UD part-of-speech tags, dependency relations, and named entities, and observe potential patterns as well as points of potential disagreement. We provide a thorough analysis in Section 4 below.

## 4. Analysis

As shown in Table 1, the MWE candidates were annotated by 10 annotators; two of which (A and B) were reference annotators involved in the compilation of the annotation guidelines. The rest were students of linguistics at the University of Ljubljana. The distribution of annotations and the average number of MWE annotations per sentence shows that most of the annotators annotated MWEs similarly frequently to the reference annotators (approx. 0.5–0.6 MWEs per sentence), with two outliers, who were either too liberal (annotator I) or too strict (annotator J).

Out of 8,864 annotated MWE candidates, 5,023 (56.67%) were assigned a single annotation, 2,103 (23.73%) two annotations, and 1,738 (19.61%) three or more annotations. As shown in Table 2, a large portion of single annotations (almost 40%) were made by the most liberal annotator (I), but a significant percentage was provided by other annotators as well, including one of the reference annotators (B, with approx. 11%). As the identification of MWEs is a difficult task, a certain degree of disagreement is to be expected. In the following subsections, we further analyze the annotations in order to discover any recurring misinterpretations that could point to potential gaps in the annotation guidelines.

### 4.1. Part-of-Speech Structure

Based on the annotated tokens and their UD part-of-speech tags, the annotated MWE candidates cover 920 different structures, with the top 17 accounting for approx. 65% of all annotations (see Table 3). Each of these covers more than 1% of the annotations, while the other categories cover less. The majority of the annotations are non-verbal, with verbs

---

8The 6,385 different candidates were counted based on the alphabetical combinations of lemmas within annotated spans.

---

Table 1: Table of MWE-annotations showing individual annotators, the number of MWE-annotations they made in the corpus, the number of all sentences annotated by them, the percentage of all annotations made, and the number of MWEs per sentence.

<table>
<thead>
<tr>
<th>Ann.</th>
<th>MWEs</th>
<th>Sent.</th>
<th>% MWE/Sent.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>292</td>
<td>500</td>
<td>1.86%</td>
</tr>
<tr>
<td>B</td>
<td>3,111</td>
<td>6,500</td>
<td>19.86%</td>
</tr>
<tr>
<td>C</td>
<td>1,742</td>
<td>2,000</td>
<td>11.12%</td>
</tr>
<tr>
<td>D</td>
<td>1,716</td>
<td>2,000</td>
<td>10.95%</td>
</tr>
<tr>
<td>E</td>
<td>1,124</td>
<td>2,000</td>
<td>7.17%</td>
</tr>
<tr>
<td>F</td>
<td>1,367</td>
<td>2,000</td>
<td>8.73%</td>
</tr>
<tr>
<td>G</td>
<td>903</td>
<td>2,000</td>
<td>5.76%</td>
</tr>
<tr>
<td>H</td>
<td>1,467</td>
<td>2,000</td>
<td>9.36%</td>
</tr>
<tr>
<td>I</td>
<td>3,563</td>
<td>2,000</td>
<td>22.74%</td>
</tr>
<tr>
<td>J</td>
<td>382</td>
<td>2,000</td>
<td>2.44%</td>
</tr>
</tbody>
</table>

Table 2: Distribution of single-annotation MWE candidates across annotators.

<table>
<thead>
<tr>
<th>Ann.</th>
<th>Single cand.</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1,953</td>
<td>38.86%</td>
</tr>
<tr>
<td>D</td>
<td>696</td>
<td>13.86%</td>
</tr>
<tr>
<td>C</td>
<td>601</td>
<td>11.96%</td>
</tr>
<tr>
<td>B</td>
<td>547</td>
<td>10.89%</td>
</tr>
<tr>
<td>H</td>
<td>380</td>
<td>7.57%</td>
</tr>
<tr>
<td>F</td>
<td>313</td>
<td>6.23%</td>
</tr>
<tr>
<td>G</td>
<td>307</td>
<td>6.11%</td>
</tr>
<tr>
<td>E</td>
<td>126</td>
<td>2.51%</td>
</tr>
<tr>
<td>J</td>
<td>91</td>
<td>1.81%</td>
</tr>
<tr>
<td>A</td>
<td>9</td>
<td>0.18%</td>
</tr>
</tbody>
</table>
Table 3: Distribution of MWE annotations based on their UD part-of-speech structure.

<table>
<thead>
<tr>
<th>Structure</th>
<th>MWE Ann.</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJ NOUN</td>
<td>4,550</td>
<td>29.04%</td>
</tr>
<tr>
<td>ADP NOUN</td>
<td>2,053</td>
<td>13.10%</td>
</tr>
<tr>
<td>ADP DET</td>
<td>401</td>
<td>2.56%</td>
</tr>
<tr>
<td>NOUN NOUN</td>
<td>391</td>
<td>2.50%</td>
</tr>
<tr>
<td>VERB ADP NOUN</td>
<td>360</td>
<td>2.30%</td>
</tr>
<tr>
<td>PART AUX</td>
<td>353</td>
<td>2.25%</td>
</tr>
<tr>
<td>ADP DET NOUN</td>
<td>298</td>
<td>1.90%</td>
</tr>
<tr>
<td>PART ADV</td>
<td>228</td>
<td>1.46%</td>
</tr>
<tr>
<td>ADJ ADJ NOUN</td>
<td>224</td>
<td>1.43%</td>
</tr>
<tr>
<td>ADP ADJ NOUN</td>
<td>214</td>
<td>1.37%</td>
</tr>
<tr>
<td>NOUN ADP NOUN</td>
<td>187</td>
<td>1.19%</td>
</tr>
<tr>
<td>VERB NOUN</td>
<td>186</td>
<td>1.19%</td>
</tr>
<tr>
<td>ADP ADJ</td>
<td>174</td>
<td>1.11%</td>
</tr>
<tr>
<td>DET SCONJ</td>
<td>174</td>
<td>1.11%</td>
</tr>
<tr>
<td>ADV SCONJ</td>
<td>171</td>
<td>1.09%</td>
</tr>
<tr>
<td>ADP NOUN ADP</td>
<td>168</td>
<td>1.07%</td>
</tr>
<tr>
<td>ADP ADP</td>
<td>165</td>
<td>1.05%</td>
</tr>
<tr>
<td>Other</td>
<td>5,658</td>
<td>35.98%</td>
</tr>
</tbody>
</table>

Table 3: Distribution of MWE annotations based on their UD part-of-speech structure.

We analyzed the distribution of the part-of-speech structures in terms of how prone they were to single annotations in order to check whether any structure is more problematic for MWE identification. Table 4 shows the 10 most frequent part-of-speech structures that are also more typical of single annotations compared to all annotations (i.e. according to the ratio in the last column, they are more likely to be annotated by just a single annotator and less likely to be annotated multiple times).

An analysis of the single annotation examples with these structures reveals a number of problematic groups, particularly within structures with a nominal distribution (e.g. NOUN NOUN, NOUN ADP NOUN). First, there are terminological candidates that may be somewhat compositional, but have a specific meaning within a certain field (e.g. omejevalnik vrtljajev ‘rev limiter’, raziskave tržišča ‘market research’, vitamin C, ‘vitamin C’). In some cases, the annotated spans are collocations that are semantically transparent, but very typical (e.g. kraj zločina, lit. ‘place of the crime’, ‘scene of the crime’; balzam za ustrnice, ‘lip balm’). Secondly, some spans denote titles or functions (e.g. poveljnik straže, ‘captain of the guard’, hranilec družine, lit. ‘feeder of the family’, ‘family provider’) or even members of an association or organization (e.g. sestre usmiljenke, ‘Sisters of Mercy’), which should be treated more as named entities despite not being capitalized. Similarly, the third problematic group contains spans that can be interpreted as named entities, but that is not entirely clear when the span is spelled with no capitalization and the context is somewhat ambiguous whether the examples refer to concrete instances or a general concept (e.g. liga prvakov, ‘league of champions’, ministrstvo za finance, ‘ministry of finance’). In addition, examples contain phrases in which one of the components exhibits a metaphoric meaning - e.g. gostja večera, ‘guest of the evening’ in the sense of ‘the guest of tonight’s show’), which prompts the annotator to treat the span as non-compositional.

Next, there are several grammatical constructions that were mistakenly annotated as multiword expressions, such as combinations of prepositions and relative pronouns (ADP DET; v kateri ‘in which’, po kateri ‘after which’, h kateri ‘to which’); some of the annotators probably annotated these because kateri as a relative pronoun only occurs next to prepositions, so they treated both components as a single unit. Similarly, sequences of prepositions and demonstrative pronouns (glede tega ‘regarding this’, iz tega ‘from this’) occurring in a very vague context could have prompted to treat them as non-compositional, as in the example below:

sl Država s tem priznava, da je prostovoljnih vojakov premalo, če ne kar nič.

en With this, the State recognizes that there are too few voluntary soldiers, if any.

Interestingly, some candidates with similar part-of-speech structures (either ADP DET or ADP PRON) do represent legitimate MWEs (e.g. po svoje, ‘in its own way’; pri nas, lit. ‘at us’, ‘in our country’), but were only annotated once, which indicates that expressions containing mostly closed-class parts-of-speech (which frequently constitute syntactic combinations according to our typology) should be described in more detail in the guidelines, with additional negative examples. Before manually annotating additional sentences in the corpus, a more targeted approach could be taken by extracting n-grams with problematic closed-class structures and creating a list of all syntactic combinations discovered this way (e.g. two-part connectors such as ne samo A, temveč tudi B ‘not only A, but also B’).

Table 5, on the other hand, shows the part-of-speech structures that were more likely annotated by multiple annotators (3 or more). The most frequent structure, VERB ADP NOUN (e.g. vzeti pod drobnogled, lit. ‘take [sth] under the microscope’, ‘to take under scrutiny’), was frequently and consistently annotated because it contains verbal MWEs previously annotated with PARSEME categories.
The two most frequently annotated structures in general (ADJ NOUN and ADP NOUN) appear almost equally frequently in both the single annotations as well as multiple annotations. This is to be expected, as the difference between a MWE and, for instance, a collocation or a terminological candidate, is a question of semantic interpretation, particularly in the context of the guidelines used for this annotation task, which relied heavily on the annotator’s interpretation on whether an annotated span would require a semantic or encyclopedic explanation in a (general) dictionary language resource.

### 4.2. Annotation Scope and Overlap

In this section, we analyze the degree to which the annotators agreed on the scope of the annotation of individual MWE candidates. Out of the 8,864 annotated candidates, 5,023 (56.67%) were annotated by a single annotator, while 3,841 (43.33%) were assigned multiple annotations. Out of these 3,841 candidates, 2,961 (77.10%) exhibited complete overlap, meaning that all the annotators annotated the exact same elements in each case. The vast discrepancy between single annotations and the percentage of candidates with complete overlap indicates that while there is disagreement on whether a span is a MWE, in the majority of examples where a span is identified as a MWE by multiple annotators, they tend to agree on the elements included. Only 880 examples showed disagreement in annotation scope. For each candidate with incomplete overlap, we first aggregated all the annotated elements and identified the ones that differed between the annotations. In the example below, the MWE candidate was independently annotated four times (Prav tako, tako kakor, Prav tako, Prav tako kakor). Only the element tako (ADV) appears in all annotations, while prav (PART) and kakor (SCONJ) do not, so they are treated as differing elements.

\[\text{si Prav tako jasen kakor prejšnji, bilo je le nekoliko hladnejše.}\]

\[\text{en Just as clear as the day before; it was only somewhat colder.}\]

Table 6 shows the distribution of differing elements by part-of-speech. While adjectives and nouns are at the top of the list, prepositions (ADP), determiners (DET), pronouns (PRON), particles...
The examples in which an adjective was the contested element reveal some interesting insights: the ADJ ADJ NOUN - ADJ NOUN dilemma raises the issue of annotating potential nested MWEs (varuh človekovih pravic, ‘human rights ombudsman’ vs. človekove pravice, ‘human rights’), as well as the issue of optional vs. obligatory elements in MWEs (e.g. človeške pravice, ‘human rights’, vs. temeljne človeške pravice, ‘fundamental human rights’). This is similar to ADP ADJ NOUN - ADP NOUN (po ocenah, ‘according to estimates’ vs. po prvih ocenah, ‘according to the first estimates’). While the guidelines provided instructions on how to treat some of the optional elements, they were mainly focused on the inclusion of verbs in examples such as pisati na roko, ‘to write by hand’). As a general rule, however, each example was to be annotated individually based on how typical the syntactic environment of the identified MWE was, along with the relevant lexical elements. For further annotation, the treatment of these elements should be further specified in order to avoid disagreement.

When nouns are the differing element, the examples again show some discrepancy when it comes to potential nested MWEs (e.g. ADJ NOUN - NOUN ADJ NOUN; ponudniki mobilnih signalov, ‘mobile signal providers’ vs. mobilni signal, ‘mobile signal’; šef obveščevalne službe, ‘secret service director’ vs. obveščevalna služba, ‘secret service’; or ADJ NOUN - NOUN ADP ADJ NOUN; rak na materničnem vratu, lit. ‘cancer on the uteral neck’, ‘cervical cancer’ vs. maternični vrat, ‘cervix’). The current annotation task did not include the annotation of nested MWEs, but the results show that the guidelines should be extended to address this topic and provide clearer instructions (either by allowing for nested annotations or by listing principles on how to determine the optimal scope of the MWE).

The examples with verbs as the differing element seem to indicate that the pool of available lexical candidates that can be substituted within a MWE affects the annotator’s scope. For instance, the structure pair ADP NOUN - VERB ADP NOUN contains both the verbless na voljo, ‘at [someone’s] disposal’ as well as imeti na voljo, at [someone’s] disposal, dati na voljo, ‘to put at [someone’s] disposal’.

<table>
<thead>
<tr>
<th>UPOS</th>
<th>Nr.</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJ</td>
<td>227</td>
<td>16.85%</td>
</tr>
<tr>
<td>NOUN</td>
<td>210</td>
<td>15.59%</td>
</tr>
<tr>
<td>ADP</td>
<td>172</td>
<td>12.77%</td>
</tr>
<tr>
<td>VERB</td>
<td>163</td>
<td>12.10%</td>
</tr>
<tr>
<td>DET</td>
<td>116</td>
<td>8.61%</td>
</tr>
<tr>
<td>AUX</td>
<td>116</td>
<td>8.61%</td>
</tr>
<tr>
<td>PRON</td>
<td>73</td>
<td>5.42%</td>
</tr>
<tr>
<td>PART</td>
<td>72</td>
<td>5.35%</td>
</tr>
<tr>
<td>ADV</td>
<td>62</td>
<td>4.60%</td>
</tr>
<tr>
<td>CCONJ</td>
<td>57</td>
<td>4.23%</td>
</tr>
<tr>
<td>SCONJ</td>
<td>56</td>
<td>4.16%</td>
</tr>
<tr>
<td>NUM</td>
<td>18</td>
<td>1.34%</td>
</tr>
<tr>
<td>PROP</td>
<td>5</td>
<td>0.37%</td>
</tr>
</tbody>
</table>

Table 6: Frequencies and percentages of parts of speech causing disagreement in MWE scope annotation.

<table>
<thead>
<tr>
<th>Diff. Str. Pair</th>
<th>Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJ ADJ NOUN - ADJ NOUN</td>
<td>208</td>
</tr>
<tr>
<td>ADJ ADP ADJ NOUN - ADP NOUN</td>
<td>65</td>
</tr>
<tr>
<td>NOUN ADJ NOUN - NOUN ADJ NOUN</td>
<td>79</td>
</tr>
<tr>
<td>NOUN ADJ NOUN - NOUN ADP ADJ NOUN</td>
<td>23</td>
</tr>
<tr>
<td>VERB ADP NOUN - VERB ADP NOUN</td>
<td>90</td>
</tr>
<tr>
<td>ADP ADJ NOUN - ADP NOUN</td>
<td>62</td>
</tr>
<tr>
<td>ADP ADJ NOUN - ADP ADJ NOUN</td>
<td>41</td>
</tr>
<tr>
<td>AUX AUX VERB ADP NOUN - VERB ADP NOUN</td>
<td>24</td>
</tr>
<tr>
<td>AUX AUX VERB NOUN - VERB NOUN</td>
<td>20</td>
</tr>
<tr>
<td>DET ADP DET NOUN - ADP NOUN</td>
<td>91</td>
</tr>
<tr>
<td>PART ADP NOUN - PART ADP NOUN</td>
<td>19</td>
</tr>
<tr>
<td>CCONJ ADP DET - ADP DET CCONJ</td>
<td>35</td>
</tr>
<tr>
<td>NUM ADP NOUN - ADP NUM NOUN</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 7: Most frequent co-occurring structures within annotations with incomplete overlap. The first column denotes the differing element, the second the structure pair, and the third the frequency of co-occurrence.
posal', \textit{biti na voljo}, 'to be at [someone's] disposal'. The relatively low number of verbs that can be used with \textit{na voljo} seemed to prompt most, but not all of the annotators to include the verb, while others left it out.

Prepositions were frequently contested when in combination with a nominal phrase, e.g. ADJ NOUN - ADP NOUN (v smislu \textit{in [the] sense} vs. formalnem smislu, 'formal sense'; \textit{v leith}, \textit{in [the] years}' vs. \textit{zadnjih leith}, 'last years') or ADJ NOUN - ADP ADJ NOUN (\textit{na} delovnem mestu 'in the workplace', \textit{v zrelih leith}, lit. \textit{in mature years}, 'at an older age'). Annotators were instructed to consult Slovene corpora to determine the most frequent scope of annotation, but while some interpreted the preposition as an obligatory element, others left it out based on their interpretation, e.g. whether the adjective in the MWE can be considered an open slot (v \textit{zadnjih/prejšnjih/naslednjih} leith, 'in the [last/previous/next] years'); similar to numerals in ADP NOUN - ADP NUM NOUN: \textit{pred [desetimih] leti}, '[ten] years ago'; or determiners in ADP DET NOUN - ADP NOUN: \textit{če [nekaj] dni}, 'in [a few] days') or whether the nominal phrase occurs frequently enough by itself (\textit{delovno mesto}, 'workplace').

There is also some disagreement with regard to the inclusion of auxiliary verbs in verbal MWEs, e.g. AUX VERB ADP NOUN - VERB ADP NOUN (\textit{jjje} vzel pod drobnogled, 'did take under scrutiny') and AUX VERB NOUN - VERB NOUN (\textit{ni} odprl ust, lit. \textit{he didn't open [his] mouth}, 'remained silent'), particularly when there is a negation, but both the negated and non-negated versions are viable (\textit{je odprl usta}, 'he spoke', \textit{ni odprl ust}, 'he remained silent').

### 4.3. Overlap with Named Entities

Because the SUK 1.0 corpus was also independently annotated with named entities, we analyzed our MWE annotations in terms of tokens that have been annotated as named entities in order to explore any potential legitimate overlaps. Only 334 (3.77\%) candidates contain at least one token that has also been annotated as a named entity, and only 115 were annotated by multiple annotators. By analyzing the distribution of the named entity annotations within these 115 candidates, we see that the majority were annotated as organizations (48\%) or have no annotation (39\%; meaning that not all the MWE elements overlap with the named entity), while other NE categories account for much smaller percentages: miscellaneous (10\%), location (2\%), person (1\%), and person-derivative (0.5\%). The guidelines mention that generic titles of institutions, documents, etc. should be annotated as MWEs, particularly if they indicate culturally specific expressions with no direct equivalents or transparent translations in other languages.

A closer look at the examples shows that in the majority of cases, the MWE annotations are nested within NE annotations (e.g. [\textit{Ustavno sodišče Slovenije}, 'the [Constitutional Court] of Slovenia'; \textit{Urad za [narodnostne manjšine], 'Office of [National Minorities]'}, but the opposite also occurs, with NES included in MWEs (\textit{na sončni strani [Alp]}, lit. \textit{on the sunny side of [the Alps]}, \textit{in Slovenia}; \textit{kod gre na [Dunaj], naj pusti trebih zunaj}, lit. \textit{whoever goes to Vienna should leave their stomach outside}, 'Vienna is very expensive' or 'large cities are very expensive') or appearing in open slots of MWEs (\textit{so voda na [Lutov] mlin}, lit. \textit{they are water to [Lut's] mill}, 'they provide an advantage to him'). These examples are useful to include in the improved guidelines to exemplify the potential overlaps between MWEs and NEs and to provide clearer instructions on how to annotate mixed candidates.

### 5. Conclusion

In the paper, we presented the results of the first step of the process of comprehensive MWE annotation in the SUK 1.0 corpus, and conducted a number of quantitative analyses to pinpoint potential weak points in the first version of our annotation guidelines. In particular, the process shows that more instructions and examples are required on how to differentiate between terminological candidates and collocations on one hand, and MWEs on the other. Although the annotators seem to achieve a considerable degree of overlap in terms of annotation scope, for some structures, the scope should be more precisely defined (e.g. the inclusion of auxiliary verbs and closed-class parts-of-speech such as prepositions). In addition, closed-class part-of-speech structures can be pre-extracted in order to generate a list of valid candidates as a reference point for annotators and, potentially, for pre-annotating some of the more trivial syntactic combinations. Pre-annotation with a list of all other MWE-candidates is also an option, but might be more difficult to implement for Slovene, which features a flexible word order and is a morphologically rich language.

Although there has not been much overlap between MWEs and NES in the annotated examples, the ones that do occur nevertheless show the need for more specific guidelines on when to treat candidates as named entities and how to treat borderline examples (e.g. when the lack of capitalization makes it unclear whether the span denotes a named entity or a generic concept) and mixed candidates (nested MWEs within NES or vice versa).

In our future work, we intend to use the UniDive MWE annotation guidelines to perform a second step annotation of the identified MWE candidates.
and determine their categories so that they can be added to the SUK 1.0 corpus alongside their PARSEME verbal MWE equivalents. Once the final annotations have been added to the corpus, a second analysis of outlying examples (either those left unannotated by the majority of annotators or those consistently annotated but not considered MWEs in the final version) can provide additional insight for further MWE identification. In addition, the annotated POS-structures can potentially be compared to the total frequencies of POS-structures within the corpus in order to pinpoint whether certain structures are more typical of MWEs in Slovene in general. Additional statistical analyses on MWE patterns can also be performed by taking into account other annotation layers present in the corpus, such as semantic role labeling and UD dependency relations.

6. Acknowledgements

The study presented in this paper was conducted within the New Grammar of Modern Standard Slovene: Resource and Methods project (J6-8256), which was financially supported by the Slovenian Research and Innovation Agency (ARIS) between 2017 and 2020. The authors also acknowledge the financial support from the Slovenian Research and Innovation Agency (research core funding No. P6-0411 - Language Resources and Technologies for Slovene and No. P6-0215 - Slovene Language – Basic, Contrastive, and Applied Studies).

The authors would like to thank the anonymous reviewers for their valuable insight, and all the annotators who participated in the project: Anna Maria Grego, Tjaša Šoltes, Tajda Liplin Šerbetar, Pia Rednak, Jana Vaupotič, Zala Vidic, Karolina Zgaga, and Kaja Gantar.

7. Ethical Considerations and Limitations

It should be noted that 80% of the people who performed the annotation were university-level students of linguistics, and while they were familiarized with the guidelines and their performance was tested and compared to the performance of experts and considered to be satisfactory in the majority of cases, the annotations need to be interpreted with their background in mind.

In addition, the SUK 1.0 corpus mostly contains written standard Slovene, so the results cannot necessarily be extrapolated to e.g. spoken or non-standard Slovene.

8. Bibliographical References


Light Verb Constructions in Universal Dependencies for South Asian Languages

Abishek Stephen, Daniel Zeman
Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics
Malostranské náměstí 25, CZ-11800 Praha, Czechia
{stephen, zeman}@ufal.mff.cuni.cz

Abstract
We conduct a morphosyntactic investigation into the light verb constructions (LVCs) or the verbo-nominal predicates in South Asian languages. This work spans the Indo-Aryan and Dravidian language families in treebanks based on Universal Dependencies (UD). For the selected languages we show how well the existing annotation guidelines fare for the LVCs. We also reiterate the importance of the core and oblique distinction in UD and its usefulness for making accurate morphosyntactic annotation judgments for such predicates.

Keywords: light verbs, universal dependencies, multiword expressions

1. Introduction
Universal Dependencies (UD) (de Marneffe et al., 2021) presents a morphosyntactically oriented approach to perform linguistic annotations anchored on binary dependency relations between intra-sentential units. These dependency relations hold primarily between content words, while function words are seen as carriers of morphosyntactic features, which typically “belong” to a content word. Such a mechanism is followed in UD to increase the typological parallelism between languages.¹

The selection of the dependency head gets a little complicated in the case of a multiword expression (MWE) where two or more words combine into a single lexical unit with or without morphosyntactic implications (Masini, 2019). One of the MWE classes where this can be witnessed is the light verb construction (LVC).

LVCs (Section 3) have a peculiar semantic composition that may provoke specific approaches to their syntactic analysis; however, in the case of South Asian languages, profound morphosyntactic clues are available and should be taken into account. The current annotations in the treebanks of these languages in UD treat the LVCs² as combinations of lexemes that morphosyntactically behave as single words and mark them using the dependency relation compound,³ or its subtype compound:lvc. In the case of South Asian languages this is problematic given the surface-identical noun incorporations and object-verb sequences. We illustrate it on two examples from the treebanks of Hindi (Figures 1 and 2) and Telugu (Figures 3 and 4). In each pair, the first example has a LVC annotated as compound while the second example with a similar construction treats the noun as an object (obj) of the verb. Our main research question is whether these distinctions are well-motivated and clearly defined based on morphosyntax. It implies some broader questions about argument selection criteria and core vs. oblique distinction in South Asian languages.

¹https://universaldependencies.org/u/overview/syntax.html
²For our study we consider all the noun-verb sequences marked as compound or compound:lvc in the treebanks as LVCs or verbo-nominal predicates.
³https://universaldependencies.org/u/dep/compound.html

Figure 1: A verbo-nominal construction in Hindi (HDTB) annotated as compound.

Figure 2: A verbo-nominal construction in Hindi (HDTB) annotated as object.
Figure 3: A verbo-nominal construction in Telugu (MTG) annotated as compound.

Figure 4: A verbo-nominal construction in Telugu (MTG) annotated as object.

Hence, using the treebanks of Indo-Aryan and Dravidian languages (Table 1) from UD 2.13 (Zeman et al., 2023), we intend to bring to light the fundamental issues around the treatment of various noun-verb sequences. We illustrate that not all noun-verb sequences qualify to be marked as compound or compound:lvc. We will focus on how the morphosyntactic implications have been overlooked by illustrating supporting examples for the same. Furthermore, we also emphasize the essential distinction between core and oblique arguments in UD (Zeman, 2017) that encompass a crucial role in the morphosyntactic treatment of the noun-verb sequences.

The paper is organized into 6 sections. Discussion of related works happens in Section 2. In Section 3, we present a portrait of LVCs in the selected UD treebanks, organized by language families. In Section 4, we discuss the structural composition of the LVCs by differentiating between incorporation and compounding. In Section 5, the morphosyntax of LVCs finds adequate theoretical treatment, confronted with treebank practice in Section 6.

2. Related Work

Kahane et al. (2018) discusses how to analyze multiword expressions in treebanks based on UD. They mainly focus on distinguishing syntactically irregular MWEs from semantically non-compositional ones and highlight issues related to intra-treebank annotation inconsistencies created because of the MWEs. The analysis concerns the English and French treebanks in UD 2.1 and they note inter-corpus variation in the usage of the dependency relation compound. But the LVCs did not receive any attention.

Nivre and Vincze (2015) portrays how LVCs pose interesting challenges for linguistic annotation, especially from a cross-linguistic perspective. They present a survey of the different ways in which LVCs are analyzed in UD 1.1. They group the languages into 3 groups and compare how the LVCs consisting of a transitive verb and a direct object are handled. For example, they report that in the English phrase take a photo, photo is attached to the verb take as a direct object (dobj) because the English treebanks in version 1.1 did not distinguish LVCs whereas the treebanks of Swedish, German, and Irish distinguish LVCs through their syntactic structure.

Since our study takes into consideration the constructions labeled as compound or compound:lvc it is worthwhile to mention that in the Persian treebank (Seraji et al., 2016) the non-canonical subjects are analyzed with respect to LVCs and such constructions are labelled as compound:lvc. In the case of the Hungarian treebank (Vincze et al., 2017), the label dobj:lvc can be found between the nominal and verbal component of the LVCs, where the dobj part of the label marks that syntactically it is a verb–object relation but semantically, it is an LVC, marked by the lvc subtype.

Among the South Asian languages, Hindi has received a considerable spotlight for LVCs. Palmer et al. (2009) talks about the LVCs as support-verb

Table 1: Treebank sizes in UD 2.13.

<table>
<thead>
<tr>
<th>Language</th>
<th>Treebank</th>
<th>Sentences</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sanskrit</td>
<td>Vedic</td>
<td>3,997</td>
<td>27,117</td>
</tr>
<tr>
<td>Sanskrit</td>
<td>UFAL</td>
<td>230</td>
<td>1,843</td>
</tr>
<tr>
<td>Hindi</td>
<td>HDTB</td>
<td>16,649</td>
<td>351,704</td>
</tr>
<tr>
<td>Hindi</td>
<td>PUD</td>
<td>1,000</td>
<td>23,829</td>
</tr>
<tr>
<td>Urdu</td>
<td>UDTB</td>
<td>5,130</td>
<td>138,077</td>
</tr>
<tr>
<td>Kangri</td>
<td>KDTB</td>
<td>288</td>
<td>2,514</td>
</tr>
<tr>
<td>Bhojpuri</td>
<td>BHTB</td>
<td>357</td>
<td>6,665</td>
</tr>
<tr>
<td>Bengali</td>
<td>BRU</td>
<td>56</td>
<td>320</td>
</tr>
<tr>
<td>Marathi</td>
<td>UFAL</td>
<td>466</td>
<td>3,847</td>
</tr>
<tr>
<td>Sinhala</td>
<td>STB</td>
<td>100</td>
<td>880</td>
</tr>
<tr>
<td>Telugu</td>
<td>MTG</td>
<td>1,328</td>
<td>6,465</td>
</tr>
<tr>
<td>Tamil</td>
<td>TTB</td>
<td>600</td>
<td>9,581</td>
</tr>
<tr>
<td>Tamil</td>
<td>MWTT</td>
<td>534</td>
<td>2,584</td>
</tr>
<tr>
<td>Malayalam</td>
<td>UFAL</td>
<td>218</td>
<td>2,403</td>
</tr>
</tbody>
</table>

*Under UD v2 guidelines this relation is renamed to obj:lvc. Besides Hungarian, it is now used also in French and Naija.
constructions in Hindi-Urdu where eventive noun phrases combine with several verbs and are analyzed based on case marking. The analysis relies on the Proposition Bank (Palmer et al., 2005) scheme. Begum et al. (2011) focus on the identification of the noun-verb combinations based on the Hindi Dependency Treebank (HDTB). Müller (2019) shows an HPSG analysis and Vaidya et al. (2014) present a TAG (Joshi, 2005) analysis for predicates with the light verbs karanā ‘to do’ and honā ‘to be’ in Hindi, demonstrating that LVCs are a highly productive predicational strategy, challenging for computational grammars.

The PARSEME (Savary et al., 2023) multilingual annotated corpus of verbal multiword expressions also includes Hindi. The underlying hypothesis for the annotations is that verbal MWEs have some degree of semantic non-compositionality and the verb is considered to be the syntactic head. Within the UD framework, typological studies around LVCs have not involved any of the South Asian languages so far.

3. Light Verb Constructions in UD

The LVCs belong to the class of complex predicates with a wide range of combinatorial potential where a verb (VERB) can combine with adjectives (ADJ), adverbs (ADV) or nouns (NOUN). Out of these, we focus on the verbo-nominal predicates comprising words with the part-of-speech tags NOUN and VERB. This subgroup is most similar to (and confusable with) object-verb sequences; it also has interesting morphosyntactic properties.

3.1. Indo-Aryan Languages

The Indo-Aryan languages are characterized by split ergativity, subject-object agreement, canonical SOV word order, and the presence of prenominal case marking. UD annotation guidelines

---

6 https://ltrc.iit.ac.in/treebank_H2014/
7 https://gitlab.com/parseme/parseme_corpus_hi

---
Dravidian treebanks is that the distinction between LVCs and regular structures has largely relied on semantic cues or direct influence of the strategy used in the English UD treebanks. Intra-language morphosyntactic clues do not seem to have been considered.

4. Structural Composition of LVCs

According to Butt (2003), the “light” in LVCs indicates that although these constructions respect the standard verb complement schema, the verb cannot be said to be predicating fully but seems to be more of a verbal licensor for nouns. Moreover, the light verbs tend to have a “funny” syntax which distinguishes them from auxiliaries and main verbs. Additionally, Butt (2003) claims that such structures are monoclausal in nature where the predicational elements “co-predicate”. Such a view does not align well with saying that they form one lexical unit, but using the compound relation in UD can be understood as saying exactly that. There seems to be a perturbing dichotomy around the lexicality of such sequences as shown in Figure 9, where two instances are analyzed as compounds and one is not. In order to establish a principled position on the structural composition of LVCs, we will now delve into the process of compounding and incorporation and discuss their entanglement with the predicate structure.

4.1. Compounding

We adopt the definition of compounds based on Haspelmath (2023b) as a construction consisting of two strictly adjacent slots for roots⁸ that cannot be expanded by full nominal, adjectival, or degree modifiers. Finkbeiner and Schlücker (2019) illustrate the non-expandability on a German example, where the adverb sehr ‘very’ cannot modify the first element in Alt-bau ‘old building’, i.e., ‘sehr Alt-bau very old building’ is not plausible.

On applying Haspelmath’s definition to Figure 9, we observe that the noun part of the compound suru kara ‘to start’ is a root morph whereas the other nouns goli ‘bullet’ and cunau ‘challenge’ are derived nominal forms of their respective root morphs. If we assume this inference to be accurate, then cunau dena ‘to challenge’ and goli calanà ‘to shoot’ should not be marked as compound. Hence if a noun-verb sequence shall be considered a compound, the nominal part should be a root without suffixes.

⁸Sanskrit UFAL uses the feature Compound=Yes to mark words that were non-final stems within a surface “compound”; however, such forms are treated as separate syntactic words only if the dependency relations between them are other than compound.⁹

⁹A root is a contentful morph (i.e., a morph denoting an action, an object, or a property) that can occur as part of a free form without another contentful morph (Haspelmath, 2023b).
Iftar Sangam was inaugurated by KKM President Ibrahim Kunnil.

When challenged by the army, the terrorists started firing.

The UD taxonomy has a more relaxed definition of compounds: it states that the compound relation should be used for combinations of lexemes that morphosyntactically behave as single words, and lexicalization or semantic idiomaticity should not be a criterion for identifying compounds. This entails that a lexicalized expression like make a decision in English does not qualify as an MWE or a compound in UD. Expressions that would qualify should have a single argument structure or in other words, the syntactic head of an LVC should select all the required arguments and the dependent noun should neither be modified nor have an argument structure of its own. But in the case of the Indo-Aryan languages, this does not seem to be the case.

In Marathi (Figure 10) the LVC prayatna karata ‘trying’ is tagged as compound:lvc where the noun prayatna ‘try’ heads the nsubj and xcomp dependency relations which is not consistent with the UD guidelines. For once we could assume it to be a language-specific decision but there are also examples like Figure 11 which say otherwise. In both the examples (Figure 10 and 11) the compound:lvc relation is headed by the verb karane ‘to do’ but the dependent nouns are different. This leads a UD user to the conclusion that in such predicates the nouns have arbitrarily chosen argument structure as no morphosyntactic motivations can be seen in the surface syntactic structure. Similar inconsistencies can also be found in other Indo-Aryan languages. This inconsistent behavior suggests that the annotation choices made for the LVCs are not strongly based on a concrete morphosyntactic mechanism.

Among Dravidian languages, Tamil and Malayalam have taken a left-headed approach considering the noun as the head whereas Telugu treats the verb as the syntactic head making the compound:lvc relation right-headed. The annotation of the LVCs is comparatively more consistent than in the Indo-Aryan languages but it seems to be heavily influenced by semantics or by the treatment of LVCs in the English treebanks. For example, the current version of the Malayalam UFAL treebank uses the compound:lvc relation for noun-verb and verb-verb sequences where the do-verb ceyyuka appears. No morphosyntactic motivation can be found in the respective documentation pages of the Dravidian languages.

We conclude that if a noun-verb construction is marked as compound(lvc), the syntactic head is eligible for modifications but not the dependent. If we need to annotate a child of the dependent node in the noun-verb sequence, then the sequence should be treated as verb with object.
4.2. Noun Incorporation

It is also worthwhile to mention the broader typological definition of incorporation by Haspelmath (2023a) according to which an incorporation is an event-denoting noun-verb compound construction in which the noun occupies an argument slot of the verb and occurs in a position where nominal patient arguments cannot occur. In most Indo-Aryan languages, verbo-nominal predicates must be analyzed as a lexical category but paradoxically enough, the noun is on par with a syntactically independent argument (Mohanan, 1995). Therefore, even though noun incorporation is a type of compounding of a syntactic object with the verb, both the object and the verb can have their own argument structures. It may thus be hard to find incorporation that satisfies Haspelmath’s definition in South Asian languages. Currently, the UD taxonomy has no special provisions to define incorporation and they are treated as compounds. As a result, there are no distinct annotations for an object-verb pair and a ‘conjunct verb’. The Hindi HDTB treebank in UD is converted from the Paninian Dependencies and in that scheme, conjunct verbs have a special tag \textit{pof} (Tandon et al., 2016). It does not denote a dependency but rather represents the fact that the noun-verb sequence is an MWE. The logic behind the usage of the \textit{pof} tag is based on the semantic coherence of the noun-verb sequence being a single predicative element although some morphosyntactic cues do come in handy (discussed in Section 5). Tandon et al. (2016) also acknowledges that the identification of conjunct verbs is problematic as it appears to be an issue for the syntax-semantics interface and the decision was left to the annotators at the cost of inconsistencies in the data. On conversion from the Paninian dependencies to UD all the \textit{pof} relations were automatically changed to compound and the inconsistencies persist. This brings us to a juncture where distinguishing object-verb sequences from noun incorporation becomes necessary. For Dravidian languages, Sudharsan (1998) states that if the noun in a noun-verb sequence cannot be inflected for case or number and even cannot be modified by an adjective then it is the case of a noun incorporated into the verb. Since incorporated nouns do not take case or plural markers and external modifiers, they are morphosyntactically different from the regular object nouns. Similarly for Indo-Aryan languages or more specifically for Hindi-Urdu, Mohanan (2017) has also recommended very similar criteria for distinguishing objects and incorporated nouns. These criteria treat noun incorporation as a type of compounding but there are also cases where such syntactic tests are inadequate, for example in cases of independent syntactic argument structures. The nominal part can be a noun or a root morph. Usually, the root morphs do not have an argument structure of their own but a noun on the other hand has the potential to have its own argument structure in such noun-verb constructions (Mohanan, 1995). To qualify for a \textit{compound:lvc} relation the noun-verb sequence should have a single argument structure but that is not always true in case of noun incorporations. This indicates a need for a distinction between compounding and noun incorporation. In the following section, we find taxonomical differences between them but it will be also worthwhile to test how similar their morphosyntax is and how we can distinguish them from object-verb sequences.

5. Morphosyntax of LVCs

Subjects and objects in UD must satisfy the condition of being core arguments, which means that they should receive the language-specific coding and treatment associated with the grammatical functions S, A, and P (Zeman, 2017; Andrews, 2007). This coding derives from primary transitive predicates and may include various strategies,
including case marking on nouns and agreement morphology on verbs. Nominals whose grammatical function is A or S are called subjects and their dependency relation to the verb is nsubj whereas the nominals whose grammatical function is P are called (direct) objects and their dependency relation to the verb is obj (Zeman, 2017). Turning back to Haspelmath’s definition of noun incorporation in Section 4, the incorporated noun cannot occupy the patient position and cannot have the function P. Hence, we illustrate the behavior of LVCs through morphosyntactic processes like verbal agreement, case marking, and nominal modification. This analysis will bring out the distinctions between compounds and object-verb sequences.

5.1. Case Marking

Hindi, Urdu, and some other Indo-Aryan languages follow a split-ergative pattern. Perfective clauses have the ergative alignment, imperfective clauses have a nominative-accusative alignment. In the latter, the subject is in the bare nominative form (without adpositions), while animate direct objects use the postposition ko. Inanimate direct objects may omit the postposition ko; if they use it, the object is understood as definite. The accusative (oblique) case is used with the postposition, but without it, the object stays in nominative. Indirect objects always use the postposition ko. In transitive perfective clauses, the subject takes the ergative postposition ne.

Nominal parts of LVC candidates are inanimate and thus harder to distinguish from direct objects. However, the ability to take the optional ko signals that the noun is an object.

A few true LVCs, such as śūrū karāṇa ‘to start’, can be transitive as a whole. Here, śūrū is not an object and the whole compound may take a real object (which follows the above criteria for objects) or a complement clause. In most cases, however, the nominal part of the LVC is a direct object, and if the whole LVC is semantically transitive, then the external “object” is coded as a nominal modifier (with the genitive postposition kā) of the noun in the LVC. It should then be annotated as nmod in UD (pula kā nirmāṇa ‘construction of bridge’ in Figure 1). Even with śūrū karāṇa the genitive strategy is a possible alternative and occurred twice in HDTB. The predating nominals in Hindi may also select arguments with other postpositions, such as par ‘on’, se ‘from’, or ko ‘to’ (Vaidya et al., 2016).

Eastern Indo-Aryan languages such as Bhojpuri do not have the ergative alignment in perfective clauses. Similarly to Hindi, animacy and definiteness play a role in marking of the direct object (Thakur, 2021). However, Bhojpuri uses the same postposition (ke) (Figure 14) for accusative, dative, and genitive, making it less obvious when it is selected by the nominal and not the verb.

In Dravidian languages too the arguments are postpositionally case-marked but in an agglutinative manner. In Tamil MWTT, we find examples like kumār muncūkku vāntāy ‘Kumar progressed (in his career/ life)’ where the nominal component muncūkku ‘to the front’ of the compound:lvc is assigned the dative case and the subject proper noun Kumar takes the nominative case. Since muncūkku is treated as the root, the analysis gets blurry but muncūkku vā ‘to progress’ might not qualify to be considered as a compound due to the dative case marking.

The presence of an adpositional phrase selected by the nominal differentiates compounding
from noun incorporation but this does not provide a suitable distinction between object-verb sequences and noun incorporations at least for the Indo-Aryan languages. In this light, we observe that currently most of the compound:lvc or compound relations describing noun-verb sequences are not true compounds as the nominal participant does show case marking.

5.2. Agreement

The split-ergative pattern in some Indo-Aryan languages allows for testing of object-verb agreement. In imperfective clauses, the gender and number of the subject are cross-referenced by the verb’s morphology. In transitive perfective clauses, the ergative postposition ne blocks agreement with the subject; but unless the direct object is marked with ko, verbal morphology cross-references the gender and number of the object (rather than subject). If the postposition ko is present, the verb takes the default masculine singular form.11

Agreement with the verb in transitive-perfective clauses is another signal that the nominal of an LVC candidate is an object rather than part of a compound. And it can also attest to the opposite: In mere pitā ne pūjā surū kār di hai ‘my father has started the prayer’, the verb has a feminine form, agreeing with pūjā, while both pitā ‘father’ and surū ‘start’ are masculine.

Eastern Indo-Aryan languages (e.g., Bhojpuri and Bengali), as well as Dravidian languages, follow the nominative-accusative pattern with subject-predicate agreement and no ergativity (Krishnamurti, 2003). In Telugu, the verb agrees with the subject when it is in the nominative case, whereas when there is a dative “subject”, the verb agrees with the incorporated noun (Nadimpalli and Lakshm, 2022). Similar observations can be made for other Dravidian languages except for Malayalam where subject-verb agreement is absent.

To conclude this section, in many instances of noun-verb sequences agreement between the noun and the verb is observed and represents a deviation from typical compound behavior.

5.3. Modification

One of the signs of compounds is that their parts (and especially the dependent part) cannot be modified individually. We have seen that the patient in Hindi LVC candidates is often encoded as a modifier of the predicative nominal, which speaks against a noun-verb compound analysis. Similarly,

11While in general postpositions block agreement in Indo-Aryan languages, Gujarati is an exception where verb agreement works despite postpositions (Subbarao, 2012, p. 97).

Figure 15: Compound analysis in Kangri (KDTB).

in Kangri in Figure 15, the nominal galla ‘matter’ is modified by the determiner isadi ‘this’, suggesting that galla mannī is not a compound.

In Telugu too, we find similar instances of the predicative nominal modification. For example, in vāḍu cālā takkuva pani cēsēḍu ‘He does very little work’, takkuva ‘less’ modifies pani ‘work’ which happens to be in a compound:lvc relation with cēsēḍu ‘do’.

5.4. Word Order

Real compounds would not allow intervening words between the noun and the verb (at least not by Haspelmath’s definition of compounds). An intervention seems to be always possible at least by the negative particle: unhornīne batāyā ki abhī pahale baica kā praśikṣaṇa surū nahīṁ huā hai. ‘He told that the training of the first batch has not started yet.’

5.5. Transitivity

The grammars of Indo-Aryan languages feature a systematic opposition of transitive (causative) and intransitive verbs. The intransitive counterpart of karanā in Hindi is honā ‘to be, become, happen’; as shown in Section 3, its cognates do the same job in the other languages. Whenever it is inappropriate to analyze X karanā as a compound, the same can be said about X honā. However, as honā is intransitive, X can hardly act as its object. In Hindi-Urdu this verb is also used as the copula, hence a copular analysis may be an alternative. Where the light verb cannot be a copula, we should probably go with secondary predication (xcomp).

6. LVCs in UD Revisited

Noun-verb compounds are very frequent in the current UD treebanks of South Asian languages. In Hindi HDTB, there are 6187 such compounds with the 5 most common verbs alone (out of which 4159 occurrences belong just to karanā ‘to do’). A similar pattern is found in the smaller Urdu treebank:
3542 occurrences with the top 5 verbs, including 2346 with krnā ‘to do’. The remaining treebanks are an order of magnitude smaller, yet we find 58 different compounds in Bhojpuri and 31 in Hindi PUD occurring twice or more. Nevertheless, the treebanks are not always consistent and it is not uncommon to see the same noun-verb combination annotated sometimes as a compound and sometimes as an object.

For example, Hindi bāta karanā ‘to talk’ is a relatively frequent expression and it is usually annotated as compound (118 instances), though occasionally it is annotated as obj (25 instances). The noun bāta can occur with the postposition ko and then it is always annotated as the object (13 instances). It can occur in the plural (11 instances without ko and 2 instances with ko) and there can occasionally be other constituents between it and the verb. In transitive perfective clauses, the verb agrees with its feminine gender: Natavarā Sinhā (Masc) ne Nirupama Sena se bāta (Fem) ki (Fem) hai ‘Natwar Singh had spoken to Nirupam Sen’. The noun bāta can be also modified by a nominal denoting the matter that is being talked about. All this is evidence that bāta should be syntactically analyzed as the object of karanā. For more statistics across the treebanks, see the Appendix.

Furthermore, based on the arguments present in Section 5, we can conclude that in the present versions of the treebanks of South Asian languages, the treatment of noun-verb sequences or LVCs as compounds is not consistent because the interplay of surface level similarities between real noun-verb compounds and noun incorporations somehow weigh down the morphosyntactic cues. There should not be a problem if noun-verb compounds satisfying the UD guidelines are marked as compound:lvc just to differentiate it from other type of compounds. This would also handle most of the noun incorporations, but once the nominal participant is case marked, modified or triggering verbal agreement, the sequence should be analyzed differently. One of the solutions could be to label the relation obj:lvc, modifying Vincze et al. (2017)’s proposal to fit the current UD version. By doing so, there will be a three-way distinction between noun-verb compounds and noun incorporations (with a single argument structure) marked as compound:lvc, object-verb sequences marked as obj and noun-incorporations with individual noun and verb argument structures as obj:lvc.

7. Conclusion

We have presented morphosyntactic clues for identifying light verb constructions in South Asian languages, which could prove instrumental in achieving consistent annotations of compound and compound:lvc dependency relations. While LVCs as semantically idiosyncratic constructions are widespread in these languages, we have shown that in many cases their syntactic behavior is transparent or very close to standard object-verb constructions. Their compound analysis should be reconsidered and the annotation could be changed to obj or obj:lvc based on the type of argument sharing.

We also touched upon the core vs oblique distinctions and highlighted the phenomenon of noun incorporations, which can be beneficial for tackling similar inconsistencies beyond the languages handled in this study.

8. Acknowledgements

This work was supported by the Grant 20-16819X (LUSyD) of the Czech Science Foundation (GAČR); and LM2023062 (LINGAT/CLARIAH-CZ) of the Ministry of Education, Youth, and Sports of the Czech Republic; and the Charles University project GA UK No. 101924; and partially supported by SVV project number 260 698.

9. Bibliographical References


Rita Finkbeiner and Barbara Schlücker. 2019. Compounds and multi-word expressions in the languages of Europe, pages 1–44. De Gruyter.

Martin Haspelmath. 2023a. Compound and incorporation constructions as combinations of unexpandable roots.


### 10. Language Resource References

Zeman, Daniel and Nivre, Joakim and Abrams, Mitchell and Ackermann, Elia and Aeppli, Noémi and Aghaie, Hamid and Agić, Željko and Ahmadi, Amir and Ahrenberg, Lars and Ajede, Chika Kennedy and Akkurt, Salih Furkan and Aleksandrvičiūtė, Gabrielė and Alfina, Ika and Algom, Avner and Alnajjar, Khalid and Alzetta, Chiara and Andersen, Erik and Antonsen, Lene and Aoyama, Tatsuya and Aplonova, Katya and Aquino, Angelina and Aragon, Carolina and Aranes, Gyld and Aranzabe, María Jesus and Arican, Bilge Nas and Arnardóttir, Pórunn and Arutlie, Gashaw and Arwidarasti, Jessica Naraiswari and Asahara, Masayuki and Åsgeirsdóttir, Katla and Aslan, Deniz Baran and Asmazoğlu, Cengiz and Ateyah, Luma and Atmaca, Furkan and Attia, Mohammed and Atutxa, Alziber and Augustinus, Liesbeth and Aveläs, Mariana and Badmaeva, Elena and Balasubramani, Keerthana and Ballesteros, Miguel and Banerjee, Esha and Bank, Sebastian and Barbú Mîtîelu, Virginica and Barkason, Starkaður and Basile, Rodolfo and Basmov, Victoria and Batchelor, Colin and Bauer, John and Bedir, Seyyit Talha and Behzad, Shabnam and Belieri, Juan and Bengoethea, Kepa and Benll, Ibrahim and Ben Moshe, Yiflat and Berk, Gözde and Bhat, Riyaz Ahmad and Biagetti, Érica and Bick, Eckhard and Bielinskiene, Agnė and Bjarnadóttir, Kristín and Blokland, Rogier and Bobicev, Victoria and Boizou, Loïc and Borges Völker, Emanuel and Börstell, Carl and Bosco, Cristina and Bouma, Gosse and Bowman, Sam and Boyd, Adriane and Braggaa, Anouck and Branco, António and Brokalté, Kristina and Burchardt, Aljoscha and Campos, Marisa and Candito, Marie and Caron, Bernard and Caron, Gauthier and Carvalheiro, Catarina and Carvalho, Rita and Cassidy, Laurent and Castro, Maria Clara and Castro, Sérgio and Cavalcanti, Tatiana and Cebiroğlu Eryigit, Gülşen and Cecchini, Flavio Massimiliano and Celano, Giuseppe G. A. and Chepló, Slavomír and Cesur, Neslihan and Cetin, Savas and Çeținoğlu, Özlem and Chalub, Fabricio and Chamil, Liyanage and Chauhan, Shweta and Chi, Ethan and Chika, Taishi and Cho, Yongseok and Choi, Jinho and Chun, Jayeol and Chung, Juyeon and Cignarella, Alessandra T. and Cinková, Silvie and Collomb, Aurélie and Çöltekin, Çağrı and Connor, Miriam and Corbetta, Claudia and Corbett, Daniela and Costa, Francisco and Courtin, Marine and Crabbé, Benoît and Cristescu, Mihaela and Cvetkoski, Vladimir and Dale, Ingerid Löyning and Daniel, Philemon and Davidson, Elizabeth and de Alencar, Leonel Figueiredo and de Alencar, Leonel and de la Manuel, Rachée and de la Peña, María and de Laurentiis, Martina and de Marneffe, Marie-Catherine and de Paiva, Valeria and Derin, Mehmet Oguz and de Souza, Elvis and Díaz de Ilarraza, Arantza and Dicker, Carla and Dinakaramani, Arawinda and Di Nuovo, Elisa and Dione, Bamba and Dirix, Peter and Dobrovolsj, Kaja and Doyle, Adrian and
Siyao and Peng, Siyao Logan and Pereira, Rita and Pereira, Silvia and Perez, Cenel-Augusto and Perkova, Natalia and Perrier, Guy and Petrov, Slav and Petrova, Daria and Peverelli, Andrea and Phelan, Jason and Pierre-Louis, Claudel and Piltulainen, Jussi and Pinter, Yuval and Pinto, Clara and Pintucci, Rodrigo and Pirinen, Tommi A and Pitler, Emily and Plamada, Magdalena and Plank, Barbara and Poibeau, Thierry and Ponomareva, Larisa and Popel, Martin and Pretkalnina, Lauma and Prevost, Sophie and Prokopidis, Prokopis and Przepliorkowski, Adam and Pugh, Robert and Puolakainen, Tiina and Pyysalo, Sampo and Qi, Peng and Querido, Andreia and Rääbis, Andriela and Rademaker, Alexandre and Rahoman, Mianur and Rama, Tanaka and Ramasamy, Loganathan and Ramisch, Carlos and Ramos, Joana and Rashel, Fatim and Raoosolu, Mohammad Sadegh and Ravishankar, Vinit and Real, Livy and Rebeja, Petru and Reddy, Siva and Regnault, Mathilde and Rehm, Georg and Riabi, Arj and Riabov, Ivan and Rießler, Michael and Rimkutė, Erika and Rinaldi, Larissa and Rituma, Laura and Riziyah, Putri and Rocha, Luisa and Rögnvaldsson, Eirikur and Roksandic, Ivan and Romanenko, Mykhailo and Rosa, Rudolf and Roşca, Valentin and Rotavi, David and Rozonoyer, Ben and Rudina, Olga and Rueter, Jack and Rūnārsdóttir, Kristján and Sadde, Shoval and Safari, Pegah and Sahala, Aleks and Saleh, Shadi and Salomoni, Alessio and Samardžić, Tanja and Samson, Stephanie and Sanguinetti, Manuela and Sanyar, Ezgi and Sárg, Dage and Sartor, Marta and Sasaki, Mitsuya and Saulíte, Baiba and Savary, Agata and Sawanakunanan, Yanin and Saxena, Shefali and Scannell, Kevin and Scarlata, Salvatore and Schang, Emmanuel and Schneider, Nathan and Schuster, Sebastian and Schwartz, Lane and Seddah, Djarné and Seeker, Wolfgang and Seraj, Mojgan and Shahzadi, Syeda and Shen, Mo and Shimada, Atsuko and Shirasu, Hiroyuki and Shishkina, Yana and Shohibussirir, Muh and Shvedova, María and Siewert, Janine and Sigurðsson, Einar Freyr and Silva, João and Silveira, Aline and Silveira, Natalia and Silva, Sara and Simi, María and Simionescu, Radu and Simtkó, Katalin and Šimková, Mária and Simonarson, Haukur Barri and Simov, Kiril and Sitchinava, Dmitri and Sither, Ted and Skachedova, Maria and Smith, Aaron and Soares-Bastos, Isabel and Solberg, Per Erik and Sonnenhauser, Barbara and Sourouf, Shafi and Sprungnoli, Rachele and Stamou, Vivian and Steingrímsson, Steinþórd and Stella, Antonio and Stephen, Abishek and Straka, Milan and Strickland, Emmett and Strnadová, Jana and Suhr, Alane and Sulestio, Yogi Lesmana and Sulubacak, Umut and Suzuki, Shingo and Swanson, Daniel and Szántó, Zsolt and Taguchi, Chihiro and Taji, Dima and Tamburini, Fabio and Tan, Mary Ann C. and Tanaka, Takaaki and Tanaya, Dipta and Tavoni, Mirko and Tellia, Samson and Testier, Isabelle and Testori, Marinella and Thomas, Guillaume and Tonelli, Sara and Torga, Liisi and Toska, Marsida and Trosterud, Trond and Turkhina, Anna and Tsarfaty, Reut and Türk, Utku and Tyers, Francis and Pórðarson, Sveinbjörn and Pörsteinsson, Vilhjálmur and Uematsu, Sumire and Untilov, Roman and Urešová, Zdeňka and Uria, Larraitz and Uszkoreit, Hans and Utka, Andrius and Vagnoni, Elena and Vajjala, Sowmya and Vak, Sokrates and van der Goot, Rob and Vanhove, Martine and van Niekerk, Daniel and van Noord, Gertjan and Varga, Viktor and Vedenina, Uliana and Venturi, Giulia and Villemonte de la Clergerie, Eric and Vincze, Veronika and Vlasova, Natalia and Wakasaka, Aya and Wallenberg, Joel C. and Wallin, Lars and Walsh, Abigail and Washington, Jonathan North and Wendt, Maximilian and Widmer, Paul and Wigderson, Shira and Wijono, Sri Hartati and Wille, Vanessa Berwanger and Williams, Seyl and Wirén, Mats and Witteman, Christian and Woldemariam, Tsegay and Wong, Tak-sum and Wroblewska, Alina and Wu, Qishen and Yamashita, Kayo and Yamazaki, Naoki and Yan, Chunxiao and Yasuoka, Koichi and Yavrumyan, Marat M. and Yenice, Arife Betül and Yildiz, Olcay and Yu, Zhuoran and Yuliawati, Arlisa and Žabokrtský, Zdeněk and Zahra, Shorouq and Zeldes, Amir and Zhou, He and Zhu, Hanzhi and Zhu, Yilun and Zhuravleva, Anna and Ziane, Rayan. 2023. Universal Dependencies 2.13. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. PID http://hdl.handle.net/11234/1-5287.

A. Appendix

Table 2 shows the most important relations going from a verb to a noun; in addition, it also shows compound relations going from a noun to a verb. It demonstrates that some treebanks favor the compound analysis much more than others, and three treebanks do not use the compound relation at all. Table 3 shows some of the most frequent light verbs across the South Asian treebanks. Cognates are clearly observable in the Indo-Aryan languages but their preference in the individual languages varies (there are substantial differences even between Hindi and Urdu).
Table 2: Selected relations between verbs and nouns in UD 2.13 treebanks (only main relation types are shown, subtypes are merged with their main types). The relations go from the verb to the noun except for the “reversed compound” columns, where the noun is the parent node. NV means that the noun immediately precedes the verb; NXV means that the noun precedes the verb but there are one or more words between them; analogously, VXN means that the verb comes first, with at least one word between it and the noun. Frequencies are shown per 10K words; an empty cell means that the relation did not occur at all while zero means that it did occur but the normalized frequency is rounded down to 0.
<table>
<thead>
<tr>
<th>English</th>
<th>Hindi</th>
<th>HDTB</th>
<th>PUD</th>
<th>Urdu</th>
<th>Kangri</th>
<th>Bhojpuri</th>
<th>Bengali</th>
<th>Marathi</th>
<th>Sinhala</th>
<th>Malayalam</th>
</tr>
</thead>
<tbody>
<tr>
<td>do / make</td>
<td>karanā</td>
<td>118</td>
<td>56</td>
<td>knā</td>
<td>170</td>
<td>karaṇā</td>
<td>48</td>
<td>kara</td>
<td>38</td>
<td>karañ</td>
</tr>
<tr>
<td>make</td>
<td>karānā</td>
<td>3</td>
<td>1</td>
<td>krānā</td>
<td>1</td>
<td>krūnā</td>
<td>48</td>
<td>kara</td>
<td>63</td>
<td>karañe</td>
</tr>
<tr>
<td>happen / be</td>
<td>honā</td>
<td>22</td>
<td>12</td>
<td>hūnā</td>
<td>37</td>
<td>hoṇā</td>
<td>76</td>
<td>ho/bā</td>
<td>72</td>
<td>hauya</td>
</tr>
<tr>
<td>happen / be</td>
<td>honā</td>
<td>22</td>
<td>12</td>
<td>hūnā</td>
<td>37</td>
<td>hoṇā</td>
<td>76</td>
<td>ho/bā</td>
<td>72</td>
<td>hauya</td>
</tr>
<tr>
<td>give</td>
<td>denā</td>
<td>24</td>
<td>6</td>
<td>denā</td>
<td>36</td>
<td>denā</td>
<td>20</td>
<td>de</td>
<td>17</td>
<td>deñe</td>
</tr>
<tr>
<td>take</td>
<td>lenā</td>
<td>6</td>
<td>3</td>
<td>lenā</td>
<td>7</td>
<td>lañā</td>
<td>4</td>
<td>la</td>
<td>3</td>
<td>la</td>
</tr>
<tr>
<td>apply / put</td>
<td>lagānā</td>
<td>7</td>
<td>1</td>
<td>lgānā</td>
<td>2</td>
<td>laga</td>
<td>4</td>
<td>la</td>
<td>3</td>
<td>la</td>
</tr>
<tr>
<td>seem</td>
<td>laganā</td>
<td>1</td>
<td>1</td>
<td>lgānā</td>
<td>2</td>
<td>laganā</td>
<td>4</td>
<td>laganā</td>
<td>4</td>
<td>laganā</td>
</tr>
<tr>
<td>keep / put</td>
<td>rakanā</td>
<td>2</td>
<td>1</td>
<td>rhānā</td>
<td>7</td>
<td>rakanā</td>
<td>4</td>
<td>rakanā</td>
<td>4</td>
<td>rakanā</td>
</tr>
<tr>
<td>stay</td>
<td>rahanā</td>
<td>2</td>
<td>0</td>
<td>rhanā</td>
<td>2</td>
<td>rhanā</td>
<td>4</td>
<td>rhanā</td>
<td>2</td>
<td>rhanā</td>
</tr>
<tr>
<td>create / make</td>
<td>banānā</td>
<td>2</td>
<td>0</td>
<td>bnānā</td>
<td>3</td>
<td>bnānā</td>
<td>3</td>
<td>bnānā</td>
<td>3</td>
<td>bnānā</td>
</tr>
<tr>
<td>be / become</td>
<td>banānā</td>
<td>2</td>
<td>0</td>
<td>bnānā</td>
<td>3</td>
<td>bnānā</td>
<td>3</td>
<td>bnānā</td>
<td>3</td>
<td>bnānā</td>
</tr>
<tr>
<td>come</td>
<td>ánā</td>
<td>2</td>
<td>3</td>
<td>ánā</td>
<td>3</td>
<td>ánā</td>
<td>8</td>
<td>ánā</td>
<td>3</td>
<td>ánā</td>
</tr>
<tr>
<td>drive</td>
<td>calānā</td>
<td>0</td>
<td>1</td>
<td>calānā</td>
<td>1</td>
<td>calānā</td>
<td>1</td>
<td>calānā</td>
<td>1</td>
<td>calānā</td>
</tr>
<tr>
<td>go / walk</td>
<td>calanā</td>
<td>1</td>
<td>2</td>
<td>člānā</td>
<td>1</td>
<td>cala</td>
<td>8</td>
<td>cala</td>
<td>8</td>
<td>cala</td>
</tr>
<tr>
<td>meet</td>
<td>milanā</td>
<td>2</td>
<td>1</td>
<td>mlñā</td>
<td>1</td>
<td>mlñā</td>
<td>1</td>
<td>mlñā</td>
<td>1</td>
<td>mlñā</td>
</tr>
<tr>
<td>show / express</td>
<td>jatānā</td>
<td>3</td>
<td>0</td>
<td>mlñā</td>
<td>1</td>
<td>mlñā</td>
<td>1</td>
<td>mlñā</td>
<td>1</td>
<td>mlñā</td>
</tr>
<tr>
<td>pick</td>
<td>uṭhānā</td>
<td>2</td>
<td>1</td>
<td>āṭhānā</td>
<td>1</td>
<td>uḍānā</td>
<td>4</td>
<td>uḍānā</td>
<td>4</td>
<td>uḍānā</td>
</tr>
<tr>
<td>cause</td>
<td>dilānā</td>
<td>1</td>
<td>1</td>
<td>dlānā</td>
<td>1</td>
<td>dlānā</td>
<td>1</td>
<td>dlānā</td>
<td>1</td>
<td>dlānā</td>
</tr>
<tr>
<td>put</td>
<td>dālanā</td>
<td>1</td>
<td>0</td>
<td>dālānā</td>
<td>1</td>
<td>dālānā</td>
<td>1</td>
<td>dālānā</td>
<td>1</td>
<td>dālānā</td>
</tr>
<tr>
<td>get / find</td>
<td>pānā</td>
<td>0</td>
<td>1</td>
<td>pānā</td>
<td>2</td>
<td>pānā</td>
<td>4</td>
<td>pānā</td>
<td>4</td>
<td>pānā</td>
</tr>
<tr>
<td>kill</td>
<td>māranā</td>
<td>1</td>
<td>1</td>
<td>mārnā</td>
<td>1</td>
<td>mārnā</td>
<td>1</td>
<td>mārnā</td>
<td>1</td>
<td>mārnā</td>
</tr>
<tr>
<td>fall</td>
<td>paḍanā</td>
<td>0</td>
<td>1</td>
<td>prnā</td>
<td>1</td>
<td>paunā</td>
<td>8</td>
<td>paunā</td>
<td>8</td>
<td>paunā</td>
</tr>
</tbody>
</table>

Table 3: Selected lemmas of verbs that are connected with a noun via the compound relation (or its subtype), with the verb as the parent, in UD 2.13 treebanks. Frequencies are shown per 10K words; an empty cell means that the verb did not occur at all while zero means that it did occur but the normalized frequency is rounded down to 0.
Sign of the Times: Evaluating the use of Large Language Models for Idiomaticity Detection

Dylan Phelps\textsuperscript{1,2}, Thomas Pickard\textsuperscript{2}, Maggie Mi\textsuperscript{2}, Edward Gow-Smith\textsuperscript{2}
Aline Villavicencio\textsuperscript{1,3}

\textsuperscript{1}Healthy Lifespan Institute, The University of Sheffield, United Kingdom
\textsuperscript{2}Department of Computer Science, The University of Sheffield, United Kingdom
\textsuperscript{3}Institute for Data Science and Artificial Intelligence, The University of Exeter, United Kingdom

\{drsphelps1, tmpickard1, zmi1, egow-smith1\}@sheffield.ac.uk
a.villavicencio@exeter.ac.uk

Abstract

Despite the recent ubiquity of large language models and their high zero-shot prompted performance across a wide range of tasks, it is still not known how well they perform on tasks which require processing of potentially idiomatic language. In particular, how well do such models perform in comparison to encoder-only models fine-tuned specifically for idiomaticity tasks? In this work, we attempt to answer this question by looking at the performance of a range of LLMs (both local and software-as-a-service models) on three idiomaticity datasets: SemEval 2022 Task 2a, FLUTE, and MAGPIE. Overall, we find that whilst these models do give competitive performance, they do not match the results of fine-tuned task-specific models, even at the largest scales (e.g. for GPT-4). Nevertheless, we do see consistent performance improvements across model scale. Additionally, we investigate prompting approaches to improve performance, and discuss the practicalities of using LLMs for these tasks.

Keywords: large language models, idiomaticity detection, prompting, scaling

1. Introduction

Large, pre-trained language models (LLMs) are becoming increasingly popular in academic, industrial, and lay spheres due to their ability to perform well across a range of tasks in a zero-shot or few-shot prompting set-up, including question answering, common-sense reasoning (OpenAI, 2023; Gemini Team, 2023), and machine translation (Xu et al., 2023; Koshkin et al., 2023; Koshkin et al., 2023; Koshkin et al., 2023; Koshkin et al., 2023). Despite this, there is yet to be an analysis of how well such models are able to handle potentially idiomatic language. Much previous work has shown that smaller, encoder-only transformer models have poor performance in identifying and representing idiomatic expressions when pre-trained on a large general dataset (Nandakumar et al., 2019; Garcia et al., 2021). However, the performance of such models increase hugely when they are fine-tuned on a task-specific dataset containing a large number of idiomatic expressions (Madabushi et al., 2021; Zeng and Bhat, 2021). This fine-tuning procedure, however, requires dedicated hardware and training, something that isn’t possible with LLMs on an academic budget.

In this work, we benchmark the performance of several widely-used LLMs (using both software-as-a-service remote implementations and local instances) on three in-context idiomaticity detection datasets; the idiom portion of FLUTE (Chakrabarty et al., 2022), MAGPIE (Haagsma et al., 2020), and SemEval 2022 Task 2a (Tayyar Madabushi et al., 2022). FLUTE and MAGPIE cover English (EN) only, while the SemEval dataset also includes expressions in Brazilian Portuguese (PT-BR) and Galician (GL).

Overall, our experiments show that large LLMs give competitive performance on idiomaticity datasets, which can be generally applied due to the lack of type specific fine-tuning, but nevertheless lag in general behind much-smaller finetuned encoder-only models. We also find that idiomaticity detection performance still scales with the number of parameters in the model. Finally, we discuss a number of considerations affecting the models’ performance and the practicality of using them for idiomaticity detection, including the training dataset and the capability of the model to follow instructions given in the prompt.

2. Datasets

We investigate the performance of LLMs on three datasets consisting of potentially idiomatic expressions in context. The datasets are chosen to provide a diverse set of potentially idiomatic expressions which feature a range of morphological forms and variations across two different tasks: textual entailment and idiomaticity detection. 1,859 different English target expressions are represented across the three datasets. We focus on English, but the inclusion of SemEval 2022 Task 2a allows us to additionally explore performance across languages.
2.1. FLUTE

FLUTE (Chakrabarty et al., 2022) frames the understanding of four kinds of figurative language (sarcasm, simile, metaphor and idioms) as a natural language inference (NLI) task, in which pairs of literal and figurative sentences are labelled as either entailing or contradicting one another. The sentence pairs are generated using a model-in-the-loop approach, with base text generated by GPT-3 which is then edited by crowdworkers and reviewed by experts.

For our analysis, we consider only the idiom section of the FLUTE dataset, which consists of 1,768 training examples across 479 idioms and a further 250 test examples across 69 idioms. No idiom appears in both the training and test sets.

Chakrabarty et al., 2022 provide benchmark performance metrics using T5 models (Raffel et al., 2020) on the FLUTE training data, reporting 79.2% accuracy (0.791 macro-F1). A FigLang22 shared task using the FLUTE dataset (Saakyan et al., 2022) attracted several entries, with the best-performing systems developed by (Gu et al., 2022) and (Bigoulaeva et al., 2022). The latter adopt a pipeline approach, improving the T5 baseline by sequentially fine-tuning on e-SNLI dataset (Camburu et al., 2018) and IMPLI (which incorporates figurative language) (Stowe et al., 2022), followed by the task dataset. Using the authors’ published outputs, we calculate a macro-average F1 of 0.952 on the idiom portion of the FLUTE test set.

2.2. SemEval 2022 Task 2a

SemEval 2022 Task 2a (Tayyar Madabushi et al., 2022) is a binary classification idiomaticity detection task, in which a potentially idiomatic noun compound, as used in a given context sentence, must be labelled as either literal or idiomatic. The dataset includes compounds across a range of idiomaticity, including fully compositional (insurance company) as well as partially (eager beaver) and entirely opaque (sugar daddy) items. The task offers both “one-shot” and “zero-shot” settings; the former is evaluated with new context instances of previously-seen items, while the latter uses compounds not present in the training data for evaluation.

The test set for the task contains 50 compounds each in English (with 916 instances), Brazilian Portuguese (713 instances) and Galician (713 instances).

Table 1 shows the macro-F1 scores in the zero-shot and one-shot settings for the baseline models (fine-tuned multilingual mBERT, per Madabushi et al., 2021) and the best-performing entries to the shared task.\(^1\)

<table>
<thead>
<tr>
<th>Setting</th>
<th>Reference</th>
<th>Language</th>
<th>EN</th>
<th>PT</th>
<th>GL</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero-Shot</td>
<td>Best</td>
<td></td>
<td>0.902</td>
<td>0.828</td>
<td>0.928</td>
<td>0.890</td>
</tr>
<tr>
<td></td>
<td>Baseline</td>
<td></td>
<td>0.707</td>
<td>0.680</td>
<td>0.507</td>
<td>0.654</td>
</tr>
<tr>
<td>One-Shot</td>
<td>Best</td>
<td></td>
<td>0.964</td>
<td>0.894</td>
<td>0.937</td>
<td>0.939</td>
</tr>
<tr>
<td></td>
<td>Baseline</td>
<td></td>
<td>0.886</td>
<td>0.864</td>
<td>0.816</td>
<td>0.865</td>
</tr>
</tbody>
</table>

Table 1: Reference scores (Macro F1) for SemEval 2022 Task 2a.

2.3. MAGPIE

MAGPIE (Haagsma et al., 2020) is a corpus of instances of potentially idiomatic expressions (PIEs – expressions which have multiple senses, including at least one with a high level of idiomaticity), in which each instance has been annotated as either idiomatic, literal, or other (proper noun, etc.) by a group of crowd-sourced workers. The PIEs in the dataset are chosen from three online dictionaries and so have a wide range of forms and frequencies.

The final dataset consists of 56,622 annotated instances, of which 70% are idiomatic, 28% are literal and 1% are other. In our experiments we use the test split of the randomly split dataset, which has 4,840 instances across 1,134 PIEs.

Haagsma et al. (2020) do not provide baseline models for the MAGPIE data, but several benchmarks are provided by Zeng and Bhat (2021).

2.4. Construction Artifacts

Recent work by Boisson et al. (2023) has found that language models tuned for metaphor identification (in which they include idiomaticity detection) on artificially-constructed datasets (i.e. those not sampled from ‘naturally-occurring’ text) can perform well when the target expression or the surrounding context are hidden from the model, “in both cases close to the model with complete information”.

As our experiments employ pre-trained LLMs without fine-tuning for the idiomaticity detection task, we anticipate that the concerns highlighted by Boisson et al. (2023) should not affect our findings. While the training regimes for many of the models we examine are not public, it seems likely that they have consumed large quantities of training data containing ‘naturally distributed’ idiomatic expressions.

It is also worth noting that we can not rule out the possibility that these LLMs’ training data includes the training or test datasets under evaluation\(^2\), and it is likely (for SemEval and MAGPIE) that the context sentences could have been ‘seen’ by the mod-

\(^1\)For the one-shot setting, the best-performing model is a fine-tuned multilingual XLM-RoBERTa, as described in Chu et al. (2022).

\(^2\)The SemEval test set is publicly available only without labels; FLUTE and MAGPIE are public.
3. Models

To be able to compare results from a range of currently-available LLMs, we evaluate both software-as-a-service (SaaS) and local instances of open models. To maximise applicability of our findings to researchers, we focus on local instances that can be run on consumer-level hardware (targeting a machine with 32GB RAM and 12GB VRAM).

Table 2 summarises the models used in our experiments, including the parameter count (where available), cost to run for SaaS models, and whether the training dataset is multilingual.

<table>
<thead>
<tr>
<th>Model</th>
<th>Params (billions)</th>
<th>Cost ($US per 1000 tokens)</th>
<th>Multilingual</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT-3.5-turbo</td>
<td>Unknown</td>
<td>0.0005</td>
<td>Y</td>
</tr>
<tr>
<td>GPT-4-turbo</td>
<td>Unknown</td>
<td>0.01</td>
<td>Y</td>
</tr>
<tr>
<td>GPT-4</td>
<td>Unknown</td>
<td>0.03</td>
<td>Y</td>
</tr>
<tr>
<td>Gemini-1.0 Pro</td>
<td>Unknown</td>
<td>0.000125</td>
<td>Y</td>
</tr>
<tr>
<td>Llama2-7B-chat</td>
<td>7</td>
<td>N/A</td>
<td>N</td>
</tr>
<tr>
<td>Llama2-13B-chat</td>
<td>13</td>
<td>N/A</td>
<td>N</td>
</tr>
<tr>
<td>Llama2-708-chat</td>
<td>70</td>
<td>N/A</td>
<td>N</td>
</tr>
<tr>
<td>Phi-2</td>
<td>2.5</td>
<td>N/A</td>
<td>N</td>
</tr>
<tr>
<td>Mistral-7B</td>
<td>7</td>
<td>N/A</td>
<td>N</td>
</tr>
<tr>
<td>Flan-T5-Small</td>
<td>0.08</td>
<td>N/A</td>
<td>Y</td>
</tr>
<tr>
<td>Flan-T5-Base</td>
<td>0.25</td>
<td>N/A</td>
<td>Y</td>
</tr>
<tr>
<td>Flan-T5-Large</td>
<td>0.78</td>
<td>N/A</td>
<td>Y</td>
</tr>
<tr>
<td>Flan-T5-XL</td>
<td>3</td>
<td>N/A</td>
<td>Y</td>
</tr>
<tr>
<td>Flan-T5-XXL</td>
<td>11</td>
<td>N/A</td>
<td>Y</td>
</tr>
</tbody>
</table>

Table 2: Characteristics of the models evaluated.

3.1. Software-as-a-service Models

3.1.1. OpenAI

OpenAI models are seen to be the current state of the art in SaaS models. GPT-4 (OpenAI, 2023), their current largest model, has been shown to achieve or exceed human-level performance in a number of commonly used benchmarks. We evaluate GPT-3.5-turbo (gpt-3.5-turbo-0613), GPT-4-turbo (gpt-4-0125-preview) and GPT-4 (gpt-4) in this work. GPT-3.5 is a smaller model created as a test run during the development of GPT-4, and GPT-4-turbo is an optimised and more recent variant of GPT-4. The parameter counts for these models are not known, but it is assumed that GPT-4 is substantially larger than GPT-3.5.

3.1.2. Google

Google provides access to a number of models of varying size and price through its VertexAI API. In this work we evaluate the performance of the Gemini Pro 1.0 model. Gemini Pro is trained on a multimodal and multilingual dataset and its performance exceeds that of GPT-3.5 on a number of benchmarks (Gemini Team, 2023).

3.2. Local Models

Additionally, we evaluate the performance of popular open models that can be run locally. The models chosen are the Llama2 models, (Touvron et al., 2023) Llama2-7B-chat and Llama2-13B-chat, Phi-2 (Li et al., 2023; Abdin et al., 2023), and the CapybaraHermes\(^3\) variant of Mistral-7B (Jiang et al., 2023).

To ensure that the models can be run on consumer-level hardware we use quantized variants of each model with 7B or more parameters. Quantization (Dettmers et al., 2022; Frantar et al., 2023) involves converting each parameter from full 16-bit floating point numbers to a set of \(2^n\) discrete values. This massively reduces the size of the models so they can be run on a wider range of hardware, with a trade-off of lower performance. We use Q5_K_S quantisation variants, which use 5-bit quantization, provided by TheBloke on Huggingface\(^4\). 5 bit quantization has been shown to have minimal impact on the performance of the model\(^5\).

To run the models we use the Huggingface transformers library (Wolf et al., 2020) for Phi-2 and llama.cpp\(^6\) for all the quantized models.

3.3. Multilingual Models

We also explore the performance of multilingual models. In particular, we target our exploration to variants of the Flan-T5 models (Chung et al., 2022): Flan-T5-Small, Flan-T5-Base, Flan-T5-Large, Flan-T5-XL, and Flan-T5-XXL.

We are interested in how multilingual models’ performance on idiomatic language-related tasks differs from monolingual ones. Moreover, we want to investigate the extent to which the performance is impacted by model size.

4. Results

Our main results across the three datasets (using our default prompts) are shown in Table 3. To make our results representative and generalisable, we ran the models multiple times, where not computation or cost prohibitive – all of the Flan models were run three times, whilst the Gemini Pro and GPT-3.5 models were run twice on SemEval, which is particularly important for reducing the variance of the results when testing different prompting methods; all other models were run once only.

---

\(^3\)https://huggingface.co/argilla/CapybaraHermes-2.5-Mistral-7B

\(^4\)https://huggingface.co/TheBloke


\(^6\)https://github.com/ggerganov/llama.cpp
Table 3: Main results of our models across the three idiomaticity datasets. All results presented are macro-average F1 scores over the two classes. Baseline results are taken from Madabushi et al. (2021), Chakrabarty et al. (2022) and Zeng and Bhat (2021). ‘Best’ results (in all cases using models fine-tuned on the task training data) are taken from Chu et al. (2022), Bigoulaeva et al. (2022) and Zeng and Bhat (2021). For SemEval, the ‘zero-shot’ setting is reported.

Comparing the results with the baseline and best-performing models, we can see that while the performance of large, contemporary LLMs may be higher than out-of-the-box encoder-only models, there is still a gap between them and the results which can achieved by encoders fine-tuned to the particular tasks. However, given the work of Boisson et al. (2023) on construction artifacts within datasets for idiomaticity detection, the ability of LLMs to disambiguate a wide-range of PIEs without additional fine-tuning shows the general ability of these models to detect idiomaticity, which may not have been achieved by fine-tuned encoders.

4.1. Model Scaling

With the exception of the Mistral-7B model, there is a significant gap in performance between the smaller, locally-run models and the larger SaaS models. We can also see the same trend for our Llama2 models, where the larger Llama2-13B model outperforms the smaller Llama2-7B one on all datasets and splits. From the results of the Flan-T5 model variants, as shown in Figure 1, there is a clear trend that increasing model size leads to improved performance. This trend appears to slow down somewhat after model size reaches around 3B parameters (Flan-T5-XL), though performance on the MAGPIE dataset continues to grow.

4.2. Prompts

Due to the differing input formats required by the various models, we use slightly different prompts. Here, we show our default prompts used for the GPT models. For SemEval and MAGPIE, we use:

“Disambiguate whether the given expression is used idiomatically or literally in the given context, returning ‘i’ if the expression is being used idiomatically or ‘l’ if literally. Expression: <PIE>. Context: <target sentence>. Only return one letter (i or l).”

For the FLUTE entailment task, we use:

“Disambiguate whether the second sentence follows from the first, returning ‘entailment’ if it does, and ‘contradiction’ if not. Sentence 1: <premise sentence> Sentence 2: <hypothesis sentence>. ”
4.3. Prompt Engineering

We investigate the effect of several prompt variations on performance for GPT-3.5-turbo on the English SemEval test set. As part of the OpenAI API, there are two prompts: "system" and "user". We first tried using the system prompt to define the task for the model, but obtained better performance using only the user prompt – this aligns with the experiences of others that GPT-3.5 often doesn’t follow the system prompt well, unlike GPT-47.

We present our results for this in Table 4. Note that variation between runs using the same prompting strategy is high (up to 0.04 F1), which leads to difficulty in discerning the effect of changing the prompt.

Expert impersonation is motivated by work which has shown that prompting LLMs to impersonate domain experts can lead to higher performance (Salewski et al., 2024). As such, we tried two approaches: starting the prompt with “You are an expert in language use.” or “You are an expert in idiomatic language.”. However, we find that neither of these approaches lead to improved performance. Interestingly, replacing the word “Literal” with "Compositional" did seem to have a positive effect. We found that removing the instruction to explicitly return only one letter (’i’ or ‘l’) led the model to occasionally return other outputs, which causes a drop in performance (as we treat such responses as invalid). For the English subset, this is the case for 3% of outputs (28 out of 916 examples).

4.3.1. Language Prompts

Since SemEval has test data in English, Portuguese, and Galician, we experiment with a) explicitly stating the language of the sentence in the prompt, and b) translating the prompt using a commercial machine translation tool. We perform this analysis for GPT-3.5-turbo, Gemini 1.0 Pro, and Flan-T5-XXL, with results shown in Table 5.

For Gemini 1.0 Pro and Flan-T5-XXL we see performance improvement for Galician under both of these approaches, with higher performance when translating the prompt. We hypothesise that both English and Portuguese are likely well-represented in the model training data, and LLMs in general work well in multilingual settings (Shi et al., 2022). However, Galician is likely to be both rare and potentially confused with Portuguese when the language is not specified, or when there is less text in that language available in the prompt. It would be interesting to experiment further with similar language pairs.

Not shown here is that we recorded reduced performance for English across all three models when specifying the language in the prompt (0.739 to 0.674 for GPT-3.5-turbo, 0.771 to 0.732 for Gemini 1.0 Pro, 0.716 to 0.706 for Flan-T5-XXL). It is possible that additional prompt tokens specifying the language may act as a ‘distractor’ when it is the de facto default, and the nature of the generative models means that we can anticipate variation in responses to identical prompts.

4.4. Few-shot Prompting

The “one-shot” setting of SemEval 2022 Task 2a (in which further examples of the target PIE in context are made available) allows for the investigation of passing examples to the model through the prompt. We thus experiment with doing so for GPT-3.5-turbo, Gemini 1.0 and Flan-T5-XXL. We try two configurations: passing one example per PIE (one-shot), and passing all the examples that are available in the dataset (few-shot)8. These results are shown in Table 6.

Interestingly, the impact of few-shot prompting varies across the models. Flan-T5-XXL benefits the most from this, with stark and consistent performance improvements across the three settings and across all three languages – the overall F1 jumps from 0.580 in the Zero Shot setting to 0.805 in the Few Shot setting.

Further to this we analyse the performance of all size Flan-T5 models, and present a heatmap illustrating the impacts on performance stemming from zero-shot and few-shot scenarios in Table 7.

Table 4: Results (macro F1) on the test set of SemEval with GPT-3.5-turbo using prompt engineering.

<table>
<thead>
<tr>
<th>Language</th>
<th>PT</th>
<th>GL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>0.739</td>
<td>0.635</td>
</tr>
<tr>
<td>“Expert in language use”</td>
<td>0.717</td>
<td></td>
</tr>
<tr>
<td>“Expert in language use” + Idiomatic vs. Compositional</td>
<td>0.538</td>
<td></td>
</tr>
<tr>
<td>No “Only return one letter (i or l).”</td>
<td>0.633</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: GPT 3.5-turbo, Gemini 1.0, and Flan-T5-XXL results for Portuguese and Galician on SemEval using multilingual prompts.

<table>
<thead>
<tr>
<th>Language</th>
<th>PT</th>
<th>GL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>0.553</td>
<td>0.567</td>
</tr>
<tr>
<td>Language Prompt</td>
<td>0.554</td>
<td>0.604</td>
</tr>
<tr>
<td>Translated</td>
<td>0.541</td>
<td>0.512</td>
</tr>
</tbody>
</table>

Table 6: Few-shot results for GPT-3.5-turbo, Gemini 1.0, and Flan-T5-XXL.

8Where available, the one-shot training data has one idiomatic example for each PIE, and one literal example. However, for some PIEs just one of these is present.
The smallest models benefited the most from seeing one or more examples before inference. In the best cases, performance in English improved by 0.432 in the one-shot setting and 0.516 in the few-shot setting. Interestingly, few-shot prompting can be seen to improve performance across Portuguese and Galician examples in all model settings, apart from T5-FLAN-Base and Large where there is little, or no improvement. It appears that Flan-T5-Base seems to be least improved by prompting with examples, with a negative effect on performance in few-shot prompting settings. In the one-shot setting, improvement in model performance is minor. The Large, XL and XXL models also benefited from one- and few-shot prompting, with Flan-T5-XL seeing the most performance enhancement. It appears that whilst models follow “bigger is better” in zero-shot settings, they do not necessarily follow this pattern under one/few-shot prompting. In fact, the best performance in the few-shot setting is with T5-Small, which at only 80M parameters achieves an overall F1 of 0.821, the best performance of any of the models we have evaluated in this paper. This is in significant contrast to performance on MAGPIE and FLUTE, where zero-shot performance is very low. The model is likely learning some artefacts from the data such as predicting only one label for a given PIE in the SemEval dataset.

Gemini 1.0 Pro also achieves consistent (though smaller) performance improvements from Zero Shot to One Shot to Few Shot, but the performance for English reverses this pattern. We also see a big jump in performance between Zero Shot and One Shot for Galician, which we again attribute to the rarity of this language and its similarity with Portuguese.

GPT-3.5-turbo is hindered by providing examples. The reasons for this are unclear, but this may be linked to the inability shown by GPT-3.5 to follow system prompts. If the model is not successfully following longer prompts then they may effectively introduce noise and lead to worse performance, as we saw when comparing results with and without system prompts.

5. Discussion

5.1. Task Labelling

The majority of the models we examined achieved high performance on the FLUTE dataset. We attribute this to the nature of FLUTE’s evaluation being distinct from MAGPIE and SemEval. For the latter two, the model is asked to label ‘idiomatic’ or ‘literal’ use of a given idiom, whereas, in the FLUTE STS task, the model is required to pick out the contradiction or entailment relationship between two sentences.

This means that a model might not necessarily require ‘knowledge’ of the target idiom to succeed, but could determine the relationship between the two sentences from other information, as facilitated by contextualised embeddings (Boisson et al., 2023). Moreover, the model is likely to have encountered similar tasks during its pre-training. Flan-T5 models are instruction-refined versions of T5 (Raffel et al., 2020; Chung et al., 2022), that have undergone exposure to over 1000 tasks during its fine-tuning process alone. Among these tasks are evaluations of entailment and contradiction judgments, akin to FLUTE, such as SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), CB (de Marneffe et al., 2019) and numerous other reasoning tasks (for details see Raffel et al., 2020; Chung et al., 2022).

5.2. Practicalities

In contrast with fine-tuned classification models, as prompted models are capable of open-ended generation, they may not output a response in the format requested. While the output may be readily interpretable by a human reader, this is not practical when evaluating large numbers of responses. Prompting for specific formats is easier for models which have undergone more instruction tuning (Ouyang et al., 2022; Rafailov et al., 2023), and is a key reason why the Mistral-7B model outperforms the Llama2 7B variant.
Prompted, generative models produce outputs which are subject to variation when they are repeatedly given the same prompt. While the user may have some control over this behaviour through ‘temperature’ parameters, this variability is inherent to generative models. When converting the outputs of such models to a labelling decision, this variability will also affect the results.

Despite their generally higher performance than the local models and their advantages when it comes to prototyping, there are a number of considerations specific to SaaS models which may be significant. These include:

1. Cost – The larger models have a higher per-1000-tokens cost, which may lead to some evaluations being cost-prohibitive. Evaluating GPT-4 on the (relatively small) SemEval test set, for example, costs $11. Running evaluation on this model, especially across multiple runs for prompt tuning, etc. may potentially price out researchers with lower budgets.

2. Safety Features – Commercial SaaS models frequently include features designed to limit models and users’ capability to process or generate content which may cause harm. These features may also impact on researchers’ ability to use the tools, as they produce what are effectively false positives. For example, when using the VertexAI API for experiments with Gemini Pro, the API consistently refused to generate responses for a small number of prompts. These included certain contexts for the expression street girl which referred to prostitution or sexualization, but also the FLUTE sentence pair “Your brother is mature and behaves in an adult manner. Your brother is a big baby.” for the expression to be a big baby. We treat any such responses as incorrect in our statistics.

3. Service Changes – Changes to the underlying model can be made by the third party at any time, and can significantly impact the performance of the models and the consistency of results. Whilst undertaking this work the default gpt-3.5-turbo model changed from one released in June 2023, to one released in January 2024.

4. Rate limits – For larger datasets, the rate limits of commercial APIs can become an issue. As it is still not fully released, for a significant amount of time during the creation of this work, the daily rate limit for GPT-4-turbo was lower than the number of tokens in MAGPIE, which prevented us from completing any evaluation runs for this model and dataset combination.

5Replacing the word ‘adult’ with ‘grown-up’ convinced the service to generate a response.

6. Conclusion

In this work we have evaluated the performance of various large language models on three idiomaticity datasets (SemEval 2022 Task 2a, FLUTE, and MAGPIE). We have investigated locally-run models up to 13B parameters, as well as significantly larger models (GPT-3.5, GPT-4, and Gemini 1.0 Pro) accessed through commercial APIs. We perform an extensive analysis of the impact of several factors on performance; model size, prompt engineering and few-shot prompting. In addition, we discuss considerations for practitioners wishing to use these models in their own work, with emphasis on cost and practicalities such as the variability of outputs and the impacts of decisions made by the companies operating these services. Our overall findings are as follows: 1) LLMs at the highest scale are able to achieve competitive results for idiomaticity detection, and performance on FLUTE in particular seems to have saturated, but these general models do not match the performance of (much-smaller) encoder models fine-tuned for the specific idiomaticity detection tasks of SemEval and MAGPIE. 2) The performance of prompted, generative LLMs seems to scale consistently with parameter count for these datasets, indicating the potential of even bigger models to achieve further increases in performance. 3) While they are based on a relatively small set of examples, our experiments with multilingual models suggest that performance gains can be obtained by specifying the target language, translating prompts and by providing examples. However, the efficacy of these modifications depends on the model used and the language in question; they appear to harm performance for English (which is, presumably, the most-represented language in the model training regimens) while producing the largest benefit for the much rarer Galician.

Acknowledgments

We would like to thank the following sponsors for supporting this work:

- The Healthy Lifespan Institute (HELSI) at The University of Sheffield, UK EPSRC grant EP/T517835/1.

- The Centre for Doctoral Training in Speech and Language Technologies (SLT) and their Applications funded by UK Research and Innovation [grant number EP/S023062/1], and

- UK EPSRC grant EP/T02450X/1, by the Turing Institute Fellowship and by the UniDive COST Action.
7. Bibliographical References


Gregori Gerganov. 2024. Llama.cpp: Inference of meta’s llama model (and others) in pure c/c++.


Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks Are All You Need II: phi-1.5 technical report.


Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model.


8. Language Resource References


Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse.


Universal Dependencies for Saraiki

Meesum Alam, Francis M. Tyers, Emily Hanink, Sandra Kübler
Indiana University
{meealam,ftyers,emhanink,skuebler}@iu.edu

Abstract
We present the first treebank of the Saraiki/Siraiki [ISO 639-3 skr] language, using the Universal Dependency annotation scheme (de Marneffe et al., 2021). The treebank currently comprises 587 annotated sentences and 7,597 tokens. We explain the most relevant syntactic and morphological features of Saraiki, along with the decision we have made for a range of language specific constructions, namely compounds, verbal structures including light verb and serial verb constructions, along with different types of relative clauses.

Keywords: Saraiki, Universal Dependencies, Indo-Aryan Languages

1. Introduction

Universal Dependencies (UD) is now a widely used annotation scheme for developing syntactic annotations and parsers for a language (de Marneffe et al., 2021; Nivre and Zeman, 2020). It already covers around 220 languages around the world and is growing rapidly. These linguistically annotated corpora are crucial sources for NLP projects of any language. However, Indo-Aryan languages have received little attention in both UD and NLP applications. There currently exist Universal Dependency treebanks for Hindi (Ravishankar, 2017), Urdu (Ehsan and Butt, 2020), and Punjabi (in Gurmukhi script) (Arora, 2022). No lesser studied Indo-Aryan languages are covered in the UD project.

We present a UD treebank for Saraiki, a language of 25 million speakers, which is considered a neglected language in Pakistan. We follow the existing UD guidelines for the annotation where possible. Here, we describe our decisions for phenomena specific to the Saraiki language.

The remaining sections are as follows: Section 2 provides background on the Saraiki language, Section 3 discusses work on treebank construction for related languages, and Section 4 describes the corpus and annotation process. Section 5 discusses part of speech and morphological characteristics of those word classes necessary to understand the discussion of language specific phenomena, and Section 6 discusses the decisions made for language specific phenomena, namely compounds, verbal structures including light verb and serial verb constructions, as well as different types of relative clauses.

2. Saraiki

Saraiki is an Indo-Aryan language widely used in Pakistan and India. The language is one of the ancient languages of the region. Saraiki is spoken by around 25 million people in Southern and Southwestern Punjab and Northern Sindh (see the map in Figure 1). Saraiki is also known as Jataki, Multani, Thali, Riasti and Deraywal in various regions of the Punjab. Saraiki, also spelled Siraiki, is counted among the widely-spoken languages in the Pakistani provinces of Punjab and Khyber Pakhtunkhwa (KPK). It is the sister language of Punjabi and Sindhi but has not received much attention in linguistics research.

Saraiki is written from right to left in Perso-Arabic script. It is head-final and follows a basic Subject-Object-Verb (SOV) structure within clauses. According to Bashir and Conners (2019), Saraiki word order is relatively free: Topic and focus marking are generally achieved by changes in word order. Saraiki does not have definite or indefinite markers, but it does have numeric ﮨﮏ (hik ‘one’) to mark indefiniteness. Saraiki is a pro-drop language, it uses clitics/pronomial suffixes in perfective transitive sentences to mark the subjects on verbs. Saraiki has split ergative alignment in addition nominative-absolutive alignment. For more details, see section 6.2.1.
Saraiki shares morphological and syntactic features with Punjabi but differs on the phonological level, which has allowed it to evolve into a distinct but related language (Bashir and Conners, 2019). As the language has been spoken in different regions of Pakistan for a long time, multiple dialects have emerged over time. Shackle (1976) distinguishes six varieties: Southern Saraik, Northern Saraiki, Sindhi Saraiki, Jhangi Saraiki, and related language (Kachroo, 2018). Additionally, there are automated conversions of Urdu (Ehsan and Butt, 2020) and Hindi (Bhat et al., 2018) treebanks from constituent annotations.

For Saraiki, there is little research in the area of NLP. Alam et al. (2023) have developed a morphological analyzer for Saraiki, and Asghar et al. (2021) created a part of speech (POS) tagger. There is also ongoing work on a Saraiki wordnet under Higher Education of Pakistan’s Funding at Sarghoda University (Gul et al., 2021), but the system has not been released yet. For the development of NLP related tools, it is equally important to understand the linguistics phenomenon of a language; Bashir and Conners (2019) have published a descriptive grammar for Saraiki, which we used as the basis for our treebank annotations.

### Table 1: Textual basis of the Saraiki Treebank.

<table>
<thead>
<tr>
<th>Source</th>
<th>Sentences</th>
<th>Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Voice (Ardila et al., 2020)</td>
<td>5 712</td>
<td>52 300</td>
</tr>
<tr>
<td>Jhok Newspaper (Dhareja, 2017–2022)</td>
<td>56 000</td>
<td>1.15M</td>
</tr>
<tr>
<td>Linguistic examples</td>
<td>—</td>
<td>1 851</td>
</tr>
</tbody>
</table>

3. Related Work

NLP applications heavily rely on linguistically annotated resources; these resources have multiple functions as they test the linguistic theories, are used to train and evaluate parsing technologies, and provide insights into specific linguistic phenomena of a language (Nivre and Zeman, 2020). However, the Indo-Aryan (IA) languages lack good digital tools because of the scarcity of available corpora. This is also true for Universal Dependency treebanks; we find some IA languages added to the repository. These treebanks cover the major languages: Hindi (Tandon et al., 2016), Urdu (Bhat and Sharma, 2012), Marathi (Ravishankar, 2017), and Punjabi (Arora, 2022). Additionally, there are automated conversions of Urdu (Ehsan and Butt, 2020) and Hindi (Bhat et al., 2018) treebanks from constituent annotations.

For Saraiki, there is little research in the area of NLP. Alam et al. (2023) have developed a morphological analyzer for Saraiki, and Asghar et al. (2021) created a part of speech (POS) tagger. There is also ongoing work on a Saraiki wordnet under Higher Education of Pakistan’s Funding at Sarghoda University (Gul et al., 2021), but the system has not been released yet. For the development of NLP related tools, it is equally important to understand the linguistics phenomenon of a language; Bashir and Conners (2019) have published a descriptive grammar for Saraiki, which we used as the basis for our treebank annotations.

4. Corpus and Annotation Process

The Saraiki treebank currently consists of 587 sentences, corresponding to 7 597 tokens in total.

Our treebank is based on sentences from three different sources: from the Saraiki Common Voice corpus (Ardila et al., 2020), from the Jhok newspaper (Dhareja, 2017–2022)\(^1\), and sentences generated during the annotations discussions, to clarify decisions on specific syntactic phenomena in Saraiki. Table 1 shows the distribution of the different text types. Saraiki is under-resourced language and it is difficult to find digital texts in this language, thus limiting our options in creating a diverse textual basis for the treebank.

In a first step, the data was converted into CoNLL-U format and manually segmented. The data have been shared with Saraiki speakers and linguistics scholars in Pakistan. This helped in making decisions on parts of speech (POS) tagging. We manually annotated the corpus for parts of speech. Since there does not exist a standard POS tagging scheme for Saraiki, we left the XPOS category for future work. The POS tagged text was used for the development of a Saraiki morphological analyzer (Alam et al., 2023). Then we started annotating the corpus for universal dependencies. We currently have 587 sentences fully annotated, and will add more annotations in the future. Once we reach 1 000 sentences, the treebank will be published via the UD project.

The annotation is carried out in two steps by the first author, a native speaker of Saraiki, in consultation with the other authors. For part of speech tagging, difficult cases are resolved based on information from the the Saraiki dictionary (Jukes, 2019), along with consulting Saraiki speakers and experts from the Urdu Universal Dependency Treebank to validate decisions. The dependency relationships are annotated using Annotatrix (Tyers et al., 2017), in consultation with all co-authors and UD experts.

5. Saraiki Parts of Speech and Morphology

As of today, there does not exist a language specific part of speech tagging scheme for Saraiki. Even though there are schemes for Punjabi (Gill et al., 2009) and Urdu (Hardie, 2003), we focussed on the Universal POS tagset (Petrov et al., 2012), leaving the XPOS category for future work. All of the UD POS tags occur in our corpus; Table 2

---

\(^1\)These sentences are used with permission from the newspaper.
are mostly borrowed from Urdu or Persian, such as یامان (‘faith’) and رب (‘God’).

**Adjectives** In Saraiki, adjectives take the case and inflection of the nouns that they modify. If a noun is not case-marked, modifying adjectives agree with it in gender and number only.

**Pronouns and demonstratives** Saraiki does not distinguish between third person proximal and distal pronouns and demonstratives. Instead, the distal forms for he, she, that, those (‘oo’n) are used for both expressions alongside their proximal forms (‘ay ‘he, she, it, this, these’).

Following Bashir and Conners (2019), who identify a morphological difference between relative pronouns that stand alone or immediately precede a noun, we annotated relative pronouns as PRON where they function as independent pronouns and DET where they function as determining adjectives. The adjectival forms, unlike the stand-alone pronominal forms, inflect robustly for number, gender, and case of the noun they precede and modify.

### 6. Annotation Decisions

In this section, we focus on language specific constructions, focusing on the treatment of (split) ergative sentences, serial and light verbs, as well as compounds and relative clauses. Remember that Saraiki is head-final and written right to left.

#### 6.1. Compounds

Saraiki has a comprehensive system of creating multiword expressions and compounds in open and closed POS categories. In section 6.2, we will focus on the V-V compound in serial verb and light verb constructions. Here, we discuss an additional type of V-V compounding, reduplication, plus compounds involving nouns, reflexive pronouns, and adverbs.

**Reduplication** This is common for emphasis, for noun compounding and pluralization. In these cases, we annotate the verbs using compound:redup, with the first verb as the head. Interestingly, reduplication can occur with all open class categories. Verb reduplication is different from light or serial verb constructions. These verbs do not provide tense, aspect, and modality information, and they are not part of complex serial verb predicates. In example (1), گھت (‘put’) is reduplicated, either for emphasis or to indicate a quick action. As described above, reduplication can be used with almost all open categories of the

<table>
<thead>
<tr>
<th>POS Tag</th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOUN</td>
<td>1314</td>
<td>17.3</td>
</tr>
<tr>
<td>VERB</td>
<td>1231</td>
<td>16.2</td>
</tr>
<tr>
<td>PUNCT</td>
<td>759</td>
<td>10.1</td>
</tr>
<tr>
<td>ADJ</td>
<td>714</td>
<td>9.4</td>
</tr>
<tr>
<td>ADP</td>
<td>630</td>
<td>8.3</td>
</tr>
<tr>
<td>PRON</td>
<td>569</td>
<td>7.5</td>
</tr>
<tr>
<td>ADV</td>
<td>501</td>
<td>6.6</td>
</tr>
<tr>
<td>PROPN</td>
<td>417</td>
<td>5.5</td>
</tr>
<tr>
<td>AUX</td>
<td>387</td>
<td>5.1</td>
</tr>
<tr>
<td>CCONJ</td>
<td>386</td>
<td>5.1</td>
</tr>
<tr>
<td>DET</td>
<td>258</td>
<td>3.4</td>
</tr>
<tr>
<td>SCONJ</td>
<td>190</td>
<td>2.5</td>
</tr>
<tr>
<td>PART</td>
<td>188</td>
<td>2.5</td>
</tr>
<tr>
<td>INTJ</td>
<td>22</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Table 2: Distribution of Universal Dependency parts of speech tags in the Saraiki Treebank.

gives a detailed picture of the distribution of the tags in the Saraiki Treebank.

**Verbs** Similar to other Indo-Aryan languages, Saraiki verbs undergo derivational and inflectional processes. Saraiki verbs inflect for number, gender, tense, aspect, and mood. Adverbs, compounds, and reflexives can be derived from verbs via derivational verbal morphology. Additionally, Saraiki uses verb stem alteration. To describe those, we use work by Bashir and Conners (2019) on the eight different verb stem alterations as the basis for our annotations.

In Saraiki, certain verbs play a dual role. When occurring within a light verb construction, they take the role of auxiliaries, providing information on the verb’s aspect. Consequently, we distinguish between VERB and AUX, according to the structure. For infinitives, we follow decisions in the Punjabi treebank (Arora, 2022): We mark them as VERB in all instances, regardless of their semantic interpretation.

**Nouns** We found three types of nouns in our treebank: case-marked nouns, non case-marked nouns, and uninflected nouns. Most nouns are case-marked in addition to being inflected for gender and number. Saraiki uses four cases: direct, oblique, vocative, and ablative. Examples of nouns that can be case-marked are مان (‘mother’) and چھاں (‘shade’). The second type of nouns are non case-marked nouns. These nouns are borrowed from neighboring languages, and are adapted to suit Saraiki morphology. Examples of this type are بال (‘male child’) and ذات (‘caste’). The last category of nouns does not take any kind of inflections; these nouns...
grammar in Saraiki. In example (2), reduplication is used to emphasize the adverb \( \text{ﻭﻝ} \) (\textit{wul} ‘again’).

```
(1) put quickly”

```

Noun-Noun Compounds In Saraiki, there are a wide range of concepts that are expressed as noun-noun compounds. We use the \textit{compound} relation in these cases. Example 3 shows a combination of \( \text{ﻣﺎﮞ} \) (\textit{maa’n} ‘mother’) and \( \text{ﭘﯿﺆ} \) (\textit{piyo} ‘father’) meaning “parents”.

```
(3) "parents"

```

Reflexive Pronouns These are constructed by combining the two words \( \text{ﺍﭘﻨ郷ﮮ} \) (\textit{apnre} ‘own’) and \( \text{ﺁﭖ} \) (\textit{aap} ‘self’) in a multi-word expression (see example 4 and Figure 2). We follow the UD guidelines and use the \textit{compound} relation to combine those two words.

```
(4) "I will go to the shop by myself”

```

6.2. Verbs

In Saraiki, the verb system is more complex than in the neighbouring languages Punjabi, Urdu, and Hindko \cite{bashir2019}. Syntactically, Saraiki exhibits split ergativity in addition to pronominal suffixation onto verbs in some contexts. It uses two types of light verb constructions: one consisting of two verbs where one verb acts as an auxiliary, contributing only tense, aspect and modality information, and another consisting of a noun or adjective in addition to the light verb. Additionally, Saraiki employs serial verb constructions. We will discuss all these phenomena and annotation decisions in more detail below. In the Common Voice corpus by \cite{ardila2020}, out of all the verbs construction we found approximately 21% light verb constructions; interestingly, half of these light verb constructions use the verb \( \text{ﺗﮭﯿﻮݨ} \) (\textit{thivaṇ} ‘to become’). These numbers are based on the current treebank, but we expect the percentages to remain stable as we add more sentences.

6.2.1. Syntactic Split Ergativity

Saraiki belongs to the group of languages that have both nominative–accusative and ergative-absolutive alignment (see \cite{dixon1994} for an overview). According to \cite{bashir2019}, Saraiki shows an ergative-absolutive pattern only in perfective contexts, a pattern common across Indo-Aryan languages. It is important to know that unlike Urdu, Punjabi, and Hindi, Saraiki lacks a dedicated ergative morpheme. Consequently, the effects of this split are observable only in verbal agreement patterns. The generalization is that verbs agree with agents of transitive verbs and subjects of intransitive verbs in the same way in the imperfective aspect, but do not agree with agents of transitive verbs in the perfective aspect. Thus, while patients are oblique in imperfective contexts, it is agents that are oblique in perfective contexts. Table 3 lays out the case alignment pattern across imperfective and perfective contexts.

The aspectual contrast giving rise to this split is exemplified below. The imperfective sentence in example (5) shows a typical nominative-accusative agreement pattern, in which the verb
agrees with the nominative argument ﻗﺎﺳﻢ (Qasim ‘Qasim’). The same case and agreement pattern is found with intransitive verbs, which agree with their nominative subject.

In the perfective sentence in example (6) in contrast, the agent of the transitive verb ﻭﺮھی (parhi ‘read’), ﻗﺎﺳﻢ (Qasim ‘Qasim’) carries the oblique case, while the direct object ﻏير (kitaab ‘book’) carries nominative case. Notably, the verb in this context agrees with its direct object rather than its subject. The generalization is thus that, in perfective contexts only, agents of transitive verbs i) are oblique arguments ii) may not control subject agreement.

Example (7) shows an example of an ergative sentence, where we annotate the agent ﺎﭨlav (oo’n ‘he’), which is in the oblique case, is the subject, and ﻓﺮھ (ghar ‘house’) is the direct object in ergative case.

We note that another type of agent marking is also available. This strategy uses pronominal suffixes (clitics) on the verb to mark the grammatical features of the agent. In this type of structure, the transitive verb in the perfective form shows object agreement, with the pronominal agent cliticized onto the end of the verb. In example (8), the verb ﺒﯿﻤت (pita-m ‘I drank’) agrees with the noun ﺪ.container (paanri ‘water’), and the agent 1.M.SG is added to the end of the verb ﺒﯿﻤت. In example (9), the verb بھبخ (khād-i-s ‘he ate’) agrees with ﺑھا (bhaj-i ‘food-F.SG’), and the agent is marked on verb.

These constructions are possible only in the perfective forms. Note that while Bashir and Conners (2019) call these pronominal suffixes, Syed and Raza (2019) call them clitics. On either treatment, this type of construction is sensitive to the morphological features of the agent, which are marked on the verb. Following the UD guidelines, we annotate the argument as direct object obj.

This morphologically embedded ergativity (differential case marking) is also found in Hebrew (Glinert, 2004) and Hungarian (Bárány, 2012).

### 6.2.2. Serial Verb Construction

Serial verbs mostly conceptualize one event and are realized as one linear, complex predicate with-
out explicit coordination or subordination markers. This feature is common in many IA languages. Example (10) shows a sentence from our treebank, and Figure 3 shows our annotation. Since we do not yet know enough about the constraints on this construction, we decided to annotate the involved verbs serially. As Saraiki is a head final language (written from right to left), we mark the last verb as the head of the clause and create compound:lvc relations with other verbs. We anticipate changes to these annotations in the future once we have a better understanding of this construction.

6.2.3. Light Verb Constructions

In Saraiki, we find sequences of verbs where the main verb is followed by another ‘light’ verb, in addition to constructions in which a light verb is followed by a noun or adjective. In both cases, the light verb has little semantic content. In V-V LVCs, the second verb mostly contributes information about aspect or modality. All such constructions have been given the dependency of compound:lvc. We show an example in (11): 

In the treebank, we also found the verb تہیوئن (thivan ‘become’), a change of state verb (Bashir and Conners, 2019) in Saraiki, which, unlike پون (hovan ‘be’), appears in SVCs, LVCs, and as an auxiliary. تہیوئن (thivan ‘become’) can also be followed by another light verb construction. Where it occurs in a light verb construction, we mark it as a root with a compound:lvc dependency to the noun or verb (see examples (12) and (13)); when تہیوئن (thivan ‘become’) is not part of the light verb construction, we mark it as an auxiliary AUX (see example (14)).
6.3. Relative Clauses

In the Saraiki treebank, we found both finite and non-finite relative clauses. According to Bashir and Conners (2019), both types of clauses are used freely in Saraiki. While Saraiki uses externally headed relative clauses, it also uses internally headed and correlative forms. Saraiki uses جِیڑا (jera ‘that, which’) as a relativizer, which agrees with its head noun in number, gender, and case. These types of constructions are also available in Urdu (Ehsan and Butt, 2020; Bhat and Sharma, 2012) and Punjabi (Arora, 2022).

The examples discussed here are part of the sentences created for analyzing specific constructions in Saraiki. We use those examples so that we can focus on the relevant construction without interference from other syntactic phenomena.

Example (15) shows an externally headed relative clause, the annotation is shown in Figure 4. In such cases, جِیڑا (jera ‘which’) functions as relative pronoun; here it modifies زمیندار (zamindar ‘farmer’). We annotate the relative pronoun as nsubj of the verb of the relative clause، ستا (sutta ‘sleep-pst’), which in turn is dependent on the noun in the matrix clause via the acl:relcl relation.

Example (16) shows a version of the sentence with an internally headed relative clause, the annotation is shown in Figure 5. Here, the head noun زمیندار (zamindar ‘farmer’) occurs inside the relative clause, i.e., between the relative pronoun and the object of the relative clause (گھر ‘house’). Since this means that the relative clause has a relativizer and the noun it refers to, we have decided that the head noun زمیندار (zamindar ‘farmer-m-sg’) serves as the direct object (obj) in the matrix clause, and the relativizer serves as its determiner in a det relation. Consequently, the verb of the relative clause is dependent on the head noun via an acl:relcl relation. This analysis means that we do not consider the head noun to be part of the relative clause, since it provides the only “attachment site” for the relative clause.

Example (17) shows the same internally headed version, but in a different word order, with a fronted relative clause. The annotation is shown in Figure 6. Based on our current understanding, we

Figure 4: The annotation of the example of an externally headed relative clause in (15).

Figure 5: The annotation of the example of an internally headed relative clause in (16).
assume that the only difference between all three variants is in information structure.

Example (17) shows the same sentence, but uses a correlative. The annotation is shown in Figure 7. Correlative relative clauses are a variant of internally headed relative clauses where the relative clause is dependent on, and in an anaphoric relation to, a pronoun in the matrix clause. In example (18), the distal pronoun ﺎﻭﮯ (oun ‘that’) serves as the correlative. Consequently, we annotate it as the direct object of the matrix clause. The fronted relative clause is dependent on this pronoun. Parallel to the internally headed examples in (16) and (17), we analyze the relativizer as a determiner dependent on the subject of the relative clause.

7. Conclusion and Future Work

We have presented a treebank for Saraiki, annotated using Universal Dependencies. We discussed the textual basis of the treebank and a range of language specific syntactic phenomena. The treebank is work in progress, it currently comprises 587 sentences. We will extend it and release it once we reach 1 000 sentences.

For future work, we will need to have a closer look at the relative clauses. Additionally, we plan to automatically annotate the morphological features using the Apertium morphological analyzer for Saraiki (Alam et al., 2023). We hope that this treebank will spur deeper investigations of Saraiki as well as the creation of NLP tools for the language. We also plan to train a syntactic parser, and investigate zero-shot techniques to extend our work to other regional languages such as Punjabi (Shahmukhi), Hindko, and Khetrani.

8. Acknowledgements

We would like to thanks Pervaiz Qadir for developing the Saraiki corpus in Mozilla Common Voice and Zahoor Dhareja for giving permission for us to use data from Jhok newspaper for the treebank. We would also like to thanks Daniel Swanson and Daniel Zeman for their help with annotation decisions.
9. Bibliographical References


Domain-Weighted Batch Sampling for Neural Dependency Parsing

Jacob Striebel, Daniel Dakota, Sandra Kübler
Indiana University
{jstrieb,ddakota,skuebler}@indiana.edu

Abstract
In neural dependency parsing, as well as in the broader field of NLP, domain adaptation remains a challenging problem. When adapting a parser to a target domain, there is a fundamental tension between the need to make use of out-of-domain data and the need to ensure that syntactic characteristics of the target domain are learned. In this work we explore a way to balance these two competing concerns, namely using domain-weighted batch sampling, which allows us to use all available training data, while controlling the probability of sampling in- and out-of-domain data when constructing training batches. We conduct experiments using ten natural language domains and find that domain-weighted batch sampling yields substantial performance improvements in all ten domains compared to a baseline of conventional randomized batch sampling.

Keywords: neural dependency parsing, domain adaptation, batch optimization

1. Introduction
Dependency parsing, like many other machine learning problems, is sensitive to domain shifts between training and test data sets (Gildea, 2001; Petrov and Klein, 2007). To combat the negative effects of domain shifts when training a parser, several domain adaptation techniques have been studied (e.g., Rosa and Žabokrtský, 2015), although their effectiveness is often limited (e.g., Dredze et al., 2007).

A major factor that determines the success of domain adaptation methods is the amount of training data that is available in the adaptation-target domain (e.g., Daumé III, 2007; Dredze et al., 2007). To overcome the frequent problem of scarcity of target-domain training data, common techniques in parsing focus on selecting optimal source data points to boost performance in the target domain (Plank and van Noord, 2011; McDonald et al., 2011; Mukherjee and Kübler, 2017), with both delexicalized (Rosa and Žabokrtský, 2015) and lexicalized (Falenska and Çetinoğlu, 2017) similarity metrics showing improved data point selection.

Furthermore, to more effectively use all available source- and target-domain data, discrepancies in sizes between data sources have been handled using loss weighting on the different data sources (Dakota et al., 2021), allowing for noise reduction and improved information sharing.

Other approaches for encoding more domain-related information into a parser are to create data- or task-specific embeddings (Stymne et al., 2018; Li et al., 2019, 2020), which yield performance gains across languages and domains. While the further inclusion of language models into parsing architectures noticeably reduces performance gaps across domains, it still cannot fully overcome syntactic differences (Joshi et al., 2018; Fried et al., 2019; Yang et al., 2022). The situation is further complicated by the fact that the source and target domains may be different from those of the language model (Dakota, 2021).

We focus on a setting in which we have access to a small amount of annotated data from the target domain. In order to address the size difference between the data available for the target domain and other domains, we investigate a method that allows the use of all available source and target data during training, thus maximizing the available signal. More specifically, we use domain-weighted batch sampling (DWBS) to train a domain-expert neural dependency parser as an alternative to the conventional approach of randomized batch sampling (RBS).

Since we use some target domain data for training in our experiments, existing naming conventions are not easily usable. For this reason, we call data from the target domain in-domain data and data from all other domains out-of-domain data (i.e., any domain that is not the adaptation-target domain); we also use source data as a synonym for out-of-domain data. Note that our sampling strategy can also be used when we do not have any in-domain data but can determine the most similar domain among the out-of-domain data.

Our experiments are designed to answer the following two questions:

1. Can we improve parser performance, given a training data imbalance between in-domain and out-of-domain data, by replacing the standard batch sampling approach (i.e., RBS) with DWBS, which uses all available training data but favors training sentences drawn from the target evaluation domain?
2. Does DWBS yield faster training times than RBS? In other words, does DWBS reduce the number of sample sentences that a parser must observe before dev loss stops decreasing?

## 2. Domain-Weighted Batch Sampling

### 2.1. Batch Sampling

When training a neural network, there are several approaches that can be taken to creating batches, and the chosen approach will impact how a network converges, memory requirements, and possible performance among other effects on the model.

The simplest way of creating a batch is to select training samples in the order in which they appear in the training data file, which is called **sequential batch sampling** (SBS). However, this strategy may not be optimal since it repeatedly exposes the network to the same sequence of examples and thus may cause the network to indirectly learn specific batch characteristics that are not representative of the task as a whole (Chollet, 2018), which can result in catastrophic forgetting (French, 1999; Dachapally and Jones, 2018). Consequently, it is more common to create randomized permutations of the training data at the beginning of every epoch, which is called **randomized batch sampling** (RBS).

### 2.2. Domain-Weighted Batch Sampling

To leverage in-domain and all out-of-domain data, we extend RBS to **domain-weighted batch sampling** (DWBS). This allows for better inclusion of multi-source out-of-domain data, while still permitting the target domain to maintain higher influence on optimization.

To perform DWBS, before training begins the training data set is partitioned into disjoint in-domain and out-of-domain subsets. For each epoch, random permutations of the in-domain and out-of-domain subsets are separately generated. Each batch is then constructed by drawing sentences (without replacement) from the two permutations until the batch size is reached. We use the hyperparameter \( \mu \) to define the probability of choosing the next sentence from the in-domain permutation. For example, if \( \mu \) is equal to 0.45, there is a 45% chance of drawing the next sentence from the target (in-domain) permutation and 55% of drawing from the source (out-of-domain) permutation.

During an epoch, eventually we will attempt to draw from a permutation in which no sentences remain, at which point the current partially constructed batch is discarded and the current epoch is complete. A side-effect of the DWBS procedure is that different epochs may have different durations in terms of number of batches.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimizer</td>
<td>Adamw</td>
</tr>
<tr>
<td>( \beta_1, \beta_2 )</td>
<td>0.9, 0.99</td>
</tr>
<tr>
<td>Correction bias</td>
<td>False</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.0001</td>
</tr>
<tr>
<td>Weight decay</td>
<td>0.01</td>
</tr>
<tr>
<td>Gradient normalization</td>
<td>1</td>
</tr>
<tr>
<td>LR scheduler</td>
<td>Slanted triangular</td>
</tr>
<tr>
<td>Cut fraction</td>
<td>0.2</td>
</tr>
<tr>
<td>Decay factor</td>
<td>0.38</td>
</tr>
<tr>
<td>Discriminative fine tuning</td>
<td>True</td>
</tr>
<tr>
<td>Patience batches</td>
<td>200</td>
</tr>
<tr>
<td>Max steps</td>
<td>153,600</td>
</tr>
<tr>
<td>Embeddings</td>
<td>bert-base-cased</td>
</tr>
<tr>
<td>Embeddings dim</td>
<td>768</td>
</tr>
</tbody>
</table>

Table 1: Hyperparameters

## 3. Methodology

### 3.1. Data

We use Universal Dependency treebanks version 2.12 (Nivre et al., 2020; de Marneffe et al., 2021), more specifically the English Web Treebank (EWT; Bies et al., 2012) and the Georgetown University Multilayer Corpus (GUM; Zeldes, 2017). EWT consists of five domains, and GUM consists of eleven domains.

From the sixteen domains of EWT and GUM, we select only the ten domains that each have a minimum of 1000 sentences, to limit negative effects during training due to different data sizes across domains. This includes all five of the EWT domains: answers, email, newsgroup, reviews, weblogs; and five from GUM: conversation, fiction, interviews, vlog and who. We then randomly sub-sample only 1000 sentences from each domain to create a balanced data set.

All of our experiments use ten-fold cross validation, where, for each fold, each domain is split into 800 train, 100 dev, and 100 test sentences. Consequently, when training each domain-expert parser, there are a total of 8000 train sentences (800 in-domain and 7200 out-of-domain), and 100 dev and 100 test sentences (all of these in-domain).

### 3.2. Parser

We use the deep biaffine attention neural dependency parser (Dozat and Manning, 2017) in the implementation by van der Goot et al. (2021b), which we have modified to allow for DWBS. When training the parser, we use the default hyperparameters provided by van der Goot et al., with the only exception being that we specify early-stopping patience,
Figure 1: Performance of the DWBS-trained domain-expert parsers on “EWT reviews” (a), “GUM fiction” (b), and averaged over all ten domains (c). X-axis: domain-weight hyperparameter $\mu$; y-axis: parser performance in LAS. Because in our experimental setup we use ten domains of equal size, whenever $\mu = 0.10$, DWBS is equivalent to conventional RBS; therefore, in each chart we highlight the baseline RBS-trained parser in blue, and we highlight the best performing DWBS-trained parser(s) in green.

In terms of batches rather than epochs, because, when DWBS is enabled, epoch duration varies with $\mu$ and it is also subject to random variation (see Section 2.2). Batch size, on the other hand, is a fixed hyperparameter. All hyperparameters are reported in Table 1.

For each domain, and for each of the ten data folds, we use the dev sentences to determine when to stop training, and we then use the test sentences to evaluate. We evaluate using the scorer from the CoNLL 2018 shared task (Zeman et al., 2018).

### 4. Results

In order to evaluate the effectiveness of DWBS, we perform experiments in which we compare a baseline model trained using conventional RBS against domain-expert parsers trained using DWBS. For each domain, we train domain-expert parsers, with the domain-weight hyperparameter $\mu$ ranging from 0.00 to 1.00 (inclusive), with a step size of 0.05. Remember that $\mu = 0.00$ means that each batch will be sampled exclusively from the out-of-domain par-

<table>
<thead>
<tr>
<th>TB</th>
<th>Domain</th>
<th>$\mu$</th>
<th>LAS R</th>
<th>LAS DW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EWT Answers</td>
<td>0.35</td>
<td>86.78</td>
<td>87.56</td>
</tr>
<tr>
<td></td>
<td>Email</td>
<td>0.35</td>
<td>86.70</td>
<td>88.00</td>
</tr>
<tr>
<td></td>
<td>Newsgr.</td>
<td>0.40</td>
<td>88.64</td>
<td>89.44</td>
</tr>
<tr>
<td></td>
<td>Reviews</td>
<td>0.35</td>
<td>88.27</td>
<td>88.74</td>
</tr>
<tr>
<td></td>
<td>Weblog</td>
<td>0.25</td>
<td>89.52</td>
<td>90.56</td>
</tr>
<tr>
<td></td>
<td>GUM Convers.</td>
<td>0.35</td>
<td>85.41</td>
<td>86.64</td>
</tr>
<tr>
<td></td>
<td>Fiction</td>
<td>0.45</td>
<td>89.86</td>
<td>91.23</td>
</tr>
<tr>
<td></td>
<td>Interv.</td>
<td>0.50</td>
<td>88.08</td>
<td>89.14</td>
</tr>
<tr>
<td></td>
<td>Vlog</td>
<td>0.60</td>
<td>87.74</td>
<td>88.57</td>
</tr>
<tr>
<td></td>
<td>Whow</td>
<td>0.35</td>
<td>90.46</td>
<td>91.11</td>
</tr>
</tbody>
</table>

Table 2: Performance in LAS per domain, comparing the baseline parser (trained using RBS) to the highest-LAS-producing domain-expert parser (trained using DWBS). LAS R: baseline parser trained using RBS; LAS DW: highest-LAS-producing domain-expert parser trained using DWBS; $\mu$: setting resulting in the highest LAS for the given domain. Improvements of more than 1.00 LAS are bolded.
Table 3: Training duration per domain measured in number of thousands of samples until model convergence, comparing the baseline parser to the highest-LAS-producing domain-expert parser. NSC: number of thousands of training samples until model convergence; RBS NSC: NSC for the baseline parser trained using RBS; DWBS NSC: NSC for the highest-LAS-producing domain-expert parser trained using DWBS; $\mu$: setting yielding the best (in terms of LAS) domain-expert parser for the given domain.

<table>
<thead>
<tr>
<th>Treeb. Domain</th>
<th>$\mu$</th>
<th>RBS NSC</th>
<th>DWBS NSC</th>
<th>$\Delta$NSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>EWT Answers</td>
<td>0.35</td>
<td>40.40</td>
<td>40.88</td>
<td>0.48</td>
</tr>
<tr>
<td>Email</td>
<td>0.35</td>
<td>39.84</td>
<td>40.00</td>
<td>0.16</td>
</tr>
<tr>
<td>Newsgroup</td>
<td>0.40</td>
<td>45.60</td>
<td>45.44</td>
<td>-0.16</td>
</tr>
<tr>
<td>Reviews</td>
<td>0.35</td>
<td>40.96</td>
<td>41.36</td>
<td>0.40</td>
</tr>
<tr>
<td>Weblog</td>
<td>0.25</td>
<td>47.04</td>
<td>48.00</td>
<td>0.96</td>
</tr>
<tr>
<td>GUM Conversation</td>
<td>0.35</td>
<td>45.52</td>
<td>41.20</td>
<td>-4.32</td>
</tr>
<tr>
<td>Fiction</td>
<td>0.45</td>
<td>40.56</td>
<td>42.96</td>
<td>2.40</td>
</tr>
<tr>
<td>Interview</td>
<td>0.50</td>
<td>45.20</td>
<td>42.00</td>
<td>-3.20</td>
</tr>
<tr>
<td>Vlog</td>
<td>0.60</td>
<td>48.16</td>
<td>42.96</td>
<td>-5.20</td>
</tr>
<tr>
<td>Whow</td>
<td>0.35</td>
<td>40.40</td>
<td>40.24</td>
<td>-0.16</td>
</tr>
</tbody>
</table>

4.1. Effect on Parsing Accuracy

The DWBS-trained parser outperforms the baseline in all ten domains tested, for some settings of $\mu$. We provide full results for two domains, plus the results averaged over all ten domains, in Figure 1; full results for the remaining domains are supplied in Appendix A. Table 2 summarizes the results by giving the LAS for the highest performing DWBS-trained parser, per domain, and giving the setting for $\mu$ that produced the parser.

The domain which benefits least from DWBS, in terms of absolute increase in LAS over the baseline, is EWT reviews, for which the best setting of $\mu = 0.35$ yields an improvement of 0.47 LAS (see Figure 1a); the domain benefiting most is GUM fiction, for which the best setting of $\mu = 0.45$ gives an improvement of 1.37 LAS (see Figure 1b). The average improvement across all ten domains, using each domain’s best setting of $\mu$, is 0.95 LAS. As shown in Table 2, five domains experience gains of more than 1.00 LAS.

Overall, the best setting of $\mu$ ranges between 0.25 (EWT weblog) and 0.60 (GUM vlog). GUM domains tend to prefer higher values of $\mu$. In other words, those domains profit more from training examples from the same domain, which is an indication that each of those domains is different from all others, either in terms of syntactic structure or annotation.

4.2. Effect on Training Duration

Our hypothesis wrt training times is that the more target-domain sentences that are included in training batches, the faster the parser should converge, since the training sentences should be more consistent and also more similar to the dev data. This hypothesis is supported by findings that alternative batch sampling techniques to RBS which are similarly motivated to DWBS yield significantly faster network training times on several tasks (Loshchilov and Hutter, 2016).

We show the average number of training examples until model convergence for the highest-LAS-producing $\mu$ per domain in Table 3. In contrast to the results presented in the previous subsection in which all ten domains show an improvement in LAS, the domains are evenly split on training time reduction with five seeing a reduction and five experiencing an increase. The greatest increase is experienced by the GUM fiction domain, which requires 2400 more sentences than the baseline to achieve parser convergence, while the greatest decrease is experienced by the GUM vlog domain, which shows a decrease of 5200 sentences until convergence. The average change in training samples is a decrease of 864 sentences. The high variability of differences in training duration suggests that DWBS does not reliably reduce the number of samples required to achieve parser convergence. This may suggest that our target domain data do not always have high internal consistency, which is in line with findings by Zeldes and Schneider (2023), who observed considerable differences in cross-domain parsing between EWT and GUM.

Interestingly, four out of the five domains showing decreased training times are GUM domains. Since GUM domains also prefer higher values of $\mu$, this could suggest that sampling more target sentences reduces training time.
5. Conclusion

In this work we investigated the effectiveness of domain-weighted batch sampling (DWBS) when training a neural dependency parser. DWBS is a technique for constructing training batches that can be used in cases when the domain that a parser will be evaluated on is known and there is also training data available in the evaluation domain. We conducted experiments using ten English domains and found that DWBS produced higher performing parsers than RBS in all ten domains. This finding suggests that when the preconditions for performing DWBS are met, it should be preferred to RBS when training a neural dependency parser.

The success of DWBS for neural dependency parsing suggests several directions for future work: In the present experiment while training each model, the domain-weight parameter $\mu$ was held constant for the full duration of training. An alternative is to begin training with $\mu$ equal to the baseline setting, and then gradually increase $\mu$ as training progresses. This will simulate gradually fine-tuning the parser in the target domain. A second area of future work is to experiment with methods of automatically classifying domains (e.g., in the style of Mukherjee et al., 2017; Mukherjee and Kübler, 2017), which would allow for the discovery of more syntactically useful domain groupings. Finally, we will investigate the effectiveness of domain embeddings (van der Goot and de Lhoneux, 2021; van der Goot et al., 2021a; Li et al., 2019, 2020), an alternative approach to domain adaptation in dependency parsing that can be combined with domain-weighted batch sampling.

6. Acknowledgments

The authors acknowledge the Indiana University Pervasive Technology Institute for providing supercomputing and storage resources that have contributed to the research results reported within this paper.

This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute.

This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract #2022-22072200002. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

7. Bibliographical References


Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021b. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics:


8. Language Resource References


A. Complete Parsing Results

(a) Performance of “EWT answers” parsers

(b) Performance of “EWT email” parsers

(c) Performance of “EWT newsgroup” parsers

(d) Performance of “EWT reviews” parsers

(e) Performance of “EWT weblog” parsers

Figure 2: Parser performance in the five English Web Treebank domains. X-axis: domain-weight hyperparameter $\mu$; y-axis: parser performance (LAS). Baseline RBS-trained parser in blue, and best performing DWBS-trained parser in green.
Figure 3: Parser performance in the five Georgetown University Multilayer Corpus domains. X-axis: domain-weight hyperparameter $\mu$; y-axis: parser performance (LAS). Baseline RBS-trained parser in blue, and best performing DWBS-trained parser in green.
Strategies for the Annotation of Pronominalised Locatives in Turkic Universal Dependency Treebanks

Jonathan Washington\textsuperscript{1}, Çağıri Çöltekin\textsuperscript{2}, Furkan Akkurt\textsuperscript{3}, Bermet Chontaeva\textsuperscript{4}, Soudabeh Eslami\textsuperscript{2}, Gulnura Jumalieva\textsuperscript{5}, Aida Kasieva\textsuperscript{6}, Aslı Kuzgun\textsuperscript{7}, Büşra Marşan\textsuperscript{8}, Chihiro Taguchi\textsuperscript{9}

\textsuperscript{1}Swarthmore College, \textsuperscript{2}University of Tübingen, \textsuperscript{3}Boğaziçi University, \textsuperscript{4}Kyrgyz-Turkish Manas University, \textsuperscript{5}Stanford University, \textsuperscript{6}University of Notre Dame, jwashin1@swarthmore.edu, cagri.coeltekin@uni-tuebingen.de, furkan.akkurt@bogazici.edu.tr, {bermet.chontaeva,soudabeh.eslami}@student.uni-tuebingen.de, {gulnur.jumalieva, aida.kasieva}@manas.edu.kg, kuzgunasli@gmail.com, busra@stanford.edu, ctaguchi@nd.edu

Abstract

As part of our efforts to develop unified Universal Dependencies (UD) guidelines for Turkic languages, we evaluate multiple approaches to a difficult morphosyntactic phenomenon, pronominal locative expressions formed by a suffix -ki. These forms result in multiple syntactic words, with potentially conflicting morphological features, and participating in different dependency relations. We describe multiple approaches to the problem in current (and upcoming) Turkic UD treebanks, and show that none of them offers a solution that satisfies a number of constraints we consider (including constraints imposed by UD guidelines). This calls for a compromise with the ‘least damage’ that should be adopted by most, if not all, Turkic treebanks. Our discussion of the phenomenon and various annotation approaches may also help treebanking efforts for other languages or language families with similar constructions.

Keywords: Turkic languages, Universal Dependencies, treebanks

1. Introduction

As the number of treebanks for a single language or a language family in the Universal Dependencies (UD) repository\textsuperscript{1} grows, consistent annotations become a concern (Gamba and Zeman, 2023a,b; Zeldes and Schneider, 2023). We report on one issue that is part of ongoing efforts to unify Universal Dependencies (UD) treebanks for Turkic languages, currently numbering at 16 in 8 different UD languages. Issues regarding the consistency of UD annotation of Turkic languages have been reported in earlier studies (Tyers et al., 2017; Türk et al., 2019; Çöltekin et al., 2022), with the main consensus being the need for more unified and consistent annotations across treebanks.

In this paper, we examine one selected issue in depth—namely, that of -ki, which attaches to nouns in the genitive and locative case. With locative nouns, it forms either attributive expressions or pronominals, while with genitive nouns, the result is always a pronominal expression.\textsuperscript{2} As explained in detail in §2, how to appropriately annotate these pronominal forms is unclear and problematic with the present UD guidelines. As a result, the current Turkic treebanks adopt different approaches to annotating this construction. Divergence also exists within different treebanks of the same language.

We believe that the discussion of this linguistic phenomenon is likely to increase the consistency of current treebanks, help researchers creating new treebanks for Turkic languages (and others facing similar issues), and may result in improvements to the general UD guidelines by highlighting issues that are not well addressed in the current guidelines.

In this paper, we provide background information on the issue of pronominalised locatives (§2), discuss in depth several possibilities for the annotation of pronominalised locatives in Turkish languages (§3), summarise these approaches (§4), and conclude (§5). While a recommendation for a preferred approach is not put forth, a potential compromise is identified.

2. The issue of pronominalised locatives

In Turkic languages, locative forms of nominals (e.g., nouns, pronouns, and proper nouns) function as a locative adjunct/modifier to the head of an embedded or root clause, as in Figure 1.

Locatives cannot modify nouns on their own. One common strategy to use locatives attributively as a modifier to a noun is with the addition of the

\textsuperscript{1}See Appendix A for information on current and upcoming Turkic UD treebanks.

\textsuperscript{2}Here, we only focus on the more varied, locative version. The outcome of the present discussion is likely to inform the issue of the annotation of genitives as well.
Figure 1: A sentence containing an attributive locative; English translation: “Children slept in the room.”

Figure 2: A sentence containing an attributive locative; English translation: “The children in the room fell asleep.”

When a locative is used attributively in this way, we opt to annotate it as nmod or nmod:loc since it is a nominal dependent (with a noun POS and lemma) of a nominal, just as in the semantically equivalent English sentence. A disadvantage of this approach is that the Case feature remains Loc and the -ki morpheme is not treated separately. However, the structure is recoverable, as these constructions are unique (in each language where it occurs) as the only time a locative nmod dependent is found.

As with other attributive expressions in Turkic languages—including adjectives per Krejci and Glass (2015) and verbal adjectives per Washington et al. (2022)—these attributive locative expressions may be used nominally, as a sort of pronominal.

By ‘pronominal’, we mean that the resulting form is not a nominal but stands in for one. For example, in Turkish büyüklere beşenmedim ‘I liked the big ones’, the derived form of the adjective büyük ‘big’ has nominal morphology and refers to a hypothetical interpretation: (1) ‘the ones in the big room’ (byüyk ‘big’ modifying oda ‘room’), which is the correct interpretation, and (2) ‘the big ones in the room’ (byüyk modifying odakiller ‘the ones in the room’) is not a possible interpretation. Any solution to annotation that considers the word as a single syntactic unit cannot

---

3In many Turkic languages this has phonologically reduced, e.g. to -kl (Azerbaijani) or -G (Kyrgyz, Tatar).
4Turkish, Azerbaijani, Kyrgyz, and Tatar are presented as they are the Turkic languages whose UD annotation is currently being considered by the authors.
5These two approaches are both acceptable in our opinion, although the latter is more specific and may make identification of this construction easier, for example in an information extraction task.
6By ‘pronominal’, we mean that the resulting form is not a nominal but stands in for one. For example, in Turkish büyüklere beşenmedim ‘I liked the big ones’, the derived form of the adjective büyük ‘big’ has nominal morphology and refers to an unmentioned nominal. See Göksel and Kerslake (2005, p.246) for a detailed discussion.
7I.e., this is a productive process that occurs in the syntax. This is not to be confused with lexical derivation, which is a historical and often not fully productive process and is usually opaque to syntax. Multiple opinions exist as to the specific mechanism by which this pronominalisation operates: through ellipsis of a nominal head, through a null-headed DP, through syntactic transformations, or otherwise.
distinguish these syntactic dependencies. Moreover, such an annotation strategy implies the latter structure, where *bûyûk* modifies the entire token *odadakiler*.

In an ideal solution to annotation, all morphological and syntactic information about the two participants would be recoverable.

To further complicate matters, the *-ki* morpheme can be attached to the same word multiple times. Although forms with multiple *-ki* morphemes can be difficult to interpret and rare in real-world usage, there is no principled limit for the number of *-ki* morphemes that can be attached to a noun. For example, to refer to ‘glasses in the cupboard in the room’, we could use the Turkish expression *oda-da-ki*-*nde-ki*-ler ‘the ones in the one in the room’. Except cognitive load, there is nothing stopping a speaker to add another *-de-ki* to refer to the drinks inside the glasses. Although we will limit our discussion to forms with a single *-ki* morpheme, the ideal solution should also work well for words with multiple occurrences of the morpheme.

In summary, considering the pronominal forms created with the morpheme *-ki* as single syntactic words results in two major issues (see Çöltekin, 2016, for an earlier discussion):

- It violates the lexical integrity principle (Haspelmath and Sims, 2010, p.203) since the syntactic dependencies refer to parts of words.
- It also results in conflicting morphological features. For example, in the example in Figure 3, ‘room’ is singular, while the resulting pronominal refers to multiple people in the room.

The following sections discuss various ways we see as possible approaches to annotating these nominalised constructions in UD.

### 3. Possible Approaches

Here we demonstrate four possible approaches to the annotation of pronominalised locative forms and discuss advantages and disadvantages of each: keeping a single token (3.1), using layered features (3.2), splitting the token before *-ki* (3.3), and splitting the token after *-ki* (3.4).

We will use the Turkish sentence *Bardak dolabindakilerim düştüler* ‘The ones of mine on the cup cabinet fell’ to illustrate how different approaches handle these forms.

The pronominal in this sentence refers to a group of items, e.g., glasses, papers, etc. This example was chosen because there are different number, case, and possession features morphologically indicated for each of the two referents of the pronominalised locative token (the referent of the noun it is formed around and the referent of the pronominal it comprises). An alternative version of this sentence with an independent noun modified by a *-ki* bearing form is provided with annotation in Figure 4 for reference.

#### 3.1. No segmentation

The first option is to have no segmentation of the word *dolabindakilerim* ‘the ones of mine on its cabinet’, as presented in Figure 5.

The advantage of this choice is practical: sub-word segmentation is a non-trivial task, and avoiding it will help make automated segmentation more precise, especially in low-resource settings. On the other hand, it is not clear what values to assign to the Number, Person[psor], or Person categories, since the values for both referents of the token *dolabindakilerim* are present: the noun is singular, locative, and has a third-person possessor, while the resulting pronominal is plural, nominative, and has a first-person (plural) possessor. This choice additionally fails to capture several aspects of the dependencies in this sentence:

- that there are two referents of the form: a noun and a pronominal;
- that there is a relationship between the form’s two referents;
- that the first noun token in the sentence is a possessor *nmod* of the form’s first referent (the noun) and not the second (the pronominal); and
- that the second referent of the form (the pronominal) and not the first (the noun) is the *nsubj* of the root.

Current treebanks employing a no-segmentation approach in Turkish assume an analysis of elision and use the concept of promotion (whereby a normally dependent function word is ‘promoted’ to the syntactic function that an elided head would normally have) to annotate dependencies. In our example *oda-da-ki*-ler ‘the ones in the room’, this approach considers the head word *çocuk*-ler ‘children’ to be elided. Hence, its dependent *odadaki* is promoted to

---

8All other combinations are also possible in other contexts; for example, *dolaplarnindakiler* (Definite) is *dolaplann-dakim*, or *dolabindakim*.

9I.e., Penn (Cesur et al., 2023a), KeNet (Kuzgun et al., 2023b), FrameNet (Cesur et al., 2023b), Tourism (Kuzgun et al., 2023a), Atis (Köse and Yıldız, 2023).

10Per https://universaldependencies.org/u/overview/syntax.html.

11Unlike the English translation where the pronoun *one* still occupies the head of the construction.
According to this approach as taken in these treebanks, the -ki bearing form in the example in Figure 5 is an nsubj dependent of the verb.

Using a Case=Loc feature (as opposed to Case=Nom) with, for example, an nsubj dependent could clarify that this pronominal has some special status. However, a naïve downstream interpretation may understand this to be, in this example, an oblique (locative-marked) subject as opposed to a pronominal locative, especially given that the lemma is that of the attributive word (here, dolap ‘cabinet’) as opposed to the referent to which the morphology and head dependency refer (here, the pronominal referring to e.g., bardak ‘cup’). Therefore, one option is to use the orphan tag when the -ki word is pronominal, shown as an option in Figure 5. The orphan relation is traditionally used in cases of head ellipsis where there is a remnant nominal that must attach to a head that it would not normally attach to. This approach solves the issue with misreading annotations; however, the orphan analysis is not informative. Furthermore, the issues with multiple Number, Person[psor], and Case features that need to be assigned to the form odadakiler remain.

Another option is to introduce a new case feature for attributive and pronominal locative, such as AttrLoc. In pronominal uses, as shown in Figure 6, it would then be clear that this structure is not, for example, an oblique subject form of the lemma, but a pronominalised form of an attributive locative formed around the lemma. This at first appears to solve the problem having multiple case features, but it is still not clear how to annotate the second case feature (which can be any of the cases available in a given Turkic language). The problems of multiple number features and possessor person features also remain.

### 3.2. Layered features

An approach that would allow for annotation of different morphological features for the two referents of a pronominalised locative token is to use layered features.

While not currently used in this way in UD, layered features enable us to annotate more than one value on a feature key. Some Turkic treebanks have already employed layered features to annotate possessive marker on a nominal (cf. dolab-in-da-ki and bardak-lar-im in Figure 4, where psor in the brackets specifies that the Person key refers to the Person feature of the possessor). By extending their usage, it is possible to use layered features to specify which stem a feature key is referring to. The application of this approach on the example sentence is shown in Figure 7.

Advantages of this approach are that (i) we can annotate multiple features sharing the same key without splitting the word, (ii) layers can be recursively applied, (iii) layered features can be applied to languages without a derivational morpheme like -ki (e.g., some Tungusic, Quechuan, and Dargin languages), and (iv) it is compatible with the hierarchical annotation of morphology in UniMorph 4.0 (Batsuren et al., 2022).

This approach, however, fails to solve the dependency relation issues presented by having a single token: it is not clear which subword token is
the ‘head’, and which is the actual referent of the external dependency relation. There is also still only one POS.

In summary, there is a strong indication that the pronominal formed by -ki contains multiple syntactic words.

3.3. Splitting before -ki

Segmentation of the pronominalised forms solves the problems with conflicting features and dependencies, as well as the non-informativeness of the orphan relation. We consider two different ways (or locations) for segmenting these forms. The first option (Figure 8), which is used in some of the current treebanks (e.g., Türk et al., 2019; Marşan et al., 2022), considers the -ki morpheme as part of the second token.

This approach allows retaining all linguistic information packed in the -ki bearing forms:

- The first subword can be annotated as taking part in other syntactic phenomena, such as compounding, independently of the full token. In our example here, the compound bardak dolabı is independent of (although a part of) the pronominal that is formed with -ki. Splitting the -ki bearing form into subwords allows illustrating such constructions more clearly.

In addition to enabling annotation of all morphological features and dependency relations, splitting before -ki prevents ending up with null morphemes (discussed in detail in §3.4). There are two disadvantages to this approach. Firstly, the current UD guidelines are not very supportive of subword tokenization, so this approach diverges from the UD framework to some extent. Secondly, due to the additional complexity, this approach can introduce noise or learnability issues for less sophisticated systems like shallow parsers.

3.4. Splitting after -ki

An alternative segmentation approach segments pronominalised locatives after -ki, as shown in Figure 9.

When splitting before -ki, the -ki morpheme is considered part of the pronominal ‘word’ (i.e., the part of the token representing the second referent). This can be viewed as inconsistent with the attributive use of -ki, where—regardless of whether or not -ki is best treated as an independent token—it is clear that -ki is not the lemma to which the
second set of morphological features belong. For example, in the annotation of the attributive use of 
-ki in Figure 4, the noun head of the -ki bearing form has the lemma bardak. However, in the an-
notation of an equivalent sentence with that noun absent and its morphology instead associated with
the -ki bearing form, such as that in Figure 8, the pronoun head of the second referent (which could
still be understood to refer to bardak), is now -ki according to the split-before approach. In other
words, the -ki is associated with a different token in these two examples—and more broadly, in these
two constructions: in an attributive construction, -ki is associated with the first participant, and in
an equivalent pronominal construction, -ki is associated with the second participant.

The approach of splitting after -ki, then, is a way to avoid what might be seen as an inconsistency
that arises when splitting before -ki. By segmenting pronominalised locatives immediately after -ki,
the -ki morpheme remains with the first of the two tokens (the dependent and not the head) whether
attributive or pronominal. This also unifies these two uses of -ki as a single phenomenon, with the
addition of the phenomenon that allows the head noun to be absent in pronominal -ki forms.

A major problem with this approach is that it requires an empty lemma, as well as an empty form
when there are no additional affixes after -ki. Empty lemmas and forms are not allowed ac-
cording to UD v2 annotation guidelines. While it would be possible not to annotate a second
token (the pronoun / second referent) if it were empty, that would reduce the consistency of this
approach, and still leaves the issue of having an empty lemma. Furthermore, as with segmenting
before -ki, there may be limitations for less sophis-
ticated automated annotation systems, although it is possible that systems capable of segmenting
words into subword units would be able to handle one approach more easily than the other—an
area for future investigation. Lastly, treating attrib-
tutive and pronominal locatives uniformly may
go against a generative syntax analysis of these
two uses, where the attributive locative form is an
ordinary member of the phrase (DP) containing
the head noun, whereas the pronominal locative
is cast directly into a DP with the accompanying
morphology and has fewer layers between the two
phrases.

3.5. Splitting after -ki with fallback

One problem with splitting after -ki is that null
nodes would result in situations where there is no
inflection, as in the sentence Bardak dolabindaki
düştü ‘The one on the cup cabinet fell’. This prob-
lem could be avoided with a fallback in such cases.

One option is to fall back to an orphan analy-
sis, per Figure 10, signalling to downstream tasks
that information is missing (specifically an elided
[pronominal] element). Using the orphan relation
has the disadvantages discussed in §3.1: it is
not informative, and does not allow for annota-
tion of multiple relations (although implies them)
or multiple sets of features. However, examples of
pronominalised locatives are not very frequent in
existing corpora, and examples of pronominalised
locatives with no further inflection are quite rare,
so this approach would not result in excessive use
of the orphan relation.

To include the elided information, enhanced de-
pendencies may be used, as in Figure 11. En-
hanced dependencies are explicitly designed to present null nodes in cases of elision.\footnote{Per \url{https://universaldependencies.org/v2/enhanced.html}.} Use of enhanced dependencies has some drawbacks. If annotated even for just one example, the entire corpus needs to have enhanced dependencies annotated. Furthermore, most parsers, querying tools, and other applications of UD lack support for enhanced dependencies and ignore them. However, this approach does preserve the information lost in the accompanying standard dependency analysis.

4. Summary of approaches

The approaches described in Section 3, and their advantages and disadvantages are summarised in Table 1.

The first approach discussed, no-segmentation (§3.1), has the benefit of ease of tokenization. Even though state-of-the-art parsers may be successful in segmenting words into subword units, not having to split words has a clear advantage, especially in low-resource scenarios.\footnote{Empty forms and lemmas would only occur in enhanced dependencies annotation, where they are permissible.} It also avoids empty word forms and empty lemmas that some of the approaches postulate. However, it fails to represent multiple sets of morphological features, and it does not allow a correct interpretation of the dependency relations the word participates in. Specifically, annotating in this way results in a situation where it is unclear which part of the word’s referents is the modifier of another head. Possible ways to remove the ambiguity would be to use the orphan relation (second row of Table 1) or an AttrLoc value for the case feature (third row of Table 1), both of which allow for differentiation of pronominalised locatives from other dependents with a similar relation to their head. However, orphan does not include any information regarding the syntactic function of the word in the sentence. With or without the orphan relation or an AttrLoc case feature, the no-segmentation approach does not resolve the issue of multiple, potentially conflicting sets of morphological features assigned to a single syntactic word.

A possible solution (described in §3.2) that allows expressing multiple sets of morphological features is to make use of layered features as exemplified in Figure 7. Although this uses the UD layered features in an unorthodox way,\footnote{E.g., introducing multi-dimensional layers, and layers indexed by ordinals.} it enables specification of multiple sets of morphological features, and, with the use of the orphan relation, pronominalised locatives can also be differentiated from other dependents with a similar relation to their head. However, as noted earlier, it does not allow identifying the dependency relations correctly. It still leaves it unclear which part of the word is modified by a modifier, and which part is a modifier to another head. Another downside is, perhaps, the complexity: such feature sets and relations are likely to be difficult to learn for parsers, and the treebank queries for relevant features/structures are likely to be misled or miss the relevant items due to the idiosyncratic nature of the annotations.

Both segmentation options resolve the main concerns with the pronominal construction: the appropriate features are easily assigned to each syntactic word, and the dependents can modify the correct syntactic word without ambiguity. The relation between the pronominal and its head is also clearer. The disadvantage of splitting before -\(k\i\) (§3.3) is the inconsistency with the attributive use. This approach suggests either splitting -\(k\i\) in attributive usage without any clear motivation—in which case it is still not the lemma of the modified noun’s morphological features as in the pronominal treatment—or treating attributive and pronominal cases differently.\footnote{We intend to investigate this empirical question in future research.} The disadvantage of splitting after -\(k\i\) (§3.4) is the introduction of empty lemmas, and empty forms when no further affixes are attached after -\(k\i\). Since empty forms are not allowed in the current basic UD dependencies, this approach would require a substantial modification to the UD guidelines. Splitting after -\(k\i\) with fallback (§3.5) solves the issue of empty lemmas but requires the use of enhanced dependencies.
Figure 11: Possible analysis segmenting after -ki with no morphology, with enhanced dependencies fallback.

<table>
<thead>
<tr>
<th>Approach</th>
<th>No empty forms</th>
<th>No empty lemmas</th>
<th>2 sets of features</th>
<th>Deprels for 2 referents</th>
<th>Consistent with attributive use</th>
<th>Easy querying</th>
<th>No need for subword-aware parser</th>
</tr>
</thead>
<tbody>
<tr>
<td>No-segmentation</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✔</td>
</tr>
<tr>
<td>orphan relation</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✔</td>
</tr>
<tr>
<td>AttrLoc feature</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✔</td>
</tr>
<tr>
<td>Layered features</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✔</td>
</tr>
<tr>
<td>Splitting before -ki</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✘</td>
<td>✓</td>
<td>✘</td>
</tr>
<tr>
<td>Splitting after -ki</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✘</td>
<td>✓</td>
<td>✘</td>
</tr>
<tr>
<td>Splitting after, enhanced</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✘</td>
</tr>
<tr>
<td>dependencies fallback</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✘</td>
</tr>
</tbody>
</table>

Table 1: A summary of the advantages and disadvantages in the discussed approaches.

5. Concluding remarks

The authors currently consider splitting pronominalised locatives before -ki a best compromise, and recommend this for annotation of Turkic tree-banks, although with a caveat.

While the authors agree with one another that segmentation is needed to properly capture these constructions, opinions differ as to which approach is ideal. Proponents of splitting the pronominalised locative before -ki do not believe that it is a problem for the approach to be inconsistent with the treatment of the attributive locative due to a generative syntax view that they are in fact distinct. Proponents of splitting the pronominalised locative after -ki realise that it would take a major change to current UD guidelines for this approach to be viable, and while finding splitting before -ki somewhat unsatisfactory, accept that it may be the current best compromise.

The issue of pronominalised locatives is just one of many specific issues where consistent UD annotation guidelines are needed for Turkic languages. This issue is also relevant to the UD (and UniDive) community at large. By bringing awareness to this issue and discussing it in depth, we hope that new annotation projects for languages with similar phenomena will be eased, and that our efforts will lead to improved overall quality of corpora and annotation guidelines.

6. Acknowledgements

We are grateful to Gülnara Karasawa for her help with creating the Tatar dataset for this study.

We thank UniDive, the COST Action CA21167, and Istanbul Technical University for supporting the organization of the workshop in September 2023 that made the present work and future collaboration towards a unified annotation of Turkic UD treebanks possible. We also thank all participants of the workshop for fruitful discussion and suggestions.

7. Bibliographical References


Aibek Makazhanov, Aitolkyn Sultangazina, Olzhas Makhambetov, and Zhandsor Yessenbayev. 2015b. Syntactic annotation of Kazakh:


8. Language Resource References


A. UD Turkic Treebanks

There are currently UD treebanks for Kazakh, Kyrgyz, Tatar, Turkish, Uyghur, Yakut, and Old Turkish, and a treebank annotating sentences with Turkish-German code switching. All languages except Turkish are represented with a single treebank, while Turkish has 9 treebanks. Table 2 lists the treebanks currently released in the UD repositories as of UD version 2.13.

<table>
<thead>
<tr>
<th>Language</th>
<th>Treebank</th>
<th>Stats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kazakh</td>
<td>KTB</td>
<td>(Tyers and Washington, 2015; Makazhanov et al., 2015a) (Makazhanov et al., 2023)</td>
</tr>
<tr>
<td>Kyrgyz</td>
<td>KTMU</td>
<td>(Benli, 2023)</td>
</tr>
<tr>
<td>Old Turkish</td>
<td>Tonqq</td>
<td>(Derin and Harada, 2021)</td>
</tr>
<tr>
<td>Tatar</td>
<td>NMCTT</td>
<td>(Taguchi, 2023)</td>
</tr>
<tr>
<td>Turkish</td>
<td>Aliis</td>
<td>(Köve and Yıldız, 2023)</td>
</tr>
<tr>
<td>Turkish</td>
<td>BOUN</td>
<td>(Türk et al., 2022; Marşan et al., 2022) (Marşan et al., 2023)</td>
</tr>
<tr>
<td>Turkish</td>
<td>FrameNet</td>
<td>(Cesur et al., 2023b)</td>
</tr>
<tr>
<td>Turkish</td>
<td>GB</td>
<td>(Çöltekin, 2015) (Çöltekin, 2023)</td>
</tr>
<tr>
<td>Turkish</td>
<td>IMST</td>
<td>(Sülbacak et al., 2016) (Türk et al., 2023)</td>
</tr>
<tr>
<td>Turkish</td>
<td>Kenet</td>
<td>(Kuzgun et al., 2023b)</td>
</tr>
<tr>
<td>Turkish</td>
<td>Penn</td>
<td>(Cesur et al., 2023a)</td>
</tr>
<tr>
<td>Turkish</td>
<td>PUD</td>
<td>(Zeman et al., 2017) (Újvárosi et al., 2023)</td>
</tr>
<tr>
<td>Turkish</td>
<td>Tourism</td>
<td>(Kuzgun et al., 2023a)</td>
</tr>
<tr>
<td>Turkish</td>
<td>German/SAGT</td>
<td>(Çetinoğlu and Çöltekin, 2022) (Çetinoğlu and Çöltekin, 2023)</td>
</tr>
<tr>
<td>Uyghur</td>
<td>UDT</td>
<td>(Eli et al., 2016) (Eli et al., 2023)</td>
</tr>
<tr>
<td>Yakut</td>
<td>YKTDT</td>
<td>(Merzhevich and Ferraz Gerardi, 2022) (Merzhevich and Gerardi, 2023)</td>
</tr>
</tbody>
</table>

Table 2: Basic statistics on current UD treebanks (as of UD version 2.13). *sent*: number of sentences, *tok*: number of tokens, *multi*: number of multi-word tokens, *types*: number of word types, *ltypes*: number of lemma types, *pos*: number of POS tags used, *rel*: number of dependency relations used (including language/treebank specific relations), *feat*: number of morphological features used.

Besides existing treebanks, the UD web page also reports Uzbek, Ottoman Turkish and yet another Turkish treebank in preparation. We are also aware of new treebanks in preparation for Kyrgyz (Kasieva et al., 2023), Azerbaijani and Kumyk.
BERT-based Idiom Identification using Language Translation and Word Cohesion

Arnav Yayavaram*, Siddharth Yayavaram*, Prajna Upadhyay, Apurba Das
BITS Pilani Hyderabad, India
{f20213117, f20213116, prajna.u, apurba}@hyderabad.bits-pilani.ac.in

Abstract
An idiom refers to a special type of multi-word expression whose meaning is figurative and cannot be deduced from the literal interpretation of its components. Idioms are prevalent in almost all languages and text genres, necessitating explicit handling by comprehensive NLP systems. Such phrases are referred to as Potentially Idiomatic Expressions (PIEs) and automatically identifying them in text is a challenging task. In this paper, we propose using a BERT-based model fine-tuned with custom objectives, to improve the accuracy of detecting idioms in text. Our custom loss functions capture two important properties (word cohesion and language translation) to distinguish PIEs from non-PIEs. We conducted several experiments on 7 datasets and showed that incorporating custom objectives while training the model leads to substantial gains. Our models trained using this approach also have better sequence accuracy over DISC, a state-of-the-art PIE detection technique, along with good transfer capabilities. Our code and datasets can be downloaded from https://github.com/siddharthyayavaram/BERT-Based-Idiom-Detection

Keywords: idioms, multi-word expressions, word cohesion, language translation, loss function

1. Introduction
An idiom refers to a special type of multi-word expression (Baldwin and Kim, 2010) whose meaning is figurative and cannot be deduced from the literal interpretation of its components. Idioms often exhibit peculiar behavior by violating selection restrictions or altering the default semantic roles of syntactic categories. Consequently, they pose significant challenges for Natural Language Processing (NLP) systems. Idioms are prevalent in almost all languages and text genres, necessitating explicit handling by comprehensive NLP systems. We refer to these phrases as potentially idiomatic expressions (PIEs) to account for the contextual semantic ambiguity in their expression. Better detection of PIEs can enhance numerous machine translation tasks.

Techniques to automatically detect and identify PIEs need to do many tasks accurately – i) automatically detect if an idiomatic expression is present in a sentence (Briskilal and Subalalitha, 2022; Tan and Jiang, 2021; Liu and Hwa, 2019), ii) if yes, identify the idiomatic tokens (Zeng and Bhat, 2021, 2022). Both of these are challenging tasks. For instance, in the sentence “Oh — for about four years, on and off, he said vaguely”, the potentially idiomatic expression “on and off” is used figuratively, whereas, it is used literally in the sentence “Participate in training, both on and off station”. Existing techniques for idiom detection rely on syntactic patterns, knowing the PIE being classified correctly, and lack generalization. In this paper, we address the above-mentioned problems and show that improvement in i) improves ii) substantially.

We employ a BERT-based fine-tuning approach with custom objectives to improve accuracy on all 3 tasks. We define our objectives in Section 4.2 based on language translation and word cohesion.

Our salient contributions are:
1: Introduction of a language translation-based metric to detect the presence of idioms.
2: A novel loss function to selectively penalize examples using sentence translation and word cohesion that can be used with any architecture for idiom detection.
3: Our models trained with custom loss functions exhibit improved generalization capabilities, evident in identifying unseen PIEs.

2. Related Work
MWE, short for Multi Word Expressions are notable collocations with multiple words, for instance “all at once” or “look something up”. (Baldwin and Kim, 2010; Constant et al., 2017). IEs (Idiomatic Expressions), are a subset of MWEs, which exhibit non-compositionality (Baldwin and Kim, 2010; Fadaee et al., 2018; Liu et al., 2017; Biddle et al., 2020). Metaphors, such as “heart of gold” and “night owl” compare unrelated things implicitly. While some MWEs and IEs use metaphorical figuration, not all metaphors are IEs; they can be direct comparisons with single words (e.g., “I am titanium”). In this paper, we study IEs.
IE Classification broadly falls under two categories – standalone phrase classification and context-based classification. Standalone classification tasks decide if a phrase could be used as an idiom without specifically considering its context (Fazly and Stevenson, 2006; Shutova et al., 2010; Tabossi et al., 2008, 2009; Reddy et al., 2011; Cordeiro et al., 2016). As opposed to context-based idiom classification techniques which take into account the entire sentence to detect the presence of idiom (Peng et al., 2014; Nedumpozhiman et al., 2019; Peng and Feldman, 2017; Tan and Jiang, 2021; Verma and Vuppuluri, 2015; Briskilal and Subalalitha, 2022; Liu and Hwa, 2019). Earliest known context-based phrase classification techniques developed per idiom classifiers, which are not scalable (Liu and Hwa, 2017). Context-based phrase classification techniques can additionally detect which tokens are idiomatic/nonidiomatic (Zeng and Bhat, 2021; Salton et al., 2016; Zeng and Bhat, 2022). Typically, the latter is dependent on the former task – only if an idiom is detected to be present in a sentence, does the classification of idiomatic and non-idiomatic tokens follow. Efforts to build complementary resources to support this task include constructing a knowledge graph (Zeng et al., 2023) and an information retrieval system to search for idiomatic expressions (Hughes et al., 2021).

Detecting idioms in the text has also become popular in non-English languages. In (Itkonen et al., 2022), authors leverage various models provided by HuggingFace in conjunction with the standard BERT model for the idiom detection task in English, Portuguese, and Galician. They emphasize on feature engineering using traits that define idiomatic expressions. These additional features result in enhancements compared to the baseline performance. In (Tedeschi et al., 2022), a multilingual transformer-based model and a dataset of idioms in 10 languages is presented. A rule-based intra-sentential idiom detection system in Hindi was presented in (Priyanka and Sinha, 2014).

3. Problem Statement

We are given the following:

- A sentence $S$ with $n$ tokens $w_1, w_2, \ldots, w_n$, where each $w_i$ represents a tokenized unit. $S$ is an syntactic ordering over $w_i$’s.
- Labels $L = \{I, NI\}$ where $I$ and $NI$ represent <idiom> and <not idiom> (or literal) classes, respectively.

This labelling produces a sequence of class labels $Z = z_1, z_2, \ldots, z_n$ where $z_i = f(w_i)$. The high-level objective of this work is to learn the function $f(\cdot)$

- A successful prediction occurs when an idiomatic subsequence $w_{i:j}$ is identified in $S$, and the corresponding labels $z_{i:j}$ are labelled as $I$. There can be more than one such subsequences.
- If the subsequence $w_{i:j}$ is literal, all corresponding labels $z_{i:j}$ are $NI$.
- If the sentence lacks an idiom, all $z_{1:n}$ are categorized as $NI$.

4. Methodology

4.1. BERT-based Idiom Identification

Figure 1 shows the high-level architecture of our method. Our loss functions are implemented over BERT (Devlin et al., 2018), a pre-trained transformer-based model developed by Google. Due to its effectiveness in capturing context and semantics for various NLP tasks, we re-use its pre-trained architecture for fine-tuning our model using binary cross-entropy loss. Despite its success, cross entropy loss is sensitive to outliers and classimbalance. We observe class imbalance in idiom classification where the label $I$ is far less frequent than label $NI$ leading to poor accuracy for $I$ tokens. To fix this, we propose to use language translation and word cohesion to manipulate the loss. In the following sections, we define two novel loss functions for the task of idiom token classification. The merit of our work lies in the fact that these custom loss functions can be used with any architecture.

4.2. Language Translation and Cohesion for Idioms

4.2.1. Translation-based Loss Function

An important property exhibited by an idiom is the difference between its literal and actual meaning. However, a phrase that is an idiom in language $L_1$ is improbable to be an idiomatic phrase in another language $L_2$. For example, take the English idiom, “raining cats and dogs”, its Hindi translation is “भारी वषार्”, which when translated back to English gives “heavy rain” which is the meaning of our initial idiom but is quite different lexically. Let $S_{L_1}$ denote a sentence containing an idiom in language $L_1$, $S_{L_1} \rightarrow L_2$ a translation of $S_{L_1}$ in $L_2$, and $S_{L_1} \Rightarrow L_2$ a translation of $S_{L_1} \rightarrow L_2$ back to $L_1$. When $S_{L_1}$ is translated to $S_{L_1} \rightarrow L_2$, the idiomatic tokens in $S_{L_1}$ will be expressed through their actual meaning in $S_{L_1} \rightarrow L_2$ because of a lack of corresponding idiom in $L_2$. Re-Translating it to $L_1$ will force the idiom to be expressed with its actual meaning in $S_{L_1} \Rightarrow L_2$. Lexically, the actual meaning of an idiom and the surface form of an idiom differ substantially from each other. We employ this simple trick to detect
the presence of an idiom in a sentence – if \( S_{L_1} \leftrightarrow L_2 \) and \( S_{L_1} \) differ lexically by some margin, \( S \) is likely to contain an idiom. A sentence that contains no idiom is likely to have the same lexical representation in the original and back-translated sentence.

We leverage the METEOR (Banerjee and Lavie, 2005) metric to quantify this observation by computing a score to reflect the lexical and syntactic similarity between the translated and reference sentences. METEOR incorporates a penalty mechanism for longer matches by organizing system translation unigrams mapped to reference translation unigrams into minimal chunks. These chunks consist of adjacent unigrams in the system translation that align with adjacent unigrams in the reference translation. Longer n-grams result in fewer chunks. In the extreme case of a complete match, only one chunk exists, while in the absence of bigram or longer matches, the number of chunks equals the count of unigram matches. An alignment is created between the system translation and the reference translation by mapping unigrams based on different criteria, such as exact match, stemming, or synonymy. The alignment is formed by selecting the most extensive subset of unigram mappings, ensuring that each unigram maps to at most one unigram in the other string. The chosen alignment is the one with the fewest “unigram mapping crosses”, which occur when lines connecting mapped unigrams intersect in a vertical arrangement of the two strings.

Unigram Precision: \( \mathcal{P} = \frac{N_{\text{correct}}}{N_{\text{backtrans}}} \)

Unigram Recall: \( \mathcal{R} = \frac{N_{\text{correct}}}{N_{\text{original}}} \)

Here, \( N_{\text{correct}} \) represents the number of correctly mapped unigrams, \( N_{\text{backtrans}} \) represents the total number of unigrams in the back-translated sentence, and \( N_{\text{original}} \) represents the total number of unigrams in the original sentence.

Harmonic Mean: \( \mathcal{F}_{\text{mean}} = \frac{10 \cdot \mathcal{P} \cdot \mathcal{R}}{\mathcal{R} + 9 \cdot \mathcal{P}} \)

Penalty: \( 0.5 \times \left( \frac{C}{U} \right)^3 \)

where \( C \) represents the number of chunks and \( U \) represents the number of unigrams matched.

Score: \( \mathcal{F}_{\text{mean}} \times (1 - \text{Penalty}) \)

It evaluates the quality of a translation by comparing it to one or more reference translations. METEOR considers various factors such as unigram precision, recall, and alignment errors to compute a score that reflects the lexical and syntactic similarity between the translated and reference sentences. For instance, the sentence “The early morning flight required them to hit the sack much earlier than usual”, is translated into Italian “Il volo mattutino li obbligava a coricarsi molto prima del solito.”, and its back-translation to English “The morning flight forced them to go to bed much earlier than usual.”, the idiomatic usage causes a large syntactic change during back-translation which will lead to a high alignment error term and comparatively lower METEOR score of 0.5919.

During the training of the BERT-based model for idiom recognition, the translation-based loss function incorporates the METEOR score as a penalty term. If the METEOR score falls below a certain threshold, it indicates that the back-translation process has significantly altered the original sentence, which we posit is due to the presence of idiomatic expressions.

\[
\mathcal{L}_{\text{retranslation}} = \mathcal{L}(1 + \lambda_1 \mathbb{I}(M \mathcal{S} < \lambda_2))
\]
where $MS$ is the meteor score for the sentence, $L$ is the original binary cross entropy loss, and $\mathbb{1}(\lvert MS \rvert < \lambda_2)$ is an indicator function. It takes a value of 1 if it is low (< $\lambda_2$) which scales the loss $\lambda_1$ times. Otherwise, it defaults to regular loss $L$.

By increasing the loss for examples where idioms are not accurately retained through back-translation, the model is encouraged to better understand and retain the meaning of idiomatic expressions. This, in turn, leads to improved performance metrics such as precision and recall, as the model becomes more adept at recognizing and appropriately handling idiomatic language during inference, resulting in better generalization to unseen data.

### 4.2.2. Cohesion based Loss Function

Idioms exhibit a lack of semantic compositional-ity or cohesion among its words also reported in earlier work (Baldwin and Kim, 2010). Given a sentence $S$ where all tokens in the subsequence $w_{i:j}$ are tagged as 1, we quantify the cohesion $C_S$ among the words in $S$ using Equation 2. It captures the mean similarity among the words in $S$.

$$C_S = \frac{1}{N} \sum_{w_i, w_j \in S, i \neq j} \text{sim}(V(w_i), V(w_j))$$

where $V(w_i)$ is an embedding vector for $w_i$, $N$ is the total number of pairs of tokens in $S$, and $\text{sim}(V(w_i), V(w_j))$ captures semantic similarity between $w_i$ and $w_j$ using $V(w_i)$ and $V(w_j)$. The ‘sim’ score is computed as the cosine similarity between the high dimensional vectors for each word. Its values range from -1 to 1, where 1 indicates high similarity and lexical cohesion, 0 represents dissimilar or orthogonal tokens, and -1 suggests that the vectors are in opposite directions. Similarly, we compute $C_{S'}$, where $S'$ is a sentence with the idiomatic tokens $w_{i:j}$ removed. The key idea is if $C_{S'}$ is substantially higher than $C_S$, then the $S$ is highly likely to contain an idiomatic phrase. This follows from the intuition that idiomatic tokens are remotely related semantically to non-idiomatic tokens in $S$ and their removal should increase the cohesion score.

We introduce this idea as loss during the fine-tuning objective. By penalizing examples with 1 classifications that are not likely to contain idioms, it is guiding the model to differentiate between idiomatic and non-idiomatic sentences. Our cohesion-based loss function $L_{\text{cohesion}}$ is expressed in Equation 3.

$$L_{\text{cohesion}} = L(1 + \lambda_3 \mathbb{1}(\lvert C_{S_1} - C_{S_2} \rvert > \lambda_4))$$

where $C_{S_1}$ and $C_{S_2}$ are the cohesion scores for sentence $S$ without and with the target idiom, respectively, $L$ is the original binary cross entropy loss, and $\mathbb{1}(\lvert C_{S_1} - C_{S_2} \rvert > \lambda_4)$ is an indicator function. It takes a value of 1 if there is sufficient difference between cohesion scores $C_{S_1}$ and $C_{S_2}$ ($> \lambda_4$) which scales the loss $\lambda_3$ times. Otherwise, it defaults to regular loss $L$.

### 4.3. Final Loss

The final loss is a linear combination of $L_{\text{retranslation}}$ and $L_{\text{cohesion}}$.

$$L_{\text{final}} = \tau_1 L_{\text{retranslation}} + \tau_2 L_{\text{cohesion}}$$ (4)

$\tau_1$ and $\tau_2$ ($0 \leq \tau_1 \leq 1$) are parameters to control the effect of both losses. These parameters depend on the accuracy of $C_S$ and $MS$, which is determined by the quality of underlying embedding vectors (Equation 2) and translation API used. More weight can be given to the more accurate value.

### 5. Experiments

In this section, we present an empirical evaluation of our models on synthetic and real-world datasets to show the capabilities of our custom loss functions. We also compare our models with state-of-the-art techniques like DISC (Zeng and Bhat, 2021) — and we observed that using our custom loss functions leads to improved accuracies.

#### 5.1. Experimental Setup

For training and testing our models, we make use of a $32 \times 2$ cores AMD EPYC5037532 server with 1 TB of RAM, and 8x A100 SXM4 80GB504. We used bert-based-uncased as our base model which we finetune.

In our experiments, we adapted the pre-trained bert-base-uncased (Devlin et al., 2018) model from Hugging Face \(^1\) and proceed with fine-tuning. We selected this model primarily for its moderate size, which strikes a balance between performance and computational efficiency. Additionally, the "uncased" variant simplifies text processing by disregarding case sensitivity, making it faster to process. These factors make it a practical choice for token classification tasks without compromising performance. We selected Hindi as the language we translate to.

We partitioned each dataset into training (80%), validation (10%), and test sets (10%). Next, we applied a BERT tokenizer on the texts for generating tokens. This step is essential because it transforms the raw text data for input into the BERT model, which operates at the token level rather

---

\(^1\)https://huggingface.co/docs/trl/en/models
than the word level. By converting words into token IDs, the tokenizer enables the model to understand and process the text effectively.

After tokenization, we aligned the labels with tokens to establish the correspondence between input tokens and the corresponding class labels. This alignment ensures that each token in the input text is associated with the correct entity label, allowing the model to learn the mapping between tokens and entity types during training. The alignment function handles cases where words are split into subwords by the tokenizer, ensuring that the labels are assigned appropriately to each token, even in the presence of subwords. We excluded special tokens representing separation between sentences and the start of the sentence from the training loss calculation by assigning them special labels.

We trained our model for three epochs, observing a sharp drop in loss over each epoch with a learning rate of '2e-5'. The training and evaluation batch sizes were set to 16. Weight decay was set to '0.01' to avoid overfitting. We set \( \lambda_1, \lambda_2, \lambda_3, \) and \( \lambda_4 \) all to 999, and \( \tau_1 \) and \( \tau_2 \) to 0.01. We repeated our experiments for three seeds and reported average accuracy values (Table 2). Additionally, it is worth noting that we observe minimal deviation in accuracy across different random seeds which underscores the robustness of the results.

5.2. Baselines

BERT-based approach (without custom loss). We fine-tuned the BERT model with binary cross-entropy loss.

BERT-based approach (with loss). We used translation, cohesion, and combination losses (described in Section 4.2) to fine-tune our BERT model.

DISC. The DISC model is based on BERT, it uses contextualized and static embeddings to encode tokens using attention, and performs token-level literal/idiomatic classification, resulting in the final output. We compare DISC with our models on the Sequence Accuracy metric described in Section 5.4.

5.3. Datasets

Table 1 describes statistics of all the datasets we have used.

1) magpie. Derived from the British National Corpus (BNC) and annotated for idiomatic expressions (PIEs)(Haagsma et al., 2020)(Consortium, 2007), the MAGPIE corpus comprises 1756 sentences containing 359 Static idioms.

2) VNC-Tokens Dataset. The VNC (Verb-Noun Combinations) corpus, sourced from the British National Corpus (BNC)(Cook et al., 2008)(Consortium, 2007), comprises 53 potentially idiomatic expressions (PIEs) with about 2500 annotated sentences, categorized as literal or figurative. Using regular expression libraries and the NLTK library, we annotated tokens as idiomatic or non-idiomatic, leveraging prior knowledge of the idiomatic expressions for pattern matching(Cook et al., 2008).

3) theidioms. We scraped 1606 of the most common English idioms from theidioms.com website using the Beautiful Soup library, resulting in a dataset of 7830 sentences. A few example sentences accompany each idiom. We use the NLTK library for lemmatization and text processing. We used a function to identify positions in sentences where a phrase similar to the idiomatic phrase occurs based on the lemmatized tokens and a similarity threshold. We use a similarity threshold of 0.9, ensuring that even slight variations of the idiomatic phrases are selected and annotated, as the idioms in the example sentences do not maintain the same format across all examples or instances of its usage. We have released a file containing the unfiltered sentences corresponding to particular idiomatic expressions.

4) formal. We utilized the EPIE corpus (English Possible Idiomatic Expressions)(Saxena and Paul, 2020), consisting of 25027 sentences. The corpus is divided into Formal and Static idioms, with 3136 sentences containing 358 Formal idioms and 21891 sentences containing 359 Static idioms. Static idioms are expressed using the exact phrase

### Table 1: Statistics of the datasets used

<table>
<thead>
<tr>
<th>Dataset</th>
<th>total number of sentences</th>
<th>#idioms</th>
<th>#sentences containing idioms</th>
<th>average sentences per idiom</th>
</tr>
</thead>
<tbody>
<tr>
<td>magpie</td>
<td>1756</td>
<td>359</td>
<td>277</td>
<td>16.05</td>
</tr>
<tr>
<td>VNC-Tokens</td>
<td>2571</td>
<td>48</td>
<td>2111</td>
<td>43.97</td>
</tr>
<tr>
<td>theidioms</td>
<td>1606</td>
<td>1606</td>
<td>1606</td>
<td>1</td>
</tr>
<tr>
<td>VNC-Tokens</td>
<td>2571</td>
<td>48</td>
<td>2111</td>
<td>43.97</td>
</tr>
<tr>
<td>theidioms</td>
<td>1606</td>
<td>1606</td>
<td>1606</td>
<td>1</td>
</tr>
<tr>
<td>Formal</td>
<td>21891</td>
<td>359</td>
<td>359</td>
<td>7830</td>
</tr>
<tr>
<td>Total</td>
<td>71,138</td>
<td>4,440</td>
<td>4,440</td>
<td>17,277</td>
</tr>
</tbody>
</table>

2https://www.nltk.org/
in all sentences, whereas formal idioms undergo lexical changes across instances. The token labeling follows the BIO convention with tags **B-IDIOM** (beginning of PIE), **I-IDIOM** (continuation of PIE), and **O** (Non-Idiom token). We merged **B-IDIOM** and **I-IDIOM** into one token to match our other datasets and treat this problem as a binary token classification task. We only focus on the formal portion of this dataset as the lexical changes to the expressions address a more robust task.

5) gtrans. We compiled a dataset of 440 sentences using GPT-3.5, featuring 22 English idioms sourced manually from online platforms. Each idiom was paired with 20 example sentences. After translating these idioms to Hindi and then back to English, we observed that Google Translate accurately retained their meanings, demonstrating its understanding of these idioms.

6) gpt+gtrans. We added 440 sentences generated by GPT-3.5 without idiomatic expressions to the gtrans dataset, resulting in a total of 880 sentences. 440 with idiomatic expressions present, and 440 without idioms. Token labeling and annotation followed similar methods as in previous datasets. Additionally, the sentences without idioms have all tokens labeled as 0.

7) theidioms 1-1. The dataset, sourced from theidioms.com, contains 1606 idioms (also present in theidioms), each with a single instance, ensuring a 1-1 mapping between sentences and idioms. We labeled tokens using pattern matching and text processing with the NLTK library. This dataset tests the model’s generalization by including idioms unseen during training.

5.4. Metrics

**Precision, Recall, F1.** We calculated precision, recall, and F1-scores for both **I** and **NI** classes, presenting them as ordered pairs.

**Macro and Weighted Average F1.** We calculated macro average as a mean of the values of the ordered pair, and the weighted average considering the relative number of each token in the complete dataset.

**Weighted-Averaged Formulae**

\[
\begin{align*}
P &= \frac{\sum_{i=1}^{N} (TP_i + FP_i) \times P_i}{\sum_{i=1}^{N} (TP_i + FP_i)} \\
R &= \frac{\sum_{i=1}^{N} (TP_i + FN_i) \times R_i}{\sum_{i=1}^{N} (TP_i + FN_i)} \\
F1 - score &= \frac{\sum_{i=1}^{N} (2 \times P_i \times R_i) \times (TP_i + FN_i)}{\sum_{i=1}^{N} (P_i + R_i) \times (TP_i + FN_i)}
\end{align*}
\]

Where **P**: Precision; **R**: Precision of the \(i^{th}\) example; **R**: Recall of the \(i^{th}\) example; **N**: Number of classes (2 in our case); **TP**: True Positives for class \(i\); **FP**: False Positives for class \(i\); **FN**: False Negatives for class \(i\); **TN**: True Negatives for class \(i\).

**Sequence Accuracy.** A sentence is only considered correct if all of its constituent tokens are correctly marked. This metric can be considered as a much more stringent metric than normal F1 and accuracy scores (Zeng and Bhat, 2021).

5.5. Results

5.5.1. With Regular Loss

Table 2 shows our results. Our base models utilizing regular binary cross entropy loss display good baseline results, however the results are consistently the lowest across all datasets and experiments compared to using custom loss functions. Our base results on EPIE formal show a large increase in metrics over the results proposed (Gamage et al., 2022). We see an increase of 1.24% in precision, 19.6% in recall and 10.9% in F1-score for the minority idiomatic class.

5.5.2. With Re-translation based Loss

Using re-translation based loss improves precision, recall, and F1 scores over binary cross entropy loss on all the datasets. It leads to large gains on theidioms, theidioms 1-1, formal, gtrans, and gpt&gtrans. This can be explained by the fact that these datasets are characterized by more comprehensive and meaningful sentences compared to MAGPIE and VNC, which often contain phrases and incomplete sentences. We also observe that the translation-based loss exhibits the highest performance on our in-house dataset, gtrans, and this outcome is anticipated, as the expressions included in the dataset primarily rely on the translation model’s capacity to grasp the genuine meaning of the idiom in its context and substitute it with a literal phrase conveying the same intended meaning. For the **formal** corpus, we see further increases of 3.3% in precision, 3.11% in recall and 3.22% in F1-score over our regular loss model. This clearly shows the superiority of translation-based loss function.

5.5.3. With Cohesion based Loss

We conducted an initial study to use cohesion based score to classify sentences into containing an idiom or not. It showed results of around 70% accuracy and varied according to the quality of the datasets. Incorporating it as an objective during training improved the accuracy further on all the datasets compared to regular
We trained our models on one dataset and tested them on another to measure the generalization capability of our models. The results show that the models trained on one dataset can generalize well to unseen data.

### Table 2: Results of applying idiom-based custom loss function on several datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Precision</th>
<th>Precision 95% CI</th>
<th>Precision Weighted Avg</th>
<th>Recall 95% CI</th>
<th>Recall Weighted Avg</th>
<th>F1 95% CI</th>
<th>F1 Weighted Avg</th>
<th>Accuracy 95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAGPIE</td>
<td>Regular Cross Entropy Loss</td>
<td>67.11</td>
<td>[64.99, 69.23]</td>
<td>71.69</td>
<td>[69.11, 74.27]</td>
<td>75.12</td>
<td>[72.39, 77.86]</td>
<td>77.46</td>
<td>[74.62, 80.30]</td>
</tr>
<tr>
<td></td>
<td>Translation</td>
<td>65.86</td>
<td>[63.74, 67.98]</td>
<td>71.09</td>
<td>[68.51, 73.67]</td>
<td>74.47</td>
<td>[71.73, 77.21]</td>
<td>76.23</td>
<td>[73.49, 79.07]</td>
</tr>
<tr>
<td></td>
<td>Retranslation Loss</td>
<td>67.24</td>
<td>[65.12, 69.36]</td>
<td>71.74</td>
<td>[69.16, 74.32]</td>
<td>75.74</td>
<td>[73.01, 78.47]</td>
<td>78.03</td>
<td>[75.3, 80.76]</td>
</tr>
<tr>
<td></td>
<td>Cohesion based Loss</td>
<td>68.92</td>
<td>[66.80, 71.04]</td>
<td>72.46</td>
<td>[70.88, 74.96]</td>
<td>76.41</td>
<td>[73.68, 79.15]</td>
<td>79.38</td>
<td>[76.64, 82.14]</td>
</tr>
<tr>
<td></td>
<td>Combination</td>
<td>68.26</td>
<td>[66.14, 70.38]</td>
<td>72.76</td>
<td>[71.18, 74.32]</td>
<td>76.70</td>
<td>[73.97, 79.44]</td>
<td>79.53</td>
<td>[76.8, 82.26]</td>
</tr>
<tr>
<td>VNC</td>
<td>Regular Cross Entropy Loss</td>
<td>66.53</td>
<td>[64.41, 68.65]</td>
<td>71.02</td>
<td>[68.44, 73.62]</td>
<td>74.44</td>
<td>[71.71, 77.14]</td>
<td>76.83</td>
<td>[74.1, 79.56]</td>
</tr>
<tr>
<td></td>
<td>Translation</td>
<td>65.28</td>
<td>[63.16, 67.40]</td>
<td>70.62</td>
<td>[68.04, 73.06]</td>
<td>74.04</td>
<td>[71.31, 76.98]</td>
<td>76.33</td>
<td>[73.6, 79.06]</td>
</tr>
<tr>
<td></td>
<td>Retranslation Loss</td>
<td>66.72</td>
<td>[64.60, 68.84]</td>
<td>71.23</td>
<td>[68.65, 74.21]</td>
<td>74.66</td>
<td>[72.02, 76.98]</td>
<td>77.06</td>
<td>[74.32, 79.8]</td>
</tr>
<tr>
<td></td>
<td>Cohesion based Loss</td>
<td>68.42</td>
<td>[66.30, 69.55]</td>
<td>71.93</td>
<td>[69.35, 74.53]</td>
<td>75.35</td>
<td>[72.72, 76.97]</td>
<td>78.04</td>
<td>[75.3, 79.82]</td>
</tr>
<tr>
<td></td>
<td>Combination</td>
<td>68.78</td>
<td>[66.66, 70.82]</td>
<td>72.43</td>
<td>[70.85, 74.96]</td>
<td>76.36</td>
<td>[73.73, 79.15]</td>
<td>79.08</td>
<td>[76.34, 81.81]</td>
</tr>
</tbody>
</table>

We observe notable improvements: precision increases by 3.67%, recall by 3.2%, and F1-score by 3.46% compared to our regular loss model. These observations validate the efficacy of utilizing both semantic cohesion and dissimilarity of idiomatic phrases within their contextual environments for our task. Instances penalized by both metrics typically represent confidently idiomatic expressions, which the model should strive to accurately classify.

### 5.5.4. With combination of losses

Using a combination of both losses improves the accuracy values on MAGPIE, VNC, formal, and gtrans by 3.4% in F1-score. This is very close to the accuracies of translation-based or cohesion-based loss functions for other datasets. In formal corpus, we observe notable improvements: precision increases by 3.67%, recall by 3.2%, and F1-score by 3.46% compared to our regular loss model. These discoveries validate the validity of utilizing both semantic cohesion and dissimilarity of idiomatic phrases within their contextual environments for our task. Instances penalized by both metrics typically represent confidently idiomatic expressions, which the model should strive to accurately classify.

### 5.5.5. Cross-domain performance across datasets

We trained our models on one dataset and tested them on another to measure the generalization capability of our models. The results show that the models trained on one dataset can generalize well to unseen data.
When we consider the examples where the DISC approach is making incorrect predictions, for instance - “Dragons can lie for dark centuries brooding over their treasures, bedding down on frozen flames that will never see the light of day.” The DISC approach incorrectly predicts only a portion of the complete expression - “see the light of day” as idiomatic, whereas our model correctly identifies the entire expression. Similarly for - “Given a method, we can avoid mistaken ideas which, confirmed by the authority of the past, have taken deep root, like weeds in men’s minds.” where the DISC model predicts “weeds in men’s minds” as the idiomatic expression with the correct instance being “taken deep root”. Our models do not falter in this case and predict all tokens for this example correctly.

In instances where the cohesion-based approach outperforms combined approaches, it is noteworthy that the Multi-Word Expressions (MWEs) are not consistently translated as expected. Consequently, the incorporation of the translation score tends to diminish overall performance. On the other hand, the translation-only model demonstrates an ability to enhance results compared to the baseline, as it successfully captures anticipated translations for certain expressions, contributing to improved overall performance.

We manually analyze the different errors that our models make on the VNC and EPIE formal datasets to gain insights into the idiom identification abilities and shortcomings in Table 5. We have categorized the errors into 5 major cases and we present examples of each type. Case 1 is where the correct idiomatic expression is identified fully but an alternate expression has also been tagged as idiomatic. This can be thought of as a limitation of the datasets rather than that of our models, as our datasets label at most one expression as idiomatic in each sentence. The second case is where an alternate expression is labeled. The reasoning for this is similar to the previous case as there may be multiple expressions that could possibly be idiomatic and our model is identifying one of them. In the third case, our model correctly identifies the idiom but also tags words surrounding the idiom. This can be ascribed to the alternate possibilities of the model and how our methodology may improve this capability. We trained the model on the theidioms dataset and tested on gtrans dataset. Table 3 shows the result. Our custom loss function based approach showcases impressive transfer capabilities.

We compared our models with DISC (Zeng and Bhat, 2021), a state-of-the-art approach for idiom token classification. We refer to the accuracy values reported in the paper to compare our technique with theirs. We kept the same train-test split for MAGPIE and VNC dataset. It should also be noted that DISC was trained for 600 epochs while our models were trained for only 5 epochs. Table 4 compares the sequence accuracies of DISC and our model. Sequence accuracy is considered as a better metric to capture the performance of such models (Zeng and Bhat, 2021). It is clear that our model outperforms DISC in sequence accuracy. This can be explained by our model’s capabilities in distinguishing between the literal and figurative idiomatic usages, possible through custom loss function training.

Table 3: Results showing transfer capabilities of our models. The model is trained on theidioms and tested on gtrans.

Table 4: Comparing DISC, a state-of-the-art idiom detection model with our technique on 2 datasets

6. Discussion

When we consider the examples where the DISC approach is making incorrect predictions, for instance - “Dragons can lie for dark centuries brooding over their treasures, bedding down on frozen flames that will never see the light of day.” The DISC approach incorrectly predicts only a portion of the complete expression - “see the light of day” as idiomatic, whereas our model correctly identifies the entire expression. Similarly for - “Given a method, we can avoid mistaken ideas which, confirmed by the authority of the past, have taken deep root, like weeds in men’s minds.” where the DISC model predicts “weeds in men’s minds” as the idiomatic expression with the correct instance being “taken deep root”. Our models do not falter in this case and predict all tokens for this example correctly.

In instances where the cohesion-based approach outperforms combined approaches, it is noteworthy that the Multi-Word Expressions (MWEs) are not consistently translated as expected. Consequently, the incorporation of the translation score tends to diminish overall performance. On the other hand, the translation-only model demonstrates an ability to enhance results compared to the baseline, as it successfully captures anticipated translations for certain expressions, contributing to improved overall performance.

We manually analyze the different errors that our models make on the VNC and EPIE formal datasets to gain insights into the idiom identification abilities and shortcomings in Table 5. We have categorized the errors into 5 major cases and we present examples of each type. Case 1 is where the correct idiomatic expression is identified fully but an alternate expression has also been tagged as idiomatic. This can be thought of as a limitation of the datasets rather than that of our models, as our datasets label at most one expression as idiomatic in each sentence. The second case is where an alternate expression is labeled. The reasoning for this is similar to the previous case as there may be multiple expressions that could possibly be idiomatic and our model is identifying one of them. In the third case, our model correctly identifies the idiom but also tags words surrounding the idiom. This can be ascribed to the alternate possibilities of the model and how our methodology may improve this capability. We trained the model on the theidioms dataset and tested on gtrans dataset. Table 3 shows the result. Our custom loss function based approach showcases impressive transfer capabilities.

5.5.6. Comparison with DISC

We compared our models with DISC (Zeng and Bhat, 2021), a state-of-the-art approach for idiom token classification. We refer to the accuracy values reported in the paper to compare our technique with theirs. We kept the same train-test split for MAGPIE and VNC dataset. It should also be noted that DISC was trained for 600 epochs while our models were trained for only 5 epochs. Table 4 compares the sequence accuracies of DISC and our model. Sequence accuracy is considered as a better metric to capture the performance of such models (Zeng and Bhat, 2021). It is clear that our model outperforms DISC in sequence accuracy. This can be explained by our model’s capabilities in distinguishing between the literal and figurative idiomatic usages, possible through custom loss function training.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Sequence Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAGPIE</td>
<td>Regular Cross Entropy Loss</td>
<td>90.19</td>
</tr>
<tr>
<td></td>
<td>Translation Retranslation Loss</td>
<td>91.31</td>
</tr>
<tr>
<td></td>
<td>Cohesion based Loss</td>
<td>91.46</td>
</tr>
<tr>
<td></td>
<td>Combination</td>
<td>91.51</td>
</tr>
<tr>
<td></td>
<td>DISC</td>
<td>87.47</td>
</tr>
<tr>
<td>VNC</td>
<td>Regular Cross Entropy Loss</td>
<td>93.75</td>
</tr>
<tr>
<td></td>
<td>Translation Retranslation Loss</td>
<td>96.88</td>
</tr>
<tr>
<td></td>
<td>Cohesion based Loss</td>
<td>96.88</td>
</tr>
<tr>
<td></td>
<td>Combination</td>
<td>96.88</td>
</tr>
<tr>
<td></td>
<td>DISC</td>
<td>93.31</td>
</tr>
<tr>
<td>Error Type</td>
<td>Sentence with PIE</td>
<td>Prediction</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Multiple Expressions Predicted</td>
<td>I then walked across to the photographers and lost my temper and then lost my head</td>
<td>lost my temper, lost my head</td>
</tr>
<tr>
<td>Alternate expression detected</td>
<td>Cantona will have to kick his heels on the sidelines if the manager had his way.</td>
<td>had his way</td>
</tr>
<tr>
<td>Extra tokens surrounding expression</td>
<td>Julia had her attention caught by the commotion.</td>
<td>attention caught by</td>
</tr>
<tr>
<td>Partial</td>
<td>His blistering turn of speed and attitude made him an instant hit with the fans.</td>
<td>hit</td>
</tr>
<tr>
<td>Predicting Nothing</td>
<td>Everyone talks about hitting a wall at the 24 mile mark.</td>
<td>Empty String</td>
</tr>
</tbody>
</table>

Table 5: Different error types along with examples and the incorrect prediction. The ground truth values have been colored blue in sentences.

The effective labeling of the identical expression in different occurrences. The fourth case “Partial”, constitutes instances where only a segment of the idiomatic expression is identified, with the specific localization of the entire idiom boundary remaining imprecise. The last error category involves the absence of predictions when the model fails to recognize idiomatic usage, even when it is present. The effectiveness of our model is contingent upon the caliber of annotation and various other external factors.

### 7. Future Work

The latest advancements in Natural Language Processing (NLP) have led to the extensive utilization of a range of transformer-based models. We can adjust our own loss functions to refine different architectures effectively. We can create an intuitive and efficient tool utilizing these fine-tuned models to detect an idiom in a given sentence. This tool should offer a straightforward and accessible experience for a broad range of users, with minimal technical expertise required. To continuously improve the overall performance of our models, we can systematically address each identified error category. This might involve analyzing error patterns and refining the fine-tuning process accordingly.

### 8. Acknowledgements

We are grateful to the anonymous reviewers for their insightful comments which substantially improved this manuscript. This work was performed using Sharanga, the high performance computing cluster at the BITS Pilani Hyderabad Campus.

### 9. Bibliographic References


Prateek Saxena and Soma Paul. 2020. **EPIE Dataset: A Corpus For Possible Idiomatic Expressions.**


Ad Hoc Compounds for Stance Detection

Qi Yu1,2, Fabian Schlotterbeck3, Hening Wang3, Naomi Reichmann1, Britta Stolterfoht2, Regine Eckardt1,2, Miriam Butt1,2
1Department of Linguistics, University of Konstanz
2Cluster of Excellence “The Politics of Inequality”, University of Konstanz
3Department of Modern Languages, University of Tübingen
firstname.lastname@[uni-konstanz, uni-tuebingen].de

Abstract

In this paper we focus on a subclass of multi-word expressions, namely compound formation in German. The automatic detection of compounds is a known problem and we argue that its resolution should be given more urgency in light of a new role we uncovered with respect to ad hoc compound formation: the systematic expression of attitudinal meaning and its potential importance for the down-stream NLP task of stance detection. We demonstrate that ad hoc compounds in German indeed systematically express attitudinal meaning by adducing corpus linguistic and psycholinguistic experimental data. However, an investigation of state-of-the-art dependency parsers and Universal Dependency treebanks shows that German compounds are parsed and annotated very unevenly, so that currently one cannot reliably identify or access ad hoc compounds with attitudinal meaning in texts. Moreover, we report initial experiments with large language models underlining the challenges in capturing attitudinal meanings conveyed by ad hoc compounds. We consequently suggest a systematized way of annotating (and thereby also parsing) ad hoc compounds that is based on positive experiences from within the multilingual ParGram grammar development effort.

Keywords: ad hoc compounds, attitudinal meaning, stance detection, German, universal dependencies, psycholinguistic validation, large language models

1. Introduction

The automatic detection of compounds is known to be a difficult problem for Natural Language Processing (NLP) (Constant et al., 2017; Baldwin and Kim, 2010), particularly in a language like German which uses compounding as a central strategy for novel word formation. In this paper we present research showing that novel, ad hoc compound formations in German can be used strategically to convey attitudinal meaning, thus making them an interesting area of research from the overall perspective of stance detection (Mohammad et al., 2016; Schiller et al., 2021) and adding urgency to finding reliable ways of automatically detecting compounds, and in particular, novel compound formations in a language. We adduce evidence that combines insights from theoretical linguistic analysis, corpus linguistic investigations and psycholinguistic experimentation to show that a subset of ad hoc compounds in German, termed enigmatic compounds (ECs; Wildgen, 1981) are indeed systematically used to convey attitudinal meaning and are therefore of inherent interest for the NLP task of stance detection.

The types of compounds falling under the rubric of ECs are illustrated in (1)–(3). We noted the use of such compounds for the expression of stance as part of a larger project investigating the framing of politically charged issues across several German newspapers. We have marked the extra expressive meaning carried by these ad hoc compound formations as attitudinal meaning (AM) in the examples.

(1) Flüchtlinge wollen Österreich meiden und lieber in Merkel-Land einreisen.
Refugees want to avoid Austria and instead enter Merkel-Country.
AM: The German refugee crisis is Merkel’s fault.
(source: Facebook)

(2) Jede 5. China-Maske ist unbrauchbar.
‘Every fifth China mask is unusable’
AM: China is notorious for low-quality products.
(source: BILD, 2020-05-03)

(3) Neue Stelle für Kopftuch-Praktikantin.
‘New position for hijab intern’
AM: Religious practices of Muslims often cause trouble for others.
(source: BILD, 2016-08-25)

Intended but deliberately masked meanings of speakers such as the AMs above are known to play a crucial role in political communication (Beaver and Stanley, 2018). Our data indicate that ECs are a useful rhetorical device for speakers/authors to implicitly convey attitudinal meaning. In particular, we observed that ECs can be employed as so-called dog-whistles (Henderson and McCready, 2019), whereby their use – at least for a certain time span – speaks to a certain subgroup and con-
veys a meaning that is on the surface rather vague, but decodable as to its hidden meaning by that subgroup. This seems particularly interesting, as ad hoc compounds are instances of innovated language and thus, dog whistles and pejorative uses in expressing attitudinal meaning clearly cannot rest on conventional lexical meanings alone. This makes an automatized stance detection task challenging yet interesting.

We consequently examine how compounds are currently treated in available dependency parsers and Universal Dependencies (UD) treebanks (de Marneffe et al., 2021; Nivre et al., 2016) for German. We find that the current treatment is uneven. We also explored the potentially greater capabilities of current large language models (LLMs) with respect to detecting attitudinal meaning in ECs, but found that while the results from LLMs may provide an explanation for substantial variation in our experimental data, they do not easily capture the effect of our experimental manipulation involving ECs. We therefore provide suggestions for a systematic UD annotation for compounds that is based on the multilingual ParGram grammar development experience (Butt et al., 1999; Sulger et al., 2013) so as to allow for a more successful learning.

This paper is structured as follows: in section 2 we provide some background on the current state-of-the-art. We follow this in section 3.1 with the results of a corpus study of three different newspapers, which yielded indications that more conservative leaning newspapers used ad hoc compounds to trigger attitudinal meaning more than other newspapers. However, our results are most robust for the conservative tabloid BILD, which is also known for an editorial policy that prefers the use of pictures coupled with short, expressive texts. The greater use of ad hoc compounds could also therefore just be a matter of newspaper writing style. To test the perception of attitudinal meaning in compounds, we therefore designed and executed an experiment that sought to establish the stance triggering effect of ECs using psycholinguistic methods. This is described in section 3.3, following a discussion of how the semantics of ECs are hypothesized to come about in section 3.2. In section 4, we report on our attempts to use current LLMs to simulate our experimental results. We did not find any indication that these models can capture the central contrasts observed in the experimental outcomes. In section 5, we combine the insights from the corpus and psycholinguistic results to formulate recommendations for the systematic annotation of compounds in corpora. Section 6 concludes the paper.

2. Background

2.1. Evaluative Language

Evaluative language is of interest for a range of NLP tasks, perhaps currently most prominent among the sentiment analysis (Pang and Lee, 2008; Taboada et al., 2011), but also hate speech detection (Davidson et al., 2017) and stance detection (Mohammad et al., 2016; Schiller et al., 2021). Sentiment analysis and stance detection are closely related tasks but differ in their overall goals. Sentiment analysis is concerned with identifying whether a given text, sentence or passage overall can be classified as being positive, negative or neutral. This has generally involved a bag-of-words approach, where the internal structure of the text is not considered and the target has generally been reviews or statements about movies, books, objects or persons. More recently, approaches to sentiment analysis have become more nuanced in that the classification aims at aspect based (what aspect is the sentiment targeted at, e.g., the acting or the plot?) or target based (what is the precise target of the sentiment, e.g. an iPhone or the ear phones that came with the iPhone?) sentiment analysis (Alturayeif et al., 2023).

Stance detection is informed by the Stance Triangle defined by Du Bois (2007), by which the author of a text is taken to want to influence or align the recipient/reader of the text with his/her beliefs. The difference between sentiment analysis and stance detection is that in sentiment analysis the object of study are texts expressing a given sentiment, prototypically reviews. In these the author articulates their opinion to an audience, but is not necessarily seeking to align the audience with their own views. Given that our overall interest lies in determining how issues are framed (Chong and Druckman, 2007), we are interested in stance detection as a subtask for determining the overall framing of a narrative or text. As far as we have been able to determine, no previous work on stance detection has attempted to include information from compounds in a focused manner, though Li and Caragea (2019) note as part of their stance detection error analysis that it would be useful to separate the individual components used in hashtags such as #VotingGOP or #NoHillary, as found in the SemEval-2016 dataset developed specifically for stance detection (Mohammad et al., 2016).

Stance detection includes identifying instances of subjective language (Wiebe et al., 2004). Subjective language can be detected on the basis of linguistically informed lexicon and/or construction based information (Biber and Finegan, 1989; Biber and Conrad, 2019; Taboada et al., 2011), or it can be detected by machine learning on the basis of annotated data (Alturayeif et al., 2023). Our data
is German, for which an automatic annotation tool for subjective language already exists (El-Assady et al., 2016, 2019). This tool provides POS-tagging and syntactic parsing of a given text along with a systematic identification of linguistic cues for subjective language such as the annotation of various modals or German discourse particles (Zimmermann, 2011). However, the tool does not include a facility for the automatic detection of ECs.

2.2. Annotation and Automatized Detection of Compounds

The compounds in (1)–(3) each contain a hyphen. However, German compounds generally do not contain a hyphen. One could hypothesize that ad hoc compounds in particular are marked with a hyphen, but our data also contains instances of ad hoc formations such as Asylprüger ‘asylum beater’ and Migrantenschreck ‘migrant scare’ that have been written without a hyphen. Nevertheless, the inclusion of a hyphen provides a potentially important clue for the automatic identification of at least a subset of compounds and one that could be picked up on easily. In surveying existing dependency parsers and treebanks annotated according to the Universal Dependencies (UD) scheme (de Marneffe and Manning, 2008; de Marneffe et al., 2021; Nivre et al., 2016), we found that only the Stanza toolkit (Qi et al., 2020) could reliably identify German compounds characterized by a hyphen. The sample of other dependency parsers for German that we tried were not reliable in the identification of compounds, with most merely labeling them with the POS-tag of NN for common nouns, as shown in Figure 1 for the Mate parser (Björkelund et al., 2010).¹ where both of the compounds Flüchtlingsorganisation ‘refugee organisation’ and Asyl-Verschärfungen ‘asylum restrictions’ are tagged as NN. The same is true for spacy,² ParZu (Sennrich et al., 2009)³ and a German dependency parser² based on the MaltParser framework⁵, as well as the very high quality morphological analyzer SMOR (Schmid et al., 2004). An investigation of UD treebanks for German collected at the INESS website⁶ yielded much the same result. See also the reports and conclusions in Baldwin et al. (2023).

A morphological analyzer can be integrated as

---

³https://github.com/rsennrich/ParZu
⁴https://pub.cl.uzh.ch/users/siclemat/lehre/ecl1/ud-de-hunpos-maltparser/html/
⁵http://www.maltparser.org/
⁶https://clarino.uib.no/iness-prod/treebanks/

Figure 1: Sample Mate parse

part of a dependency parser and so we set out to test SMOR for our purposes. We worked with this system because it has been designed especially to deal with the productive word formation possibilities in German, including ad hoc compounds occurring in newspaper texts. However, a pilot study with respect to our data showed that while SMOR is indeed able to identify ad hoc compounds successfully, the uneven nature of the overall results means that quite a bit of manual postprocessing would be required to obtain a useable data set. For example, the ad hoc compound Pegida-Anhänger ‘Pegida follower/supporter’ could not be analyzed at all, while the lexically established word Bezirksamt ‘district office’ was incorrectly analyzed. Instead of the correct split into the morphemes Bezirk+s+amt (the s is a so-called linking element that appears for phonological reasons), the word was split into Bezirk+Sa mt ‘district velvet’ as one of the three most likely results.

Thus, the challenges posed by automatic compound detection (Constant et al., 2017; Baldwin and Kim, 2010) continue to be a problem, and one that we argue – gains more urgency given our findings. Given that ECs express attitudinal meaning and as such can provide an important linguistic cue for stance detection, search for these cues should be operationalized.

3. Enigmatic Compounds

In this section, we combine results from a corpus linguistic study and a psycholinguistic experiment to show that ECs can be used systematically to express attitudinal meaning. We first present results from a corpus study that demonstrates a systematic use of ECs to express a negative stance in newspapers (section 3.1). We then discuss how ad hoc compounds invite such attitudinal meaning from a theoretical linguistic aspect (section 3.2), and report a psycholinguistic experiment (section 3.3) to confirm that ECs are indeed a systematic part of language use. All data and code resulting from this work are publicly available at: https://github.com/qi-yu/enigmatic-compounds.

3.1. Corpus Study

Our corpus study was conducted as part of a larger investigation into the framing of the Syrian refugee crisis by German newspapers in the time span of
2014–2018. We chose the three German newspapers with the highest circulation rates (IVW, 2023): BILD, Frankfurter Allgemeine Zeitung (FAZ) and Süddeutsche Zeitung (SZ). These three newspapers cover a representative range of political leanings within the German media landscape, with BILD being the most conservative on the political spectrum, the SZ the most left leaning, and the FAZ also leaning towards the conservative end. Moreover, they also build a diverse sample of different styles, with BILD characterized as a tabloid newspaper whereas FAZ and SZ contain high quality, in-depth reporting. Examples (4)–(6) illustrate the different styles: they are headlines from articles reporting on the same event and published around the same time.

(4) **BILD**, 2014-09-29: *Folter-Skandal in deutschen Asylbewerberheimen: „Die Wachleute schlagen und treten uns“*  
‘Torture-scandal in German asylum seekers’ accommodations: “The guards beat and kick us”

(5) **FAZ**, 2014-09-30: *Misshandlung von Asylbewerbern: Sicherheitsleute werden überprüft*  
‘Mistreatment of asylum seekers: security guards undergo checks’

(6) **SZ**, 2014-09-30: *Ermittlungen nach Misshandlungsverdacht in drei Flüchtlingsheimen*  
‘Investigations into suspected mistreatment in three refugee accommodations’

As part of this investigation, we noticed that compounds seemed to be used to express a negative stance towards refugees and the handling of the crisis by the government (see, e.g., *Folter-Skandal ‘torture-scandal’ in (4)*). A more in-depth investigation of this phenomenon was hampered by the difficulty of automatically detecting compounds. We therefore decided to experiment with training a language model on the basis of annotated data. The best performing model was a logistic regression language model on the basis of annotated data. The model that resulted in a value of 0.68 for F1.

Given these unsatisfactory results, we asked ourselves whether it was indeed necessary to detect these compounds. As we report on in the following sections, the result of our investigations has established that ECs indeed have the potential for providing important information for stance detection. Efforts should be redoubled so as to be able to operationalize ECs for stance detection.

Our data set consisted of a total of 23,889 articles. Given the necessity for manual annotation of the compounds (since automatic detection is a challenge), we considered only the articles’ headlines for our study. We manually identified 19,353 referential/neutral ad hoc compounds and 828 ECs in these headlines. We structured our resulting data set into pieces of information as follows: the target compound, the sentence in which it appeared, the year it was released, the newspaper source, and the annotation (0 = referential, 1 = enigmatic). We categorized the compounds as enigmatic if they met the following two criteria:

(i) the compound carries an attitudinal meaning;
(ii) the compound is an innovative, ad hoc formation and is thus not established in a recognized dictionary or lexicon of German.

To validate the application of criterion (ii), the German dictionary *Duden* as well as the online dictionary *Digitales Wörterbuch der deutschen Sprache* were consulted. For instance, based on these criteria, the compound Karajan-Schüler ‘Karajan student’ was defined as referential (neutral), as it does not seem to express an additional evaluative meaning; only its literal meaning is being transmitted. In contrast, the compound Flüchtlings-Tsunami ‘refugee tsunami’ was categorized as enigmatic, as it does not only refer to a large amount of refugees, but it also carries an additional AM to the effect that refugees are overwhelming the transit and host countries.

Our overall results of the annotation per newspaper are given in Table 1. They show that BILD uses by far the most ECs. We furthermore sampled the top most ECs per newspaper per year and found that BILD predominantly used these in contexts of discussing security or issues of criminality, whereas the FAZ and the SZ placed a greater emphasis on problems of capacity and the rights of individual refugees. For example, with compounds such as Asylprügler ‘asylum beater’, *Migrantenschreck ‘migrant scare’, and Amok-Afrikaner ‘amok African’, BILD focuses on criminality related to the refugees in Germany through the use of ECs. This is in line with the hostile reporting style previously observed for tabloid newspapers (see Innes, 2010; Kleins teuber and Thomass, 2007).

<table>
<thead>
<tr>
<th>Newspaper</th>
<th>#Enigmatic</th>
<th>#Neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td>BILD</td>
<td>726</td>
<td>10,059</td>
</tr>
<tr>
<td>FAZ</td>
<td>58</td>
<td>5,525</td>
</tr>
<tr>
<td>SZ</td>
<td>44</td>
<td>3769</td>
</tr>
</tbody>
</table>

Table 1: Total number of enigmatic and neutral compounds in newspaper headlines.

Whether or not the ECs are employed as attention-getters as part of BILD’s sensationalist

---

7https://www.duden.de
8https://www.dwds.de
writing style (see Greussing and Boomgaarden, 2017) becomes irrelevant in the face of their extensive use by BILD in combination with the negative attitudinal meanings triggered by these ECs: they are a significant contributing factor to the overall articulated stance towards a topic.

3.2. Compound Meaning

Compounds have a range of interpretational possibilities because their meanings are not compositional. Earlier theoretical linguistic studies on compound meaning share the common assumption that there is some covert, meaning-decisive semantic relation $R$ between the constituents of a compound:

\[(7) \quad \text{Let } C_1 C_2 \text{ be a compound where } [C_1] = m_1 \text{ and } [C_2] = m_2. \text{ Then: } [C_1 C_2] = R(m_1, m_2)\]

Levi (1978) and Fanselow (1981) propose taxonomies of semantic relations that play a role in ad hoc compound interpretation, and Meyer (1993), Ryder (1994) and Benczes (2009) propose different assumptions on how the semantic relations in (7) are derived. In the simplest case, ad hoc compounds serve as abbreviations for phrases, as in Kapitel: Schüler von Karajan ‘student of Karajan’. In (1)-(3); however, there is clearly an attitudinal meaning, an extra meaning dimension that is not found in the equivalent non-compound phrase. Consider, for example, China-Maske ‘China mask’ in the context in (2): it has a negative attitudinal meaning that is not conveyed by the compositional alternative phrase chinesische Maske ‘Chinese mask’.

Sassoon (2011) opens an avenue towards an explanation of attitudinal enrichment in ECs. The author summarizes comparative studies in the conceptual structure of nouns and adjectives: nouns denote similarity-based concepts with a prototype structure (Murphy, 2002), whereas adjectives denote rule-based properties (Kennedy, 1999). The distinction is backed up by converging evidence from neurolinguistics, patholinguistics and language acquisition. Sassoon’s proposal predicts that the modifier in ECs (China- in China-Maske) contributes to a similarity-based concept. This happens, plausibly, by adding a further dimension in which exemplars must match the prototype. Specifically, similarity-based categorization rests on prototypical values that can be attributed to this dimension. In our example the similarity-based categorization invites a comparison to typical ‘products from China’, which provides a hook for the accommodation of an interpretation including negative expectations about products from China. The corresponding adjective in a phrasal alternative (‘Chinesen mask’), in contrast, adds a simple categorial property ‘be Chinese’ (yes/no). Sassoon thus predicts that the processing of modifiers does not trigger novel stereotypes and should not provide an entry-point for attitudinal meaning.

We were interested in this prediction as it also provides a systematic way of testing whether the attitudinal meaning associated with ECs we found as part of the corpus study in section 3.1 is a general, systematic part of language use or whether it is perhaps attributable to the particular corpus. If the attitudinal meaning associated with ECs is found to be a systematic part of language, it provides another argument for taking ECs seriously as part of the overall task of stance detection. We describe the psycholinguistic experiment we set up to test Sassoon’s prediction in section 3.3.

3.3. Experiment

3.3.1. Methods

Materials and Design We manually selected 21 text snippets from newspapers and social media which contain ECs along the lines of (1)–(3) that trigger negative AM according to our own intuitions. We restricted ourselves to negative AMs in our experiment as these were more prevalent in the corpus study. To test for the AM-triggering effects of ECs, three variants were created from each snippet. Table 2 provides examples of such snippets (translated into English). The three variants were:

(i) **COMPOUND**: original text snippet with the EC.

(ii) **PHRASAL**: EC substituted by a corresponding phrasal construction.

(iii) **NEUTRAL**: EC substituted by a corresponding noun that is attitudinally neutral.

The PHRASAL condition controls for truth-conditional information, as it conveys the same truth-conditional information as the COMPOUND condition but in a pragmatically unmarked phrasal expression, not an ad hoc compound. The condition NEUTRAL is intended as a baseline: though there is no stylistic difference in terms of innovative language use between PHRASAL and NEUTRAL, these two conditions differ in their information load, as the modifier part of the PHRASAL (and COMPOUND) condition provides extra information that is not necessary for reference resolution but can be inferred from the prejacent context (see Table 2). Comparing the PHRASAL and the NEUTRAL condition thus allows us to examine whether the addressees’ perception of the attitudinal strength is affected by such additional but in principle unnecessary information while keeping the style constant. With these three conditions, we test the following two hypotheses:
(i) **COMPOUND VS. PHRASAL** (different style, same information load): compounding amplifies the perceived attitudinal strength;

(ii) **PHRASAL VS. NEUTRAL** (same style, different information load): the additional information that is not necessary for reference resolution amplifies the perceived attitudinal strength.

The items were distributed over 3 lists using a Latin square. 24 stylistically similar text snippets were added to each list as fillers. For each item, participants rated its attitudinal strength by answering question (8) on a 7-point Likert-scale.

(8) **How does the author talk about ________?**

1 = positive
1 = neutral
2 = negative

As our overall interest is in the framing of politically charged discourse, we also collected the political leaning of each participant by asking question (9) at the end of the experiment. This allows us to further control whether participants' perception of attitudinal strength is affected by their political leaning:

(9) **In politics, people often use “left” and “right” to denote political leanings. Where would you place your own political leaning?**

1 = left
2 = neutral
3 = right

Participants The participant recruitment and data collection was carried out online via Prolific.9 212 German native speakers, identified through Prolific’s demographic prescreening function, took part in the study (103 female, 102 male, 7 other genders; mean age = 26.52 years, SD = 8.10 years). The experiment was carried out anonymously and voluntarily. Each participant received a compensation of £8.50 per hour, a fair rate suggested by Prolific.

3.3.2. Results

Figure 2 shows the rating distributions of each condition. Overall, all items were rated rather negatively, with more negative ratings for **compound** than **phrasal** and **neutral** conditions. We fitted a **cumulative link model (CLM)** with random effects using the R package **ordinal** (Christensen, 2018) to test these differences statistically. CLM is a variant of logistic regression generalized to multinomial ordinal dependent variables. A CLM models the probability, \( P(Y \leq j) \), that an ordinal response variable \( Y \) is less than or equal to a specific category \( j \in \{1, \ldots, J\} \) \((J \geq 2)\) according to the equation below, where \( \theta_j \) is the intercept of level \( j \), \( x \) is a vector of predictors, and \( \beta_j \) is a vector of coefficients:

\[
\logit(P(Y \leq j)) = \log \frac{P(Y \leq j)}{P(Y > j)} = \theta_j - x^T \beta
\]

In our initial model, we predicted participants’ ratings using condition and participants’ political leaning as well as their interactions. For the predictor condition, **phrasal** is set as reference level (cf. hypotheses above). For the predictor political leaning, we mapped the seven original levels (see (9) above) to three aggregated levels in order to ease the model interpretation: 1-3 = **LEFT**, 4 = **NEUTRAL**, 5-7 = **RIGHT**. We used dummy encoding to code the three levels. Random intercepts and random slopes were fitted for items and participants, as likelihood ratio tests showed that they improved the model fit. In a following model selection step based on likelihood ratio tests, the predictor political leaning and the interaction term were removed as they were not significant in improving the model fit (likelihood ratio test without interaction: \( \chi^2(2) = 0.384, p = 0.826 \); likelihood ratio test with interaction: \( \chi^2(6) = 2.004, p = 0.919 \)).

Our final model showed a significant difference between compound and the reference level phrasal. Compared to **phrasal**, compound led to a significant decrease in the logit of ratings in lower (i.e., more positive) categories (compound vs. phrasal: \( \beta = 0.526, SE = 0.152, p < 0.001 \)). No significant difference between neutral and phrasal was found (neutral vs. phrasal: \( \beta = -0.272, SE = 0.176, p = 0.123 \)).

3.3.3. Discussion

The result of our experiment with a large population is in line with the corpus study. The significant decrease of the likelihood of positive ratings indicates that the authors’ negative attitudes are perceived as more pronounced when ECs are used instead of the **phrasal** counterpart. The difference in information load between **phrasal** and **neutral** condition did not show significant influence on the participants’ perception of attitudinal strength. Furthermore, the non-significant effect of political leaning as well as the non-significant interaction between political leaning and condition show that the increased perception of attitudinal meaning in ECs is general part of how language works, rather than being domain or population specific.

4. Simulations with Large Language Models (LLMs)

Recent advances of LLMs have underscored their remarkable utility across a wide variety of NLP
The federal government purchased more than 108 million masks from China for German clinics and medical practices. However, about 10 percent of these China-masks are unusable for medical purposes.

The big refugee-mistake: no labor market miracle has been brought by refugees. Unfortunately, most of the newcomers were not Syrian doctors and engineers.

Table 2: Example stimuli (translated into English from German). The variation between different conditions are marked in bold.

<table>
<thead>
<tr>
<th>COMPOUND</th>
<th>PHRASAL</th>
<th>NEUTRAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>The federal government purchased more than 108 million masks from China for German clinics and medical practices. However, about 10 percent of these China-masks are unusable for medical purposes.</td>
<td>The federal government purchased more than 108 million masks from China for German clinics and medical practices. However, about 10 percent of these Chinese masks are unusable for medical purposes.</td>
<td>The federal government purchased more than 108 million masks from China for German clinics and medical practices. However, about 10 percent of these masks are unusable for medical purposes.</td>
</tr>
<tr>
<td>The big mistake about refugees: no labor market miracle has been brought by refugees. Unfortunately, most of the newcomers were not Syrian doctors and engineers.</td>
<td>The big mistake about refugees: no labor market miracle has been brought by refugees. Unfortunately, most of the newcomers were not Syrian doctors and engineers.</td>
<td>The big mistake: no labor market miracle has been brought by refugees. Unfortunately, most of the newcomers were not Syrian doctors and engineers.</td>
</tr>
</tbody>
</table>

Figure 2: Distribution of participants’ ratings by condition.

We conducted experiments testing two of the latest versions of ChatGPT, namely GPT-4 and GPT-3.5-turbo (Achiam et al., 2023; Touvron et al., 2023). However, the challenges associated with compound detection, particularly in identifying the associated attitudinal meanings of certain types of compounds like ECs remain significant. An avenue worth exploring is whether current LLMs encounter comparable challenges in this domain, particularly within the context of our psycholinguistic experiment. Recent work similar in spirit focused on human-likeness of LLMs’ linguistic performance, e.g., testing language models on different syntactic phenomena, (Wilcox et al., 2018, 2020; Futrell et al., 2019; Arehalli et al., 2022) semantic judgements (e.g., Levy et al., 2017; Kauf et al., 2023), and on subtle pragmatic phenomena like irony or compliance with Gricean maxims (Hu et al., 2023; Tsvilodub et al., 2023).

We conducted experiments testing two of the latest versions of ChatGPT, namely GPT-4 and GPT-3.5-turbo (Achiam et al., 2023), employing various temperature settings. We designed a prompt that closely simulates the task employed in the experiment, and fed experimental items from the previous psycholinguistic experiment with human participants to these LLMs. Among these configurations, the one utilizing GPT-4 with a temperature set to 0 yielded the best results. Overall, we found that the best LLM captured a significant portion of the observed by-item variance in our experimental results ($R^2 = .48$, $p < .001$; see Fig. 3). Contrary to our experimental results, however, at the condition level, there was no indication of any alignment with human data ($R^2 = .43$, $p = .55$).

Our current LLM simulations thus provide initial evidence that these models currently have difficulty picking up cues for AMs conveyed by ECs. While further analyses (e.g., of the involved contextual embeddings or attention patterns) or future LLMs may provide a closer match between human ratings and modeling results, the current lack of effect was observed concurrently with the models’ ability of capture substantial variation in other dimensions of our experimental results. This further highlights the specific subtleties and challenges involved in the detection and interpretation of ECs.

5. Recommendations and Outlook

We have now established that ECs systematically convey attitudinal meaning which can provide information for the NLP task of stance detection. We have also established that the current state of the art with respect to dependency parsers and UD treebank representations does not facilitate the automatic detection and identification of ECs. We furthermore showed that LLMs also struggle with the identification of EC contributions that are natural for humans, despite their otherwise impressive capabilities.

In this section, we propose that the UD community adopt a uniform approach towards the annotation of compounds. A systematic and uniform approach towards annotation will be able to result
in better down-stream machine learning and thus better results with respect to dependency parsers. Concretely we recommend adopting the approach deployed within the multilingual ParGram grammar development effort (Butt et al., 1999; Sulger et al., 2013). This is illustrated in Figures 4 and 5 from the German ParGram grammar (Dipper, 2003). The grammar is hosted on the INESS XLE website and can be used interactively.¹⁰ The German ParGram grammar is based on Lexical Functional Grammar (LFG; Dalrymple, 2001), which has a context-free phrase structure part (the c-structure) and a dependency part (the f-structure). A c-structure of the compounds in question are simply tagged as common nouns (N[comm]). However, as shown in Figure 5, the German grammar also contains a finite-state morphological analyzer (DMOR, a precursor of SMOR; Schiller, 1994) and if one uses the built-in facility to look into the morphological analysis, one can see that the morphological analyzer separates out the parts of the compound into a base noun (the head noun) and the modifier, with the modifier then being flagged as such in the

¹⁰https://xle.uni-konstanz.de/iness/xle-web
dependency analysis at f-structure (Figure 4). We propose a UD annotation of the following form: a separation out of the head noun from the modifier, with the modifier being identified clearly as such in the dependency analysis. The curly brackets in the f-structure denote a set. This indicates that this attribute may have more than one value. Translating this into UD, we would assume that a head noun can have more than one modifier, all of which would be represented as sisters (at the same level) in the dependency graph.

However, a systematic annotation scheme only provides us with part of the necessary information for the detection of ad hoc compounds. Another part will necessarily involve the consultation of existing dictionaries, as was done as part of our corpus study (section 3.1). This type of lexical information can be further supplemented by lists of nouns and likely combinations, as was done in Schulte im Walde and Borgwaldt (2015). We propose that the data set we gleaned from the German newspaper study could be used in this way: one can compile an initial list of compounds for any given domain, identify the parts (i.e., heads and modifiers) of the compounds, and use the combined list of heads and modifiers as a seed list. This seed list can be then fed into models calculating clusters of lexically similar words for the identification of further ad hoc compounds. We leave this approach for exploration in further research.

6. Conclusion
We have presented a study of German ad hoc compounds that establishes that a subset of these compounds, dubbed enigmatic compounds, is systematically used to convey extra attitudinal meaning. We showed this via a combination of theoretical linguistic analysis, a corpus study and a psycholinguistic experiment. We also showed that the extra attitudinal meaning was predominantly used to express a negative stance in the newspapers and thus see enigmatic compounds as providing an important source of information for the end user NLP task of stance detection. A survey of existing dependency parsers and treebanks for German showed an uneven treatment for the annotation of German compounds and we therefore proposed a systematic annotation scheme that is based on the existing multilingual ParGram grammar development experience. We believe that a systematic annotation combined with lexical resources of the type developed in this paper will help ameliorate the challenge of automatized compound detection.

7. Acknowledgements
This project was funded by the Deutsche Forschungsgemeinschaft (DFG – German Research Foundation) under Germany’s Excellence Strategy – EXC-2035/1 – 390681379.

8. Bibliographical References


Stefanie Dipper. 2003. Implementing and documenting large-scale grammars – German LFG. *Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS)*, 9(1).


Author Index

Ahmadi, Sina, 70
Ahrenberg, Lars, 33
Akkurt, Furkan, 207
Alam, Meesum, 188
Alves, Diego, 81
Antoine, Jean-Yves, 88
Barbu Mititelu, Verginica, 147
Ben Khelil, Cherifa, 88
Bon, Mija, 154
Butt, Miriam, 231
Chiarcos, Christian, 147
Chontaeva, Bermet, 207
Čibej, Jaka, 154
Çöltekin, Çağrı, 56, 207
Dakota, Daniel, 198
Das, Apurba, 220
Degaetano-Ortlieb, Stefania, 81
Díaz Hernández, Roberto, 24
Ding, Chenchen, 63
Eckardt, Regine, 231
Ehren, Rafael, 115
Eslami, Soudabeh, 207
Evang, Kilian, 115, 147
Fioravanti, Irene, 138
Francis, Emilie, 43
Gamba, Federica, 125
Gantar, Polona, 154
Gatti, Lorenzo, 70
Gervasi, Osvaldo, 138
Giouli, Voula, 147
Gow-Smith, Edward, 178
Hadj Mohamed, Najat, 88
Hadrich-Belguith, Lamia, 88
Hanink, Emily, 188
Hellwig, Oliver, 50
Iwamoto, Ran, 4
Jain, Kanishka, 98
Jobanputra, Mayank, 56
Jumalieva, Gulnura, 207
Kaiser, Hour, 63
Kallmeyer, Laura, 115
Kanayama, Hiroshi, 4
Kasieva, Aida, 207
keskes, Iskandar, 88
Krek, Simon, 147
Krstev, Cvetana, 106
Kübler, Sandra, 188, 198
Kuzgun, Aslı, 207
Lareau, Francois, 14
Levshina, Natalia, 2
Liu, Li, 14
Mao, Jiannan, 63
Markantonatou, Stella, 147
Marković, Aleksandra M., 106
Maršan, Büşra, 207
Mascioli, Arianna, 43
Matsumoto, Tadahiro, 63
Mehta, Maitrey, 56
Mi, Maggie, 178
Mihajlov, Teodora Sofija, 106
Miyamoto, Kohtaroh, 4
Morad, Peshmerge, 70
Muraoka, Masayasu, 4
Ohko, Takuya, 4
Osenova, Petya, 147
Perri, Damiano, 138
Phelps, Dylan, 178
Pickard, Thomas M. R., 178
Reichmann, Naomi, 231
Savary, Agata, 88
Schlotterbeck, Fabian, 231
Schmidt, Elena, 81
Sellmer, Sven, 50
Spina, Stefania, 138
Stanković, Ranka, 106, 147
Stephen, Abishek, 125, 163
Stolterfoht, Britta, 231
Stoyanova, Ivelina, 147
Striebel, Jacob, 198
Szawerna, Maria Irena, 43
Taguchi, Chihiro, 207
Tanaka, Hideki, 63
Tayyar Madabushi, Harish, 1
Teich, Elke, 81
Tiberius, Carole, 147
Tyers, Francis, 188
Upadhyay, Prajna Devi, 220
Utiyama, Masao, 63
Vaidya, Ashwini, 98
Villavicencio, Aline, 178
Wang, Henning, 231
Washington, Jonathan, 207
Yayavaram, Arnav, 220
Yayavaram, Siddharth, 220
Yu, Qi, 231
Žabokrtský, Zdeněk, 125
Zeman, Daniel, 147, 163