RANLPStud 2025

Proceedings of the 9th Student Research Workshop

associated with

The 15th International Conference on Recent Advances in Natural Language Processing RANLP'2025

8-10 September, 2025

STUDENT RESEARCH WORKSHOP
ASSOCIATED WITH THE INTERNATIONAL CONFERENCE
RECENT ADVANCES IN
NATURAL LANGUAGE PROCESSING'2025

PROCEEDINGS

8-10 September 2025

Series Online ISSN 2603-2821

Designed by INCOMA Ltd.
Shoumen, BULGARIA

Preface

The RANLP 2025 Student Research Workshop (RANLPStud'2025) is a special track of the established international conference Recent Advances in Natural Language Processing (RANLP'2025).

The RANLPStud is being organised for the 9th time and this year is running in parallel with the other tracks of the main RANLP 2025 conference. The target of RANLPStud'25 is to be a discussion forum and provide an outstanding opportunity for students at all levels (Bachelor, Masters, and Ph.D.) to present their work in progress or completed projects to an international research audience and receive feedback from senior researchers.

The RANLPStud'25 received a good number of submissions, this year fifteen (13) papers were submitted to the event coming from Asia, The Americas (North and South) and Europe, a fact which was reflecting the great number of events, sponsors, submissions, and participants at the main RANLP conference.

We have accepted 3 excellent student papers for oral presentations, two of them shared the Best Paper Award and 6 submissions were presented as posters.

We did our best to make the reviewing process in the interest of our authors, by asking our reviewers to give as exhaustive comments and suggestions as possible, as well as to maintain an encouraging attitude. Each student submission was reviewed by at least two Programme Committee members, who are specialists in their field and were carefully selected to match the submission's topic.

This year, as usual, we invited both strictly Natural Language Processing (NLP) papers, and submissions at the borderline between two sciences (but bearing contributions to NLP).

The topics of the accepted submissions include: chatbots and conversational agents; dialogue systems; electronic dictionaries, terminologies and ontologies; information extraction; information retrieval; irony and sarcasm detection; language generation; language resources and corpora; linked data; mathematical, statistical, machine learning and deep learning models; morphology; multilingual NLP; NLP for healthcare; opinion mining and sentiment analysis; parsing; POS tagging; question answering; speech recognition; syntax, semantics, discourse, pragmatics, dialogue, lexicon; text categorisation; theoretical and application-orientated papers related to NLP.

We are thankful to the members of the Programme Committee for having provided such exhaustive reviews and to the conference mentors, who provided additional comments to participants.

The RANLPStud 2025 Organisers

Boris Velichkov, FMI, Sofia University "St. Kliment Ohridski", Bulgaria
Ivelina Nikolova-Koleva, IICT, Bulgarian Academy of Sciences and Graphwise, Bulgaria
Milena Slavcheva, IICT, Bulgarian Academy of Sciences, Bulgaria

Organizers:

Boris Velichkov, FMI, Sofia University "St. Kliment Ohridski", Bulgaria
Ivelina Nikolova-Koleva, IICT, Bulgarian Academy of Sciences and Graphwise, Bulgaria
Milena Slavcheva, IICT, Bulgarian Academy of Sciences, Bulgaria

Programme Committee:

Cengiz Acarturk (Jagiellonian University, Poland)
Svetla Boytcheva (Graphwise, Bulgaria)
Daniel Dakota (Indiana University, United States)
Hansi Hettiarachchi (Lancaster University, United Kingdom)
Eduard Hovy (University of Melbourne, Australia)
Tracy Holloway King (Adobe Inc., United States)
Dmitry Ilvovsky (HSE University, Russian Federation)
Ivan Koychev (Sofia University "St. Kliment Ohridski", Bulgaria)
Sandra Kübler (Indiana University, United States)
Paloma Moreda Pozo (University of Alicante, Spain)
Rafael Muñoz Guillena (University of Alicante, Spain)
Ivelina Nikolova-Koleva (Bulgarian Academy of Sciences and Graphwise, Bulgaria)
Petya Osenova (Sofia University "St. Kliment Ohridski" and Bulgarian Academy of Sciences, Bulgaria)
Pavel Rychly (Masaryk University, Czech Republic)
Milena Slavcheva (Bulgarian Academy of Sciences, Bulgaria)
Sylvia Vassileva (Sofia University "St. Kliment Ohridski", Bulgaria)
Boris Velichkov (Sofia University "St. Kliment Ohridski", Bulgaria)
Kristina Yordanova (University of Greifswald, Germany)
Valentin Zmiycharov (Sofia University "St. Kliment Ohridski", Bulgaria)

Table of Contents

A Multi-Baseline Framework for Ranking Global Event Significance Using Google Trends and Large Language Models

Zenan Chen
School of Information, Journalism and Communication, University of Sheffield, UK
zchen249@sheffield.ac.uk

Abstract

Determining global event significance lacks standardized metrics for quantifying worldwide impact. While Google Trends has demonstrated utility in domain-specific studies, its application to global event ranking remains limited. This paper presents a framework combining Google Trends data with large language models for automated global event ranking. This study leverages Command R+ and Llama 3.3-70B-Instruct to generate contextually relevant event keywords and establishes significance through comparative search volume analysis against baseline keywords, incorporating temporal weighting mechanisms to address chronological biases. The proposed methodology identified globally significant events across technology, health, sports, and natural disasters from a dataset of 1,094 events (2020-2024) extracted from Wikipedia.

1 Introduction

Global events, from pandemics and natural disasters to geopolitical crises and economic disruptions, shape international relations and affect millions of lives across national boundaries. Understanding their relative significance is essential for effective resource allocation, risk assessment, media coverage and informed policymaking. However, quantification of global event significance presents fundamental challenges due to the absence of standardized international metrics and inconsistently formatted information across different sources. This capability is essential for news organizations prioritizing coverage, policy makers identifying emerging issues, and researchers analyzing global trends. Traditional approaches to event ranking often rely on expert judgment, media coverage analysis, or domain-specific indicators (Wang et al., 2008; Kong et al., 2012), each introducing scalability limitations and subjective biases.

This study addresses these constraints by introducing an automated analysis framework that leverages search behavior data to measure event significance through demonstrated public attention patterns. Building upon the established relationship between search volume and public interest, this work utilizes Google Trends (GT) data (Google Trends) to create standardized significance metrics through baseline comparison analysis. While previous applications of search data have focused on specific domains such as financial or health contexts (Husnayain et al., 2020; Knipe et al., 2020), the framework developed here extends across diverse event categories and geographic contexts. Critical temporal biases are addressed through weighted analysis, with cross-event comparability ensured via consistent baseline reference points.

The paper's primary contributions include: (1) a multi-baseline comparison framework ensuring consistency; (2) an automated keyword generation system eliminating manual selection biases; and (3) an cross-domain application to diverse event categories. To our knowledge, this represents the first systematic integration of GT and LLMs for global event significance ranking.

After reviewing existing approaches to event significance assessment and search data applications in event analysis research, the paper presents the ranking methodology in detail. The approach encompasses automated keyword generation through large language models (LLMs) and multi-baseline aggregation strategies.

2 Related Work

In this section, current methods used to measure and rank significance of global events will be described. Then, the usage of GT in measuring the significance of events will be discussed.

Proceedings of the Student Research Workshop associated with RANLP-2025, pages 1–9, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/issn.2603-2821.2025_001

2.1 Event assessment and ranking methods

Several approaches have been developed for global event assessment and ranking. AI-GlobalEvent (Sufi, 2022) analyzes and identifies global breaking news with sentiment extraction, yet it lacks mechanism for measuring event significance. News event ranking has been categorized into three distinct methods: ranking news streams, incorporating external sources, and employing query-based approaches (Setty et al., 2017).

Topic popularity forms the basis of one ranking system developed for incremental corpora, though this approach fails to incorporate historical features and may undermine assessments of long-lasting event significance (Corso et al., 2005). The Top Story Identification Task (Soboroff et al., 2010) evaluated news based on perceived popularity within the blogosphere. Query-based methods, by contrast, rank news relative to specific user queries rather than providing assessments of independent event significance (Setty et al., 2017).

User engagement metrics have gained prominence in recent news ranking frameworks. One approach integrates news diversity (measured through user-generated Twitter content), completeness, and speed metrics, effectively synthesizing external source-based ranking with independent event assessment (Karimi et al., 2021). Additionally, a more recent frameworks for extracting key news events from media streams through temporal trend analysis and unsupervised clustering techniques that identify events capturing significant public attention (Nakshatri et al., 2023).

Our approach employs GT as an external data source to measure public attention. Unlike query-dependent systems that rank events relative to specific user searches, our framework provides query-independent significance assessment by measuring inherent event importance through aggregated search behavior patterns.

2.2 Using Google Trends to evaluate the event significance

GT provides reports on search term popularity using the Google search engine (Cebrián and Domenech, 2022). Existing studies primarily use GT to evaluate health and financial event significance. GT search intensity correlates with various societal impact measures, including economic effects, policy changes, cultural discourse, and disease outbreaks (Liu et al., 2020; Simionescu and Cifuentes-Faura, 2022; Mavragani and Ochoa, 2019), making it a valuable indicator for assessing public interest in events or topics.

Unlike traditional media coverage metrics or expert assessments, search behavior reflects genuine public interest, providing objective measures of how events resonate with audiences. Recent work has explored combining LLMs with GT data for automated keyword generation in search engine optimization applications, demonstrating the potential of integrating language models with search trend analysis (Vadlapati, 2024).

3 Methods

The methodology consists of four primary stages (Figure 1). First, global events were extracted from Wikipedia's chronological pages. Second, LLMs generated contextually relevant search keywords for each event, as GT requires specific search terms rather than complete descriptions. Third, these keywords were used to collect GT search data with baseline comparisons over five years. Finally, composite significance scores were calculated to enable systematic event ranking based on global attention.

3.1 Wikipedia event extraction

Wikipedia was selected as the event source because contributors worldwide collaboratively summarize significant annual events in standardized chronological articles[1], providing global coverage with diverse perspectives and enabling systematic extraction. These chronological articles contain events categorized by month with occurrence times and brief descriptions (Hienert and Luciano, 2015). While 198 language versions exist, the English version was selected for practical methodological reasons: (1) English serves as a lingua franca for international news and global events coverage, making English search terms more likely to reflect international rather than purely regional significance, and (2) using a single language ensures consistency in keyword extraction and search query formulation throughout the research process.

This research extracted all 1,094 events and their descriptions appearing in Wikipedia from 2020 to 2024, including political elections, natural disasters, economic crises, sports competitions, technology breakthroughs and disruptions, health crises, and military conflicts worldwide.

[1]The page for 2020 events: `https://en.wikipedia.org/wiki/2020`

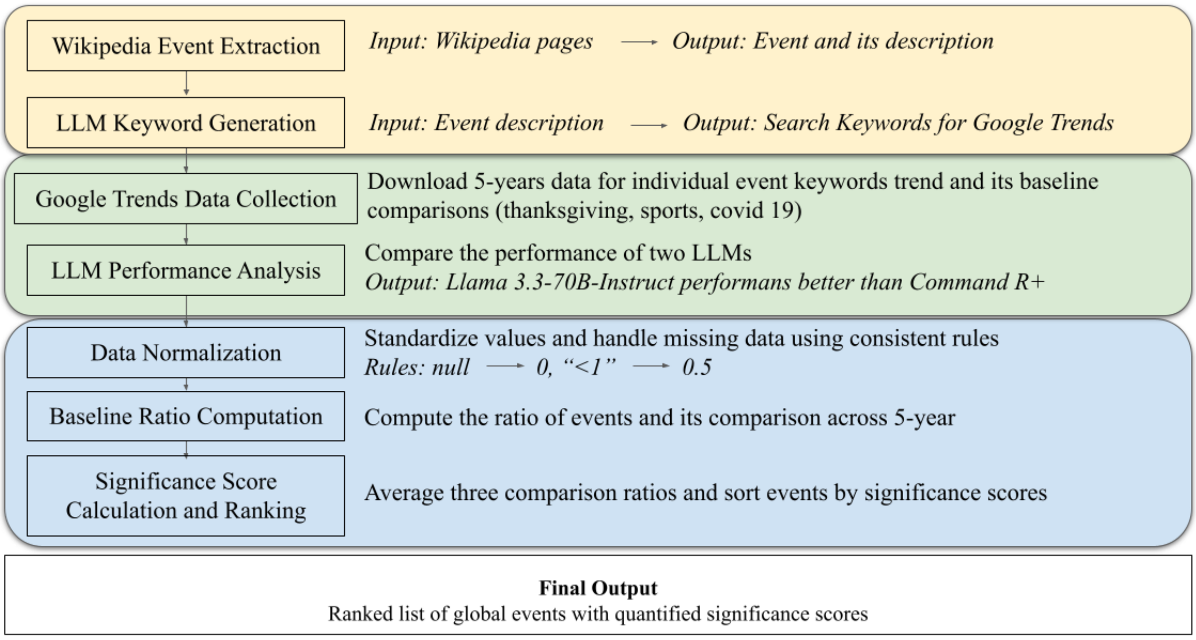


Figure 1: Event significance analysis workflow from data collection to ranking generation

3.2 LLM keywords generation

To identify the most significant global events, this study employed GT to rank events based on search intensity. The underlying assumption was that events more frequently searched on Google at the worldwide level correspond to greater influence at the global scale, as public search behavior serves as a reliable indicator of collective attention (Costola et al., 2021). Major sporting events, significant political development or breakthrough scientific discoveries can all generate high search volumes.

LLMs were employed to automatically extract keywords from Wikipedia's event descriptions. This process is essential because Wikipedia descriptions often contain detailed narratives unsuitable for direct GT searches, which require concise, targeted terms. For example, a Wikipedia entry about a natural disaster might contain extensive geological and casualty information, but effective GT keywords would be simplified terms like the disaster name and location

> **Source Events (Wikipedia)**: Flash floods struck Jakarta, Indonesia, killing 66 people in the worst flooding in over a decade.
> **Keywords extracted manually:** flash flood Jakarta

Specifically, Llama 3.3-70B-Instruct (Meta Llama, 2024) and Command R+ (Cohere, 2024) were selected based on their established capabilities in text generation tasks and availability. Identical prompts[2] for comparison purposes, examples and reasoning frameworks were applied to both LLMs.

The generated keywords were standardized to ensure GT compatibility by removing special characters and formatting according to GT query requirements. The quality of extracted keywords was initially evaluated based on:

- Peak timing match: whether search peaks of keywords generated align with event occurrence based on Wikipedia entry
- Delayed recognition: whether the keywords produced measurable search data on GT rather than data insufficiency warnings, even if the peak search occurred with some delay

The initial exploration on the 20 randomly sampled events revealed the key difference in Llama 3.3-70B-Instruct's consistent preservation of specific year information – a critical requirement for GT data collection methodology. This temporal specificity proves essential when analyzing events such as natural disasters, political elections, and other time-sensitive occurrences where precise chronological context directly impacts search behavior patterns.

> **Source Events (Wikipedia)**: The 2020 Serbian parliamentary election is held to elect all 250 members of the National Assembly of Serbia and

[2]https://github.com/Zenanc/Prompt_for_keywords_generation

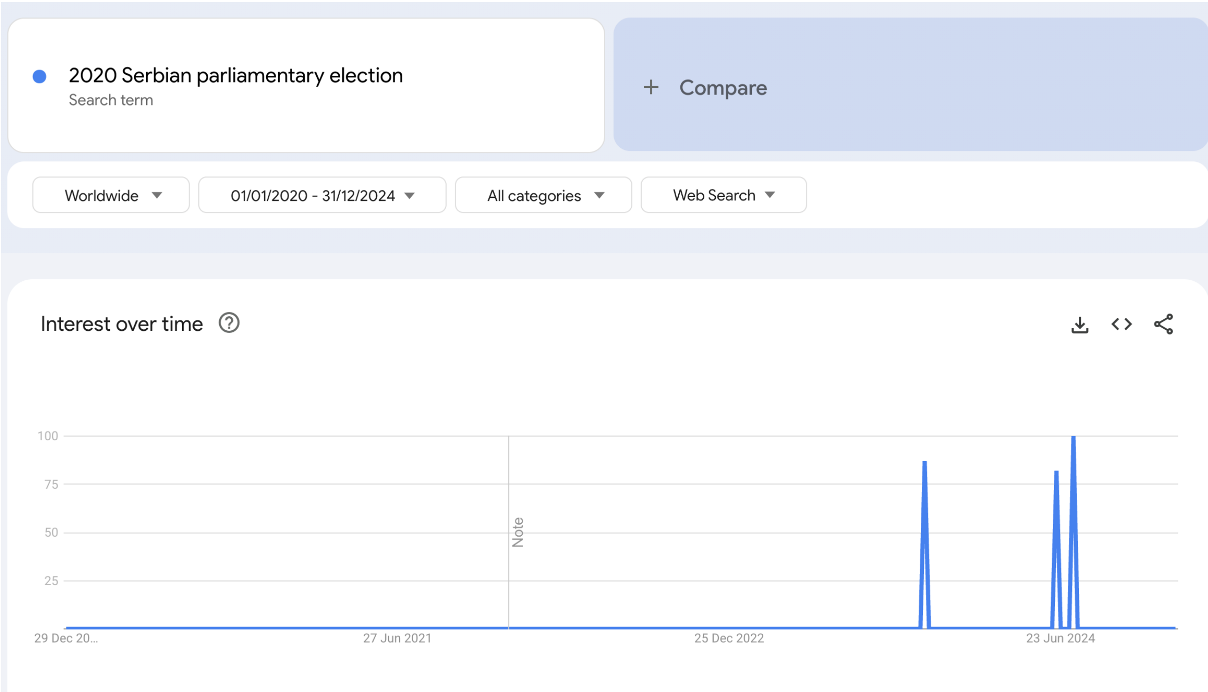


Figure 2: Google Trends search data for event keyword "2020 Serbian parliamentary election" (2020-2024) worldwide

> the ruling For Our Children coalition won 188 out of 250 seats.
> **Keywords extracted by Command R+:** Serbian parliamentar[sic] election results
> **Keywords extracted by Llama 3.3-70B-Instruct:** 2020 Serbian parliamentary election

To evaluate the performance of these two LLMs, a secondary comparison was conducted. For events, where Llama 3.3-70B-Instruct generated keywords containing year information, the corresponding Command R+ keywords were augmented with the same temporal markers. This approach enabled a controlled comparison between the two LLMs' keyword generation capabilities while isolating the effect of temporal specificity. By standardizing the temporal component across both models' outputs, the analysis can determine whether the performance differences stem from temporal markers alone or from other qualitative aspects of keyword selection.

3.3 Google Trends data collection

GT provides relative search interest data where higher values indicate greater popularity of the search term within the chosen time frame and region (see Figure 2). However, this presents limitations for direct event comparison: GT normalizes data to each term's maximum within the selected timeframe rather than providing absolute search volumes—when comparing event keywords, the highest-searched keyword's peak becomes 100, with all others scaled proportionally, and restricts simultaneous comparisons to five keyword groups maximum (each describe one event). Therefore, events cannot be compared directly using raw GT data.

The GT methodology involves inputting keywords and selecting both geographic regions (worldwide, US, UK, etc.) and specific time periods. This study utilized worldwide geographic scope from January 1, 2020 to December 31, 2024.

To ensure comparability across all events, a two-step data collection method was employed. In the first step, individual GT data for each event keyword from 2020 to 2024 was downloaded to identify the peak search week for each event as it is impossible to obtain more fine-grained data than week intervals within a historical search period. Peak week analysis serves as a validation mechanism for LLM-generated keywords, where smaller temporal gaps between event occurrence and search peaks indicate superior keyword formulation that better reflects real-time public attention patterns.

In the second step, pairwise comparison data between each event keyword and a set of baseline keywords were downloaded across the full 2020-2024 period. This approach ensured consistent comparative scaling and enabled events to be ranked according to their significance scores against identical baseline keywords. The full five-year window was necessary as relative scaling between event keywords and baselines keywords varies dramatically depending on the temporal window selected.

Three baseline keywords were initially selected: “covid 19”, “weather”, and “black friday”. These represented consistent everyday search interest (weather), recurring seasonal events (black friday), and major global phenomena (covid 19), providing a comparative framework capturing various dimensions of human search behavior.

However, preliminary findings revealed these keywords exhibited very high search volumes, resulting in very few events producing sufficient aggregate significance scores. To address this limitation, a refined set was adopted: “sports” (everyday search interest), “thanksgiving” (seasonal events), and “covid 19” (retained for global significance). This adjustment maintained the comparative framework’s foundation while preserving detectability across a broader range of global events.

3.4 Peak week identification and LLM performance analysis

Historical GT data for each event keyword from 2020-2024 were downloaded via DataforSEO API[3], a third-party service that enables researchers to access the same publicly available trend data accessible through GT’s web interface.

Llama 3.3-70B-Instruct generated 1,024 valid keywords from 1,094 events, with 70 keywords excluded due to generation failures (producing no output or generating fewer than the required minimum of two keywords per event). Command R+ generated 1,069 keywords from 1,094 events, with 25 excluded for duplication across different events. After Google Trends data collection, 885 out of 1,024 Llama 3.3-70B-Instruct keywords and 842 out of 1,069 Command R+ keywords successfully returned valid peak data with temporal timeframes. Failed keywords either contained formatting errors incompatible with GT queries or represented events with insufficient search volume.

The duplication issue in Command R+ demonstrates overgeneralization when processing semantically similar events. For instance, two distinct Covid-19 events generated identical keywords, compromising specificity required for accurate trend analysis as distinct events become indistinguishable in search data.

For keywords exhibiting multiple search peaks throughout the five-year period, the first highest peak was selected, as this initial peak typically represents the moment of maximum public attention to an event, providing the most representative measure of initial global impact. Additionally, selecting the first peak minimizes potential confounding effects from anniversary coverage, follow-up events, or media retrospectives in subsequent years.

Following peak identification, temporal gaps between event occurrence and search peaks were computed. Since exact peak dates within each week are unavailable, gap calculations employed a standardized approach: events occurring within the peak search week were assigned a gap value of zero, indicating perfect temporal alignment; events preceding the peak week were measured from the event date to the peak week’s start date; events following the peak week were measured from the peak week’s end date to the event date.

The results showed that Llama 3.3-70B-Instruct maintains superior temporal accuracy even when Command R+ keywords were augmented with identical year information (Table 1). Specifically, Llama 3.3-70B-Instruct achieved relatively better temporal alignment (Gap = 0) for 256 events (28.92%), compared to 95 events (11.28%) for Command R+. This performance differential persisted across short-term temporal windows, with Llama 3.3-70B-Instruct capturing 327 events (36.94%) within a seven-day window versus 118 events (14.01%) for Command R+.

Notably, Command R+ exhibited a pronounced tendency towards extended temporal gaps, with 53.56% of events showing gaps exceeding 365 days, compared to 35.37% for Llama 3.3-70B-Instruct. Examination of specific cases revealed the nature of this temporal displacement. For instance, “Greece wildfires” reached its search peak on June 19-24, 2023, despite the actual event occurring on August 3, 2021, where the keywords registered an initial but smaller search peak (August 8-14, 2021). Similarly, “Abdallah Hamdock resignation protest” reached its only search peak on July 16-22, 2023, while the actual events occurred on January 2, 2022. These examples suggested that Command R+’s keyword formulation, despite temporal augmentation, generated search terms that failed to capture immediate public attention or pro-

[3] https://dataforseo.com/

Model Name	Command R+ (N=842)	Llama 3.3-70B-Instruct (N=885)
Gap = 0	95 (11.28%)	**256 (28.92%)**
$0 <$ Gap ≤ 7 days	23 (2.73%)	71 (8.02%)
$7 <$ Gap ≤ 30 days	33 (3.92%)	40 (4.52%)
$30 <$ Gap ≤ 365 days	241 (28.62%)	205 (23.16%)
Gap > 365 days	451 (53.56%)	313 (35.37%)

Table 1: The temporal gap comparison of Command R+ and Llama 3.3-70B-Instruct

duced keywords that aligned with later waves of interest rather than initial event-driven search.

The persistent performance gap indicated that temporal accuracy depends not only on the inclusion of year information, but on the underlying semantic and lexical choices that determine how effectively keywords match actual search behavior patterns during critical attention periods.

3.5 Data normalization

A consistent transformation was applied to account for the three value types returned by GT: (1) "null" indicates insufficient data for the event within the given time period, assigned a value of 0; (2) "$<$1" indicates searches below the minimum reporting threshold, assigned a value of 0.5 for numerical calculations to distinguish it from 0 and 1; and (3) numerical values 1-100 represent quantified relative search interest.

3.6 Baseline ratio computation

To ensure all the events were comparable across the five-year period, each event significance was measured by comparing search intensity against fixed baseline events. For each event keyword E and the set of baseline keywords $B = \{\text{covid 19}, \text{sports}, \text{thanksgiving}\}$, the significance ratio relative to each baseline was computed:

$$r_{E,B} = \frac{\sum_{t=2020}^{2024} V_{E,t} \cdot w_t}{\sum_{t=2020}^{2024} V_{B,t} \cdot w_t} \tag{1}$$

where:

- $V_{E,t}$ represents search volume for event E in year t
- $V_{B,t}$ represents search volume for baseline B in year t
- w_t represents time weight for year t

This cumulative approach captured the total social impact of events rather than just peak attention, recognizing that sustained or recurring interest might often indicate lasting influence on public consciousness.

3.7 Time weighting strategy

This weighting method attempted to address the temporal bias: events occurring in 2020 naturally had higher search volumes during 2020-2024 compared to events occurring in 2024. Higher weights for recent years corrected this systematic advantage of earlier events.

Given that the events span 2020-2024, but comparison requires fixed temporal windows for consistency, distance-based time weights were applied:

$$w_t = 0.05 * (t - 2020) + 0.1 \tag{2}$$

3.8 Significance score calculation

For event E with baseline ratio vector ($r_{E,covid}$, $r_{E,sports}$, $r_{E,thanksgiving}$), the significance scores are calculated using three different methods. Each method has distinct mathematical properties affecting how baseline performances are combined.

The arithmetic mean provides the most intuitive aggregation, testing all baseline comparison equally:

$$Score_{AM}(E) = \frac{1}{|B|} \sum_{b \in B} r_{E,b} \tag{3}$$

However, this method is disproportionately influenced by extreme outliers.

The geometric mean emphasizes proportional relationships and requires consistent performance across baselines for high scores:

$$Score_{GM}(E) = \left(\prod_{b \in B} r_{E,b} \right)^{1/|B|} \tag{4}$$

rank	keywords	weighted geometric score	peak week start	peak week end
1	US Open	29.666	2023-09-03	2023-09-09
2	Stock market	27.341	2020-03-08	2020-03-14
3	2022 FIFA World Cup	17.073	2022-11-27	2022-12-03
4	2023 Cricket World Cup	7.812	2023-10-29	2023-11-04
5	lunar eclipse	7.417	2022-05-15	2022-05-21
6	WHO COVID	6.557	2020-03-22	2020-03-28
7	2023 Rugby World Cup	5.303	2023-09-10	2023-09-16
8	James Webb Space Telescope	4.928	2022-07-10	2022-07-16
9	Cape Verde	4.736	2024-01-28	2024-02-03
10	2024 ICC T20 World	4.701	2024-05-26	2024-06-01

Table 2: Top 10 Events Ranked by Weighted Geometric Mean Score

However, this method may undervalue events with mixed significance patterns.

The harmonic mean takes a more conservative approach, penalizing low-performing baseline comparisons:

$$Score_{HE}(E) = \frac{|B|}{\sum_{b \in B} \frac{1}{r_{E,b}}} \tag{5}$$

This method is excessively conservative for events with mixed significance patterns.

To assess the ranking stability of the three aggregation methods, real events with diverse baseline ratio patterns were analyzed. These genuine cases revealed the practical advantages and limitations of each method in handling mixed-baseline performance and varying ratio distributions.

The real event data analysis demonstrated significant differences among aggregation methods in handling mixed baseline performance patterns. For instance, "Openai Chat" illustrated these ranking differences effectively. This event ranked 44th with arithmetic mean, 14th with geometric mean, and 8th with harmonic mean, despite having identical underlying data. The dramatic ranking variation (from 44th to 8th) demonstrated how different aggregation methods can fundamentally alter event prioritization. Arithmetic mean's low ranking (44th) suggested that OpenAI Chat's performance is diminished by extreme baseline ratios that skewed the average. Harmonic mean's high ranking (8th) indicated strong consistent performance across baselines, while geometric mean provided a moderate assessment (15th) that balanced both exceptional and weaker baseline performances.

This ranking instability highlighted that geometric mean offered the most reliable approach for event significance assessment, providing consistent evaluation that neither inflated nor unfairly penalized mixed performance patterns.

4 Result and discussion

This study successfully collected comparison data for 804 global events (with Llama 3.3-70B-Instruct), enabling significance scoring through comparison with three established baseline keywords. The proposed scoring methodology effectively ranked all 804 events, with the weighted geometric mean proving a solid measure of relative significance across diverse event categories. Table 2 presents the top-ranked events using keywords generated by Llama 3.3-70B-Instruct.

The ranking results demonstrated the methodology's effectiveness in capturing diverse event categories, including sports events ("US Open", "FIFA World Cup"), scientific achievements ("James Webb Telescope"), natural phenomena ("lunar eclipse"), and tragic events ("WHO COVID"). This diversity validated the methodology's capacity to identify various forms of global attention rather than only crisis-driven events.

The prominence of sporting events in the top rankings reflected their substantial capacity to generate global attention, with the "US Open" achieving the highest (score: 29.666) and the "FIFA World Cup" ranking third (score: 17.073). It might be because mega-events can capture widespread public engagement across diverse demographic and geographic segments.

The second ranked "Stock market" (score: 27.341), corresponded to the global stock market crash that began on February 20, 2020, following growing instability due to the Covid-19 pandemic. This crash ended on April 7, 2020, representing one of the most significant financial disruptions in recent history, and it validated the methodology's capacity to capture major adverse economic events.

The bottom events, such as "2024 Namibian election Netumbo" (score: 0.001487), "Falkland Islands land mine" (score: 0.001013), and "Sichuan earthquake Luding" (score: 0.000984), demonstrated clear differentiation from top global events. This ranking aligned with expected global significance patterns, as these represented region-specific political developments, geographically isolated incidents, and localized disasters respectively.

4.1 Validation

To validate the weighting method, this study assessed whether the time weighting scheme introduces systematic bias towards recent events. The validation examined event distribution across score bins by year, calculated expected versus actual numbers of top 10% events, and computed correlation between time weights and bias ratios.

It is revealed that most events (85%) fall within the 0-1 score range using geometric mean values (minimum score: 0.000053, maximum score: 0.979104), and this distribution pattern remains stable across years with 129-170 events annually. Very few events achieve scores above 5 (only 7 out of 804), with highest score categories containing just 3 events across all years.

The correlation coefficient of -0.9175 between time weights and bias ratios indicated that recent years actually produce fewer high-scoring events than expected, providing evidence that the time weighting methodology does not artificially inflate recent event significance. The slight overrepresentation in 2020-2021 likely reflects genuinely significant historical events (such as the Covid-19 pandemic and related global disruptions).

5 Conclusion

This study presents a novel framework for ranking global event significance through multi-baseline comparison using GT data. The methodology employs three distinct baseline keywords-"covid 19" (burst pattern), "sports" (stable pattern), and "thanksgiving" (seasonal pattern)-to provide robust comparative assessment across diverse temporal contexts. This diversified approach decreases the risk of biased significance scores from single-baseline fluctuations.

Key methodological findings demonstrate that Llama 3.3-70B-Instruct outperforms Command R+ in generating keywords from event descriptions that capture immediate public response to global events, achieving superior temporal alignment between event occurrence and search peaks (28.92% vs. 11.28% alignment). Among aggregation methods, geometric mean proves most effective, providing balanced significance assessment while avoiding the outlier sensitivity of arithmetic mean and excessive conservatism of harmonic mean.

This approach represents the first systematic integration of GT and LLMs for global event ranking, introducing a scalable, automated methodology that eliminates manual keyword selection biases while maintaining cross-temporal comparability. The study's novelty lies in combining automated keyword generation with multi-baseline aggregation, offering practical applications for news organizations, policy makers, and researchers requiring objective event significance assessment.

Limitation

There are few limitations of this proposed methodology. First is that major events often generate diverse search terminologies, potentially diluting their apparent significance. The outbreak of Covid-19 pandemic, for instance, might be searched as "Covid-19", "coronavirus", "covid-19 pandemic" or other variants, fragmenting the search signal.

Second, the API's data collection process occasionally returns null values despite the existence of actual search data, potentially leading to the systematic exclusion of some events.

Third, the methodology's reliance on English-language search terms may introduce geographic and linguistic bias, potentially undermining events of significance in non-English speaking regions. This limitation could affect the global representativeness of the event rankings.

Acknowledgments

This research was supported by funding from the Economic and Social Research Council (ESRC). I would like to express my sincere gratitude to Dr Judita Preiss and Professor Peter A Bath for their invaluable supervision and continuous support throughout this research. I also thank Nicholas Musembi for his technical assistance in downloading the Wikipedia events.

References

Eduardo Cebrián and Josep Domenech. 2022. Is Google Trends a quality data source? *Applied Economics Letters*, 30(6):811–815.

Cohere. 2024. Cohere's Command R+ Model. Accessed: 2025-08-10.

Gianna M. Del Corso, Antonio Gullí, and Francesco Romani. 2005. Ranking a stream of news. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 97–106.

Michele Costola, Matteo Iacopini, and Carlo R.M.A. Santagiustina. 2021. Google search volumes and the financial markets during the COVID-19 outbreak. *Finance Research Letters*, 42.

Google Trends. Google Trends. Accessed: 2025-08-10.

Daniel Hienert and Francesco Luciano. 2015. Extraction of Historical Events from Wikipedia. In *The Semantic Web: ESWC 2012 Satellite Events*.

Atina Husnayain, Anis Fuad, and Emily Chia-Yu Su. 2020. Applications of Google Search Trends for risk communication in infectious disease management: A case study of the COVID-19 outbreak in Taiwan. *International Journal of Infectious Diseases*, 95:221–223.

Samaneh Karimi, Azadeh Shakery, and Rakesh Verma. 2021. Online News Media Website Ranking Using User Generated Content. *Journal of Information Science*, 47(3):340–358.

Duleeka Knipe, Hannah Evans, Mark Sinyor, Thomas Niederkrotenthaler, David Gunnell, and Ann John. 2020. Tracking online searches for emotional wellbeing concerns and coping strategies in the UK during the COVID-19 pandemic: a Google Trends analysis. *Wellcome Open Research*, 5:220.

Liang Kong, Shan Jiang, Rui Yan, Shize Xu, and Yan Zhang. 2012. Ranking news events by influence decay and information fusion for media and users. In *CIKM '12: Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1849–1853.

Ying Liu, Geng Peng, Lanyi Hu, Jichang Dong, and Qingqing Zhang. 2020. Using Google Trends and Baidu Index to analyze the impacts of disaster events on company stock prices. *Industrial Management & Data Systems*, 120(2):350–365.

Amaryllis Mavragani and Gabriela Ochoa. 2019. Google Trends in Infodemiology and Infoveillance: Methodology Framework. *JMIR Public Health and Surveillance*, 5(2).

Meta Llama. 2024. meta-llama/Llama-3.3-70B-Instruct. Accessed: 2025-08-10.

Nishanth Nakshatri, Siyi Liu, Sihao Chen, Dan Roth, Dan Goldwasser, and Daniel Hopkins. 2023. Using LLM for Improving Key Event Discovery: Temporal-Guided News Stream Clustering with Event Summaries. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Vinay Setty, Abhijit Anand, Arunav Mishra, and Avishek Anand. 2017. Modeling Event Importance for Ranking Daily News Events. In *WSDM '17: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 231–240.

Mihaela Simionescu and Javier Cifuentes-Faura. 2022. Can unemployment forecasts based on Google Trends help government design better policies? An investigation based on Spain and Portugal. *Journal of Policy Modeling*, 44(1):1–21.

Ian M. Soboroff, Iadh Ounis, and Craig Macdonald. 2010. Overview of the TREC 2008 Blog Track.

Fahim K Sufi. 2022. AI-GlobalEvents: A Software for analyzing, identifying and explaining global events with Artificial Intelligence. *Software Impacts*, 11.

Praneeth Vadlapati. 2024. AutoTrendyKeywords: Real-Time AI-Driven Trend-Based SEO Using LLMs. *International Journal of Computer Engineering and Technology*, 16(3):241–255.

Canhui Wang, Min Zhang, Liyun Ru, and Shaoping Ma. 2008. Automatic online news topic ranking using media focus and user attention based on aging theory. In *CIKM '08: Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1033–1042.

Investigating Hierarchical Structure in Multi-Label Document Classification

Artemis Dampa
Institute of Informatics and Telecommunications, NCSR "Demokritos", Greece
Department of Informatics and Telematics, Harokopio University of Athens, Greece
adampa@iit.demokritos.gr

Abstract

Effectively organizing the vast and ever-growing body of research in scientific literature is crucial to advancing the field and supporting scholarly discovery. In this paper, we study the task of fine-grained hierarchical multi-label classification of scholarly articles, using a structured taxonomy. Specifically, we investigate whether incorporating hierarchical information in a classification method can improve performance compared to conventional flat classification approaches. To this end, we suggest and evaluate different strategies for the classification, on three different axes: selection of positive and negative samples; soft-to-hard label mapping; hierarchical post-processing policies that utilize taxonomy-related requirements to update the final labeling. Experiments demonstrate that flat baselines constitute powerful baselines, but the infusion of hierarchical knowledge leads to better recall-focused performance based on use-case requirements.

1 Introduction

The exponential growth of scientific publications has created an urgent need for efficient indexing and organization of academic content. With vast and continuously expanding digital libraries, automatic categorization of scientific articles has become essential to facilitate effective search, discovery, and, ultimately, the acceleration of scientific research (Kim and Gil, 2019). This need is particularly acute in specialized domains, where researchers must navigate an increasingly dense body of literature.

In this work, we focus on the task of fine-grained, hierarchical multi-label classification of scholarly articles, experimenting on the field of Computational Linguistics. In Figure 1, we overview the hierarchical multi-label classification task. Given a document $d \in D$ where D is the set of all possible documents, and a set of labels $L = l_1, l_2, ...$

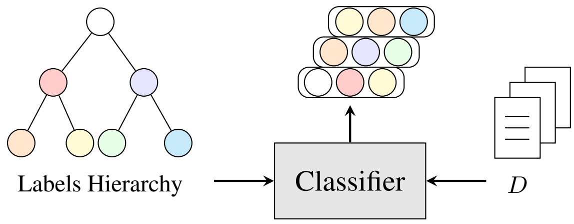


Figure 1: An overview of the hierarchical, multi-label classification task.

that have a hierarchical parent-child relation $P : L \times L \rightarrow \{0, 1\}$, where $P(x, y) = 1, x \in L, y \in L$ indicates that x is a parent of y, the task is to find a function $C : D \rightarrow \mathbb{S}(L)$ where $\mathbb{S}(\cdot)$ is the power-set operator, such that given a set of correct (but possibly a priori unknown) annotations $G : D \rightarrow \mathbb{S}(L)$, $C(x) = G(x), \forall x \in D$.

Our classification experimental setup uses a predefined taxonomy comprising a multitude of topics and subtopics (181 in total) (Ahmad et al., 2024a), offering a detailed and structured representation of research areas (see also Section 4). This setting poses unique challenges, elaborated on in Section 3, due to idiosyncrasies related to the assignment of a varying number of labels from each level of the hierarchy to a single document.

Historically, the scientific community has approached hierarchical classification using two broad strategies: flat classification (Barbedo and Lopes, 2006; Sun et al., 2003), where the hierarchical structure is ignored and each label is treated independently, or hierarchical classification (Zangari et al., 2024), where models exploit the parent-child relationships among labels to guide predictions. Although flat approaches simplify the problem and often yield strong baselines, they discard potentially valuable structural information. In contrast, hierarchical approaches preserve these relationships, offering a more semantically coherent labeling, but they are often sensitive to errors made at higher lev-

Proceedings of the Student Research Workshop associated with RANLP-2025, pages 10–19, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/issn.2603-2821.2025_002

els of the hierarchy, as such mistakes can propagate downward and lead to incorrect final predictions.

This paper presents a systematic study comparing flat and hierarchical (cascade-based) classification approaches in the context of scholarly document classification. Thus, we investigate whether exploiting hierarchical information leads to performance gains over flat baselines. Specifically, our contributions focus on three main axes:

Hierarchical Sampling We evaluate methods that enforce the hierarchical structure of the taxonomy by employing node-specific classifiers with hierarchy-aware negative sampling to respect the hierarchy during training.

Soft-to-Hard Label Mapping We explore heuristics to determine the optimal number of labels per document, based on taxonomy structure and empirical distribution. These heuristics include traditional threshold-based methods, fixed-number (top-k) strategies, and more recent LLM-based approaches that utilize generative models to infer the most contextually appropriate set of labels.

Hierarchy-enforcing Post-processing Policy We examine different approaches that ensure hierarchical consistency by altering predicted labels according to hierarchical constraints (assigning parent labels of predicted child nodes or removing child labels if their parents are not predicted).

To support our findings, we conduct statistical analyses that assess the significance of performance differences across multiple metrics. Our experiments reveal insights into the trade-offs between flat and hierarchical approaches and offer practical guidelines for choosing an appropriate strategy depending on the task constraints.

2 Related Work

The task of hierarchical multi-label text classification has seen significant progress through various approaches, each tackling challenges related to large-scale classification, label dependencies, and hierarchical structures.

In the work of Ahmad et al. (2024b), the authors introduce a hierarchical multi-label classification task in the field of computational linguistics. In this task, the authors offer a granular categorization approach based on the taxonomy provided in Ahmad et al. (2024a). The latter also offers a corpus of scholarly articles annotated with topics and subtopics drawn from a structured hierarchy of key NLP areas.

Several approaches have been proposed to handle multi-label text classification. Rajendram Bashyam and Krestel (2024) address hierarchical multi-label classification as extreme multi-label (XMC) flat classification problem, using an X-transformer designed for XMC (Zhang et al., 2021) and TF-IDF-based weak labeling, imposing hierarchy only post-prediction. Liu et al. (2017) introduce XML-CNN, a deep learning model designed for XMC. It enhances document representation using dynamic max pooling, binary cross-entropy loss, and a bottleneck layer to reduce model size. Another work (Hristov et al., 2021) also tackles clinical text classification as an extreme multi-label classification problem, using clustering and cluster-label mapping. S-GCN (Zeng et al., 2024) models multi-label text classification using a global graph based on words, texts, and labels co-occurrence, combining semantic encoding with graph convolution.

In hierarchical classification, Huang et al. (2019) propose a model that classifies documents at multiple levels by integrating text and hierarchy using a Hierarchical Attention-based Recurrent Layer. Similarly, Xu et al. (2021) employ a graph convolutional network (GCN) to learn associations between words, categories, and their relationships, incorporating correlations between levels. Tanigaki et al. (2024) introduce an integrated neural network with cascading self-attention mechanisms, where multi-head attention reconstructs text features at each level while a secondary network enforces inter-level dependencies. TELEClass (Zhang et al., 2025) tackles hierarchical text classification with minimal supervision by enriching the label taxonomy with the use of LLMs. Kosmpoulos et al. (2014) extend cascade classification for predicting the correct leaf of hierarchical structures by estimating the probability of each root-to-leaf path.

Although these works have made significant strides, they share common limitations. Flat classification methods often ignore the hierarchical relationships between labels, while cascade methods are prone to early misclassification. Additionally, many approaches assume a fixed number of labels per level, which does not capture the variability of label counts that can occur at different levels of the hierarchy. Our work aims to shed light on how to address these issues by exploring the effectiveness of hierarchical versus flat approaches in overcoming these challenges.

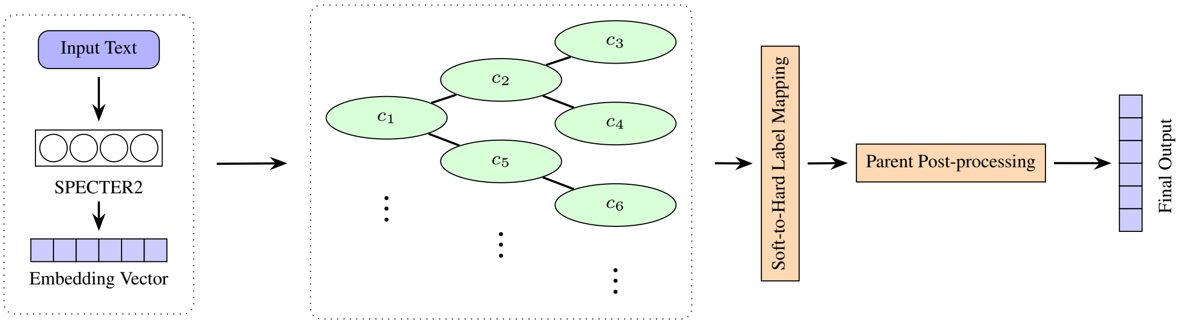


Figure 2: Diagram of the hierarchical multi-label classification process. The figure illustrates the stages of document representation, node-specific classifiers c_i training, soft-to-hard label mapping, hierarchy-enforcing post-processing, and generation of final output.

3 Methodology

In this section, we describe different methods to approach hierarchical multi-label text classification. However, unlike typical hierarchical classification (Sun and Lim, 2001), this task (a) allows the assignment of multiple labels per document; (b) labels can appear at any level of the hierarchy; (c) incomplete paths are allowed, i.e. there is no requirement for labeled documents to have leaf-only labels.

To address these challenges and effectively capture the nuanced structure of scientific discourse while respecting hierarchical label dependencies, our hierarchical approach combines pretrained document embeddings, node-specific classifiers trained using hierarchical sampling, label decoding strategies (soft-to-hard label mapping) and hierarchy-enforcing policies (see Figure 2) and is compared against its flat counterpart.

3.1 Document Representation

To obtain semantically rich document embeddings, we utilize SPECTER2-base (Singh et al., 2023), a pretrained transformer model designed for scientific documents. For each document, we concatenate the title, abstract, and selected metadata fields (author, year, venue, publisher and booktitle) as input to enhance representation. The input is preprocessed using the SPECTER2 tokenizer, with truncation and padding applied to ensure fixed-length. The resulting representation is derived from the model output layer, which captures a high-level summary of the document semantics.

3.2 Cascade Classification with Hierarchical Sampling and Flat Counterpart

Rather than training one multi-output flat classifier which ignores any hierarchical relationships between labels, we split the problem into multiple binary classification tasks, following a cascading approach inspired by Kosmpoulos et al. (2014), adapted for multi-label classification and multi-level label prediction, i.e. including internal nodes within a hierarchical label tree. For each category node c_i in the hierarchy, we train a dedicated logistic regression (LR) classifier to predict whether a document belongs to that category. All classifiers are trained independently. We choose LR due to its efficiency and ease of probabilistic interpretation.

A central challenge is to ensure that classifiers can distinguish semantically similar categories rather than simply separating positive examples from all negatives. To address this, we apply a hierarchy-aware sampling approach per classifier:

Positive samples Documents explicitly labeled with that node are selected as positives.

Negative samples Improve training effectiveness and respect the hierarchical structure as:

(a) **Sibling nodes:** documents labeled with sibling categories, that is, categories that share the same parent as the target node.

(b) **Parent-exclusive samples:** documents labeled with the parent category but not with the current node or any of its siblings.

(c) In cases where a node has no siblings, **siblings of the parent node** are used to maintain informative negative sampling, that is, documents associated with the siblings of its parent node.

The idea behind this design is to encourage the classifiers to focus on subtle inter-category distinctions, thereby enhancing their ability to capture fine-grained differences between closely related topics. By assigning to each classifier the task of distinguishing among a smaller set of categories, the approach also reduces computational resources required and overall classification complexity.

Algorithm 1: Training Hierarchical Multi-Label Classifiers

Input: Set of documents $D = \{d_1, ..., d_N\}$, Pretrained model M (SPECTER2), Hierarchical taxonomy H, Logistic Regression LR classifier
Output: Trained classifiers $C = \{c_1, ..., c_n\}$ for each node n in H

for *each node n in H* **do**
  $S^{(n)}_{pos} \leftarrow \{d \mid d \in n\}$
  $S^{(n)}_{sib} \leftarrow \{d \mid d \in siblings(n)\ , d \notin n\}$
  $S^{(n)}_{par} \leftarrow \{d \mid d \in parent(n)\ , d \notin n\}$
  $S^{(n)}_{neg} \leftarrow S^{(n)}_{sib} \cup S^{(n)}_{par}$
  if $S^{(n)}_{neg} = \emptyset$ **then**
    $S^{(n)}_{neg} \leftarrow \{d \mid d \in siblings(parent(n))\ , d \notin n\}$
  for d *in* $S^{(n)}_{pos} \cup S^{(n)}_{neg}$ **do**
    $X_d \leftarrow M(d)$
    $y_d \leftarrow 1$ if $d \in S^{(n)}_{pos}$, 0 if $d \in S^{(n)}_{neg}$
  end
  $c_n = LR(X, y)$
end

For the flat counterpart, we follow the same overall training strategy but omit the hierarchy-aware negative sampling, instead using the standard approach in which all samples not belonging to the target label serve as negatives for each classifier.

3.3 Soft-to-Hard Label Mapping

Each node classifier outputs a soft score in the range $[0, 1]$, indicating the model confidence that a document belongs to the corresponding category. To convert these scores into final hard label predictions, we propose three decoding *strategies*:

- **Threshold strategy:** A fixed confidence threshold $\theta \in [0, 1]$ is applied. Labels with scores above θ are selected.
- **Label number strategy:** A predefined number k of top-scoring labels is assigned per document.
- **LLM strategy:** A large language model ranks labels, optionally using predictions from the above strategies as priors.

We further discuss the selection of appropriate parameters in Section 4.

3.4 Hierarchy-enforcing Post-processing Policy

All strategies can include an additional hierarchy-enforcing step (hereafter referred to as *parent policy*) to guarantee valid hierarchical paths and adhere to the logical structure of topical taxonomies:

No-parents policy No post-processing is applied and predicted labels are left intact.

With-parents policy For a predicted label at any level, its ancestors are recursively included in the final label set (if not already present) to satisfy hierarchy constraints.

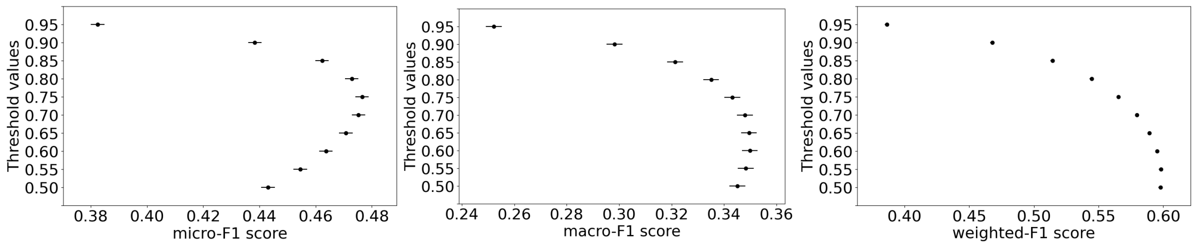
Strict policy A stricter approach keeps predicted labels only if all their parent labels are also predicted. This ensures more infusion of hierarchical structure but potentially introduces early misclassification errors.

Moderate policy A more moderate approach keeps labels if at least one of their parent labels are predicted, trying to balance flexibility and structural consistency.

4 Experiments

We evaluate our hierarchical multi-label classification approach on a corpus of approximately 42,000 scholarly articles from the ACL Anthology (Ahmad et al., 2024a; Rajendram Bashyam and Krestel, 2024) including title, abstract and various metadata such as authors, time of publication, publisher, book, venue. More specifically, our classifiers are trained on a joint set of 1,050 fully labeled documents from the collection and 41,107 weakly labeled documents, while 255 documents are reserved for additional testing. The classification task involves assigning each document to one or more relevant topics from a tree-structured taxonomy of 181 categories, organized across three levels. The train-test split of the dataset follows previous related work (Ahmad et al., 2024a; Rajendram Bashyam and Krestel, 2024) to offer comparable results.

We conduct experiments using 10-fold cross-validation (Sechidis et al., 2011) over the training data subset, with iterative stratification to ensure robust and representative evaluation under label imbalance and sparsity. This method extends traditional stratified sampling to multi-label data by ensuring that the distribution of labels is preserved



(a) Micro-F1 Scores (b) Macro-F1 Scores (c) Weighted-F1 Scores

Figure 3: F1 scores of hierarchical approach with threshold strategy for varying θ values between 0.5 and 0.95 applying no-parent policy analyzed using Tukey's HSD test for (a) micro-, (b) macro-, (c) weighted-F1

across folds, improving the fairness and consistency of training and evaluation splits. Model performance is also evaluated on a fixed test set of 255 scholarly documents, to demonstrate generalizability. For each node in the taxonomy, we train a binary classifier using LR (with the default parameters and a maximum of 1,000 iterations). Our training logic incorporates hierarchy-aware negatives, as detailed in Algorithm 1.

This experimental setup aims to answer the following research questions:

RQ1: Which proposed methods or parameter settings outperform the baselines and alternatives?

RQ2: How robust is each method with respect to its hyper-parameters?

RQ3: Can we pre-determine suitable hyper-parameters or develop heuristics to guide their selection?

RQ4: How can one encode hierarchical information in the learning process? Can this encoding improve the classification performance?

RQ5: How does the choice of hierarchical sampling impact model performance?

RQ6: What is the impact of different document representations on classification performance?

4.1 Baselines and Comparison

To benchmark the hierarchical approach, we compare against the following baselines:

- A SciNCL (Ostendorff et al., 2022) model fine-tuned on the flattened labels of the 1,050 labeled documents, which ignores the hierarchy, as provided by (Ahmad et al., 2024b).
- A dummy classifier, which selects labels randomly but preserving label frequency patterns. This serves as a weak lower-bound baseline.
- A flat approach employing a one-vs-all strategy, where a separate classifier is trained for each label using the same training dataset as the hierarchical model as described in Section 3.2.

These comparisons help establish the hierarchy-aware design performance relative to the other approaches (RQ1 - best method), thus evaluating how encoding hierarchical information affects classification performance (RQ4 - hierarchy infusion).

4.2 Label Selection Strategies

We explore the effect and performance of the three approaches described in Section 3.3 to convert classifier outputs into hard label predictions:

- Threshold strategy: Initially, we set the threshold $\theta = 0.6$ based on preliminary tests shown in Figure 3 without applying any parent policy and select all predicted labels with probabilities above θ. We later confirm this value as optimal through exhaustive search.
- Label number strategy: We analyze the label count distribution in the training set and set the number of labels k per document to 5, corresponding to both the mean and median of the distribution. This selection is further validated through an exhaustive search.
- LLM strategy: We use LLaMA 3.1 to validate label predictions based on content.

All strategies are tested across the different parent policies (RQ4 - hierarchy infusion). We vary our hyper-parameters, measuring impact on performance to assess sensitivity and validate RQ2 (hyper-parameters robustness).

Additionally, we conduct an oracle experiment on the validation: we assume knowledge of the true label count per sample and select the top-k predictions accordingly. This informs the feasibility of learning a meta-classifier to estimate label count per document (RQ3 - hyper-parameters heuristics).

Method	Micro Prec	Micro Rec	Micro F1	Macro Prec	Macro Rec	Macro F1	Weighted Prec	Weighted Rec	Weighted F1
Dummy	0.074	0.293	0.118	0.037	0.147	0.048	0.162	0.293	0.186
Flat SciNCL	0.356	0.328	0.341	0.016	0.046	0.024	–	–	–
Flat LR	**0.803**	0.601	**0.687**	**0.625**	0.392	**0.467**	**0.790**	0.601	**0.673**
Label ($k = 11$ & no-parents)	0.370	0.604	0.459	0.386	0.456	0.349	0.648	0.604	0.584
Label ($k = 7$ & with-parents)	0.302	0.552	0.391	0.352	0.383	0.297	0.509	0.552	0.479
Label ($k = 20$ & strict)	0.668	<u>0.679</u>	<u>0.673</u>	0.521	0.413	<u>0.446</u>	0.665	<u>0.679</u>	<u>0.664</u>
Label ($k = 20$ & moderate)	0.441	0.296	0.354	0.339	0.311	0.291	0.261	0.296	0.263
Threshold ($\theta = 0.6$ & no-parents)	0.368	0.628	0.464	0.375	<u>0.473</u>	0.350	0.647	0.628	0.595
Threshold ($\theta = 0.8$ & with-parents)	0.309	0.549	0.396	0.346	0.387	0.298	0.520	0.549	0.485
Threshold ($\theta = 0.5$ & strict)	<u>0.784</u>	0.588	0.672	<u>0.560</u>	0.348	0.420	<u>0.767</u>	0.588	0.653
Threshold ($\theta = 0.5$ & moderate)	0.582	0.249	0.349	0.400	0.261	0.279	0.315	0.249	0.261
LLM (Label $k = 20$ & with-parents)	0.606	0.520	0.560	0.468	0.381	0.393	0.646	0.520	0.552
Oracle Top-k (no-parents)	0.502	0.502	0.502	0.485	0.381	0.363	0.765	0.502	0.570
No strategy (no-parents)	0.327	**0.686**	0.443	0.343	**0.518**	0.345	0.602	**0.686**	0.598

Table 1: Evaluation of baseline and hierarchical methods, using micro, macro, and weighted precision, recall, and F1 score across 10-fold cross-validation. Best results per column are in bold, while second-best are underlined.

4.3 Ablation Studies

We perform analyses to identify factors influencing model performance. We examine the impact of negative sampling within our hierarchical framework and assess how the inclusion of metadata in document representation affects classification accuracy. These studies address RQ5 (negative sampling) and RQ6 (document representation) and provide insights to refine our approach and enhance model effectiveness (see Sections 5.3, 5.4).

4.4 Statistical Analysis and Evaluation

We assess model performance using the following metrics (Yang, 1999): (a) micro precision/recall/F1, which measure global performance, favoring frequent classes; (b) macro precision/recall/F1, which give equal weight to all classes, highlighting rare-category performance; (c) weighted precision/recall/F1, which weight the contribution of each label by its support.

For each configuration, results are aggregated over the cross-validation folds. To determine the statistical significance of differences between methods and parameter choices (RQ1 - best method), we perform Tukey's HSD test and report letter groupings to identify significantly different clusters.

Together, these experiments allow us to systematically address our research questions by comparing classifiers and selection strategies (RQ1 - best method), evaluating sensitivity to key parameters (RQ2 - hyper-parameters robustness), exploring document-specific heuristics for label prediction (RQ3 - hyper-parameters heuristics), and assessing the role of hierarchical information (RQ4 - hierarchy infusion), hierarchical sampling (RQ5 - negative sampling) and metadata enriched input (RQ6 - document presentation) in improving performance.

Method	micro-F1	macro-F1	weighted-F1
Label(NS)	0.673	0.446	0.664
Threshold(NS)	0.672	0.420	0.653
Label(RS)	0.590	0.422	0.560
Threshold(RS)	0.673	0.420	0.652

Table 2: Comparison of best-performing label selection strategies (strict $k = 20$, $\theta = 0.5$) with hierarchy-aware negative sampling (NS) against hierarchy-aware negative sampling enhanced with random sampling (RS).

5 Results

We present the performance of our hierarchical classification models using different label selection strategies and parent policies and compare them against flat and dummy baselines. Results are reported as average scores across 10-fold cross-validation in all tables, except Table 4, using micro, macro, and weighted precision, recall, and F1. All tables report statistically significant differences, validated using Tukey's HSD test (with $\alpha = 0.05$).

5.1 Overall Performance

Table 1 shows that the hierarchical method outperforms the flat SciNCL and dummy classifier across all metrics, but does not surpass the flat LR one-vs-all method in terms of precision and F1 scores. This provides an answer to RQ1 (best method), indicating that incorporating node-level classifiers does not always yield a performance advantage.

5.2 Strategies & Hyper-parameters

To explore RQ2 (hyper-parameters robustness), we varied hyper-parameters and observed that performance varied accordingly. Specifically, we performed an exhaustive search over different label counts (1-25) and threshold values (0.5-0.95) to identify the optimal values per strategy and policy.

The analysis on the label count showed that performance improved with an increasing number of labels, up to an optimal point beyond which gains began to diminish. This finding contradicts the initial belief that optimal performance would align with the mean or median of the label count distribution. Furthermore, variations in threshold values affected performance, with lower thresholds generally resulting in improved results. This outcome is expected, as higher confidence thresholds reduce the number of predicted labels, consequently leading to similar overall performance between the two strategies. This suggests that the method is sensitive to these hyper-parameters, with optimal performance achieved under specific conditions.

Based on this hyper-parameter tuning, we report the results for the best parameter values and policies in Table 1. The label number strategy with $k = 20$ and strict parents performed best overall. The threshold strategy with $\theta = 0.5$ and strict parents had slightly lower recall but higher precision, which can be advantageous in applications where minimizing false positives is critical. The LLM-based label selection strategy showed modest improvements for models with suboptimal hyper-parameter settings, but it significantly lagged behind the top-performing strategies. The setting without any strategy and parent policy improved recall but suffered from over-selection, leading to moderate F1. Across different strategies, enforcing parent policies most of the time boosted micro, macro and weighted F1 scores by at least 6% (up to 28%), 1% (up to 12%) and 6% (up to 9%), respectively, compared to their counterparts with no-parent policy, confirming the importance of structural consistency (RQ4 - hierarchy infusion).

To address RQ3 (hyper-parameters heuristics), we implemented an oracle strategy during validation that uses the true number of labels per document to select the top-k predictions. This approach achieved 36.31% macro-F1 which is over 8% lower than the best-performing hierarchical method. These results suggest that a meta-classifier for estimating label cardinality alone is not sufficient, and that label ranking combined with hierarchical structure infusion through parent policies plays a more critical role in achieving high performance. A heuristic based on the standard threshold of 0.5, performs competitively with our best hierarchical approach, supporting the idea that simple statistics can inform effective parameter choices when combined with hierarchical information. However, directly setting a predefined number of labels, which yields the best results within our hierarchical framework, can work best in combination with the parent policy that dynamically reduces the number of predicted values (i.e., by removing orphan child label predictions). As a result, the final label count per document varies, even though the initial number was fixed.

Method	micro-F1	macro-F1	weighted-F1
Flat simple	0.670	0.456	0.656
Flat enriched	0.687	0.467	0.673
Label simple	0.425	0.311	0.499
Label enriched	0.464	0.334	0.538

Table 3: F1 scores of flat and hierarchical (with $k = 7$ labels and no-parents policy) approaches using only title and abstract inputs, compared to metadata-enriched inputs, obtained through 10-fold cross-validation.

5.3 Hierarchical Sampling Study

Motivated by the relatively high recall that comes at the expense of precision, along with the generally increased number of positive predictions (both true and false), we conducted an additional study on the negative sampling strategy. In our hierarchical sampling approach, the lower levels of the hierarchy include fewer negative samples, which results in a distribution shift between the constructed training data and the original dataset.

To evaluate whether this imbalance affects model performance, we doubled the number of negative samples per classifier by randomly adding half of the samples from the full negative space. This adjustment was intended to test whether the initial assumption that the hierarchical structure would help the model better differentiate between similar documents holds true, or whether it instead introduces confusion, suggesting that the model might benefit more from increased exposure to diverse negative examples. The results of this study shown in Table 2, indicate that the best-performing strategies do not benefit from enhanced negative sampling, suggesting that the hierarchical frame-

Method	micro-F1	CLD	macro-F1	CLD	weighted-F1	CLD
Flat	0.669	A	0.374	A	0.676	A
Label ($k = 20$ with strict policy)	0.637	C	0.366	A	0.658	B
Threshold ($\theta = 0.5$ with strict policy)	0.657	B	0.349	B	0.661	B

Table 4: Comparison of the top-3 models on the test set in terms of micro, macro, and weighted F1 scores. Tukey's HSD significance test results: models sharing the same group letter are not significantly different at $\alpha = 0.05$. The CLD column (Compact Letter Display) shows the group letters assigned to each model.

work provides sufficient discriminative context.

5.4 Document Representation Study

To investigate the impact of input document representation on classification performance, we conducted a focused study comparing two alternative representations. Specifically, we aimed to assess whether including metadata fields enhances performance or whether a simpler representation suffices. To ensure a fair comparison, we kept all other components such as algorithm, label strategy, parent policy, and hyper-parameter values, constant, and varied only the document input.

The main representation used throughout this paper combines the title, abstract, and key metadata fields: author, year, venue, publisher, and booktitle. For comparison, we created a simplified version consisting of only the title and abstract concatenated. The results, presented in Table 3, clearly show that the metadata-enriched representation outperforms the simpler alternative on both flat and hierarchical approaches, confirming the value of incorporating contextual metadata in improving classification performance.

5.5 Statistical Significance

We applied Tukey's HSD post-hoc test to all configurations. Results in Table 4 are based on the 255 documents held out as the test set from the used corpus (Ahmad et al., 2024a). They indicate that the flat model forms a statistically superior group compared to the top-performing hierarchical models in terms of micro and weighted F1 scores. However, for macro F1, flat model belongs to the same significance group as the hierarchical with label count $k = 20$ and strict-parents policy.

6 Conclusion

In this work, we investigated the task of fine-grained hierarchical multi-label classification of scholarly articles, using a predefined taxonomy. We conducted a systematic comparison between flat classification methods and hierarchy-aware approaches, including cascade models with hierarchy-aware negative sampling and parent-enforcing post-processing. To this end, we utilized an existing corpus from NLP scholarly articles (ACL collection).

Our results demonstrate that the hierarchical approach outperforms the flat baseline in terms of recall but falls behind in precision and overall F1 score. While explicitly modeling the hierarchy adds complexity, enforcing hierarchy through the proposed parent policies generally improves performance compared to ignoring hierarchical structure.

Statistical analyses confirm that the observed differences are significant across most metrics, showing that hierarchy-aware strategies can help reduce false negatives. However, on the final test set, the hierarchical and flat approaches do not differ significantly in macro F1, suggesting that the hierarchical approach remains competitive when aiming for balanced performance.

Our study also demonstrated that the selection of a policy for the infusion of hierarchical information into classification significantly affects the result. Although the results we achieved with the most promising infusion policy were not sufficiently better from the flat approach, we argue that it is important to examine other approaches for this infusion.

Thus, as a future direction, more effective ways to represent and integrate hierarchical information should be explored. Motivated by the observed boost from metadata-enriched representations, incorporating knowledge-informed features may enhance the ability of the model to leverage hierarchy without relying solely on rigid label dependencies.

Acknowledgments

This work was supported by the European Union's Digital Europe Programme (DIGITAL) under grant agreement No 101146490. I would like to thank my supervisors G. Giannakopoulos, I. Varlamis and V. Karkaletsis for their valuable guidance throughout this work.

References

Raia Abu Ahmad, Ekaterina Borisova, and Georg Rehm. 2024a. Forc4cl: A fine-grained field of research classification and annotated dataset of nlp articles. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, page 7389–7394, Torino, Italia. ELRA and ICCL.

Raia Abu Ahmad, Ekaterina Borisova, and Georg Rehm. 2024b. Forc@nslp2024: Overview and insights from the field of research classification shared task. In *Natural Scientific Language Processing and Research Knowledge Graphs*, page 189–204, Cham. Springer Nature Switzerland.

Jayme Garcia sArnal Barbedo and Amauri Lopes. 2006. Automatic genre classification of musical signals. *EURASIP Journal on Advances in Signal Processing*, 2007(1):1–12.

Anton Hristov, Aleksandar Tahchiev, Hristo Papazov, Nikola Tulechki, Todor Primov, and Svetla Boytcheva. 2021. Application of deep learning methods to snomed ct encoding of clinical texts: From data collection to extreme multi-label text-based classification. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, page 557–565, Held Online. INCOMA Ltd.

Wei Huang, Enhong Chen, Qi Liu, Yuying Chen, Zai Huang, Yang Liu, Zhou Zhao, Dan Zhang, and Shijin Wang. 2019. Hierarchical multi-label text classification: An attention-based recurrent network approach. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, page 1051–1060, Beijing China. ACM.

Sang-Woon Kim and Joon-Min Gil. 2019. Research paper classification systems based on tf-idf and lda schemes. *Human-centric Computing and Information Sciences*, 9(1):30.

Aris Kosmpoulos, Georgios Paliouras, and Ion Androutsopoulos. 2014. The effect of dimensionality reduction on large scale hierarchical classification. In *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*, pages 160–171, Cham. Springer International Publishing.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 115–124, Shinjuku Tokyo Japan. ACM.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood contrastive learning for scientific document representations with citation embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, page 11670–11688, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lakshmi Rajendram Bashyam and Ralf Krestel. 2024. Advancing automatic subject indexing: Combining weak supervision with extreme multi-label classification. In *Natural Scientific Language Processing and Research Knowledge Graphs*, page 214–223, Cham. Springer Nature Switzerland.

Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2011. On the stratification of multilabel data. In *Machine Learning and Knowledge Discovery in Databases*, page 145–158, Berlin, Heidelberg. Springer.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2023. Scirepeval: A multi-format benchmark for scientific document representations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, page 5548–5566, Singapore. Association for Computational Linguistics.

Aixin Sun and Ee-Peng Lim. 2001. Hierarchical text classification and evaluation. In *Proceedings 2001 IEEE International Conference on Data Mining*, page 521–528.

Aixin Sun, Ee-Peng Lim, and Wee-Keong Ng. 2003. *Hierarchical Text Classification Methods and Their Specification*, page 236–256. Springer US, Boston, MA.

Koichi Tanigaki, Koji Cho, and Shuichi Tokumoto. 2024. Cascading taxonomic attention networks for hierarchical text classification. In *2024 16th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, page 369–372.

Linli Xu, Sijie Teng, Ruoyu Zhao, Junliang Guo, Chi Xiao, Deqiang Jiang, and Bo Ren. 2021. Hierarchical multi-label text classification with horizontal and vertical category correlations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, page 2459–2468, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yiming Yang. 1999. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1):69–90.

Alessandro Zangari, Matteo Marcuzzo, Matteo Rizzo, Lorenzo Giudice, Andrea Albarelli, and Andrea Gasparetto. 2024. Hierarchical text classification and its foundations: A review of current research. *Electronics*, 13(7).

Delong Zeng, Enze Zha, Jiayi Kuang, and Ying Shen. 2024. Multi-label text classification based on semantic-sensitive graph convolutional network. *Knowledge-Based Systems*, 284:111303.

Jiong Zhang, Wei-Cheng Chang, Hsiang-Fu Yu, and Inderjit Dhillon. 2021. Fast multi-resolution transformer fine-tuning for extreme multi-label text classification. In *Advances in Neural Information Processing Systems*, volume 34, page 7267–7280. Curran Associates, Inc.

Yunyi Zhang, Ruozhen Yang, Xueqiang Xu, Rui Li, Jinfeng Xiao, Jiaming Shen, and Jiawei Han. 2025. Teleclass: Taxonomy enrichment and llm-enhanced hierarchical text classification with minimal supervision. In *Proceedings of the ACM on Web Conference 2025*, page 2032–2042, Sydney NSW Australia. ACM.

Large Language Models for Lexical Resource Enhancement: Multiple Hypernymy Resolution in WordNet

Dimitar Hristov
Institute for Bulgarian Language
Bulgarian Academy of Sciences
dimitar@dcl.bas.bg

Abstract

Large language models (LLMs) have materially changed natural language processing (NLP). While LLMs have shifted focus from traditional semantic-based resources, structured linguistic databases such as WordNet remain essential for precise knowledge retrieval, decision making and aiding LLM development. WordNet organizes concepts through synonym sets (synsets) and semantic links but suffers from inconsistencies, including redundant or erroneous relations. This paper investigates an approach using LLMs to aid the refinement of structured language resources, specifically WordNet, by an automation for multiple hypernymy resolution, leveraging the LLMs semantic knowledge to produce tools for aiding and evaluating manual resource improvement.

1 Introduction

In recent years, an acceleration in the development of AI, machine learning and specifically generative models have greatly expanded the capabilities for solving tasks in the field of natural language processing (NLP). Large language models have proven to be a powerful tool for word sense disambiguation, sentiment analysis, abstractive summarization, paraphrasing with sentiment change, and other tasks.

The focus in natural language processing has in large part been shifted from development of structured language resources to the now more popular large language models. LLMs, which are themselves not just language resources but powerful often general-purpose tools, allow easy adaptability and specialization through fine tuning and prompt engineering.

There are, however, two reasons for the continued development of structured data resources. Such structured language data resources include the various forms of dictionaries - entry-based data with predefined parts such as word, inflection, definitions, examples - as well as ontology-based resources like WordNet (Miller et al., 1990; Fellbaum, 1998), BalkaNet (Tufis et al., 2004) and EuroWordNet (Vossen, 1998), incorporating vast amounts of knowledge with high accuracy. First and key in many spheres such as medicine and biology, structured data resources are deterministic, precise and validated. This ensures decisions are made on a consistent and provably correct data. This contrasts to the results from LLMs, where hallucinations - factually or logically unsound responses - occur to often for the extracted information to be readily usable without additional validation in high-stakes environments.

Additionally, this same power of LLMs is based on a very large preexisting knowledge base, which is incorporated in the model through training and fine-tuning. Existing structured language data resources are a significant knowledge-baring part of the training corpus for LLMs, meaning their continued development is essential for the progression of large language models. Even then, not all ontologies and knowledge bases have been used for model training.

The aim of this paper is to explore the viability of LLMs as tool for aiding and evaluation of structured language data enhancement with a focus on WordNet.

1.1 WordNet and multiple hypernymy

WordNet is an ontology-based structured language resource aiming to represent the interconnectedness of language concepts by constructing a network of concepts represented

Proceedings of the Student Research Workshop associated with RANLP-2025, pages 20–26, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/issn.2603-2821.2025_003

by synonym sets or synsets – sets of words or multi-word expressions with a common meaning – and the various semantic relationships between them. The resource has a graph-based structure well suited for deterministic approaches in NLP task solving.

One of the key ideas of WordNet is the codification of inheritance as hypernymy and hyponyny relations, linking a more general to a more specific concept - concept A (hyponym) is a type of concept B (hypernym), e.g., {bee:1} is a type of {hymenopterous insect:1; hymenopteran:1; hymenopteron:1; hymenopter:1}, which is itself a type of {insect:1}.

As any manually created database of knowledge, differences of language perception, ambiguity and other factors may occasionally cause errors in both the lexical data and the structure within WordNet (Richens, 2008; Verdezoto and Vieu, 2011). These can include: missing or erroneous words in the synset, errors in definition, synset ambiguity (one synset representing multiple concepts), multiple synsets for the same concept, wrong relation types, missing relations. Koeva and Hristov (2023) define one such potential issue - erroneous or extra hypernyms where no or other relations should be. They give a manually crafted dataset with resolved multiple hypernymy, resulting in a tree hypernymy structure, which requires further evaluation.

This paper will test whether the process of resolving multiple hypernymy can be automated through the use of LLMs and prompt engineering, evaluate the results and propose uses for LLMs in the WordNet improvement process.

1.2 Paper outline

Section 1 introduced the context and aim of the paper, while Section 2 links to the base research on which the task is defined. The methodology of the experiment, data and implementation are described in Section 3. Section 4 analyses the outputs and measurements of the results with a proposal for uses of the setup. Section 5 explores a list of potential improvements and extentions of the current work.

2 Related work

This paper looks into an approach to automate an otherwise manual task related to the creation and maintenance of structured language resources. In the particular task chosen for the experiment, the automated task is connected to the nature of hypernymy relations between synsets and their validity. A manual execution of multiple hypernymy resolution has been performed by Koeva and Hristov (2023) with promising results, invoking a question on whether such phenomena can be evaluated and modified in an automated or semi-automated way.

Lippolis et al. (2025) explore the automatic construction of an ontology draft using subtask-decomposed prompting, as well as prompting technique based on Chain Of Thought (CoT), where LLM inference is done separately on atomic data point - in this case competence questions, later merge together in a full ontology. A similar approach of dividing the problem into per-unit tasks is taken within the current work.

3 Methodology

The aim of the paper is to evaluate the effectiveness and efficacy of LLMs as a tool to aid with WordNet structural enhancement. This was achieved through emulating a standard workflow - solving hypernymy resolution tasks separately in a series.

3.1 Structure

The experiment is structured as a series of instruction-based multiple-choice tasks. The experiment is performed with generic out-of-the-box LLMs without any additional task-specific training or fine tuning. An inference is run for each separate synset with multiple hypernymy, using a prompt as described in A which provides:

1. General instructions - LLM's role (WordNet expert), task context (synsets and hypernymy relations), input format (how synset data is provided) and output format (a single synset ID);

2. Examples - this part is optional and is either missing (A.1 0-shot), or provides 1 or 5 examples (A.2 1-shot or few-shot);

3. The main task - a list of the current hypernym synsets, the question (Which synset above is the best hypernym?) and a description of the synset for which a hypernym is to be chosen.

3.2 Data

The experiment uses data on synset words, relations and meaning from Princeton WordNet 3.0. The data set was filtered to include only details on synsets with two or more hypernyms - a total of 1421 synsets - their word lists, meanings and hypernymy relations. As evaluation was done using the resulting data from Koeva and Hristov (2023), the data was synchronized, leaving only those synsets for which one of the already existing hypernyms was selected. Koeva and Hristov (2023) assigned a new hypernym to 77 of the synsets. Additionally, 5 synsets were selected to be used as examples in 1-shot and few-shot prompts, leaving 1339 synsets for evaluation.

The five manually chosen examples are:

1. Hyponym {mathematical space:1; topological space:1} "(mathematics) any set of points that satisfy a set of postulates of some kind" with hypernyms:
   - {space:1; infinite:2} "the unlimited expanse in which everything is located"
   - {set:41} "(mathematics) an abstract collection of numbers or symbols"
   - Chosen hypernym: {set:41}
2. Hyponym {Calamagrostis:1; genus Calamagrostis:1} "reed grass" with hypernyms:
   - {monocot genus:1; liliopsid genus:1} "genus of flowering plants having a single cotyledon (embryonic leaf) in the seed"
   - {genus:2} "(biology) taxonomic group containing one or more species"
   - Chosen hypernym: {monocot genus:1; liliopsid genus:1}
3. Hyponym {altar boy:1} "a boy serving as an acolyte" with hypernyms:
   - {acolyte:1} "someone who assists a priest or minister in a liturgical service; a cleric ordained in the highest of the minor orders in the Roman Catholic Church but not in the Anglican Church or the Eastern Orthodox Churches"
   - {male child:1; boy:3} "a youthful male person"
   - Chosen hypernym: {male child:1; boy:3}
4. Hyponym {potato:1; white potato:1; Irish potato:1; murphy:1; spud:4; tater:1} "an edible tuber native to South America; a staple food of Ireland" with hypernyms:
   - {starches:1} "foodstuff rich in natural starch (especially potatoes, rice, bread)"
   - {solanaceous vegetable:1} "any of several fruits of plants of the family Solanaceae; especially of the genera Solanum, Capsicum, and Lycopersicon"
   - {root vegetable:1} "any of various fleshy edible underground roots or tubers)"
   - Chosen hypernym: {solanaceous vegetable:1}
5. Hyponym {water:6} "a liquid necessary for the life of most animals and plants" with hypernyms:
   - {food:1; nutrient:1} "any substance that can be metabolized by an animal to give energy and build tissue"
   - {nutrient:2} "any substance (such as a chemical element or inorganic compound) that can be taken in by a green plant and used in organic synthesis"
   - {liquid:11} "a substance that is liquid at room temperature and pressure"
   - Chosen hypernym: {liquid:11}

3.3 Implementation

The experiment was implemented using scripts written in bash script or Python, Ollama[1] for local inference execution and the LangChain framework[2] with the LangChain Ollama

[1] https://ollama.com/
[2] https://www.langchain.com/

integration library for the application. Inference was done on four widely available LLMs - Google Gemma 3 with 4 billion parameters, Meta Llama 3.1 with 8 billion parameters, Mistral with 7 billion parameters and Microsoft Phi-4 with 14 billion parameters.

The models were retrieved from the Ollama model library as 4-bit quantized. The temperature (creativeness) setting was set to 0.7, while the number of examples was varied between none for zero-shot execution, 1 for 1-shot execution and 5 for few-shot execution, resulting in a total of 12 runs. In cases where the inference execution returned an invalid response, i.e., not a well-formatted synset ID or not the ID of one of the given hypernym synsets, up to two additional inferences were performed for the specific synset.

The code, data and generated results are available on GitHub[3].

4 Results and evaluation

The main measure used for the evaluation of the results from running the experiment was agreement - the ratio of synsets, for which an LLM has assigned the same hypernym as set in the manual dataset, or the ratio of synsets for which two LLMs have assigned the same hypernym. This measure shows generally whether LLMs' probabilistic generation can emulate a human's logic, and whether a confidence measure can be established for the LLM's results. All measurements are presented in Appendix B Agreement tables.

Tables 1, 2 and 3 present the agreement measure between each individual LLM and the manual dataset, as well as between each 2 LLMs. The measurements for agreement with the manual hypernymy resolution range between 45% and 55% regardless of number of examples or LLM, suggesting that no correlation is present between the manual approach and the LLM inference. However, the agreement between LLMs is consistently higher at 52.7-71.5% for 0-shot, 63.9-77.6% for 1-shot and 62.0-75.8% for few-shot. This suggests that (1) examples improves the understanding of the task, leading to more consistent results from LLMs, and (2) different LLMs may have more similar training data, most certainly all containing WordNet knowledge in addition to other publicly available datasets, while a human possesses different and additional knowledge, causing the consistency between LLMs and no apparent correlation between LLM results and the manual resolution.

Table 4 presents the ratio of synset hypernyms assignments for which there is a majority opinion - at least 3 of the 4 LLMs have proposed the same assignment. The results show Gemma 3 as an outlier, with participation in the majority for 66.8% of synsets for 1-shot inference, while other models agree with the majority for 76.9-79.1% of synsets. Tables 5 and 6 present the combined agreement of the LLMs with the manual data where (1) at least 3 LLMs have produced the same output, and (2) where all 4 LLMs have proposed the same resolution (unanimity). These tables show a potential for:

- using LLMs as a starting point for aided manual performance of hypernymy resolution, with a promising 47.9% unanimity, 36.3% non-unanimous majority and only 15.7% without agreement;
- using LLMs as an evaluation tool for a performed manual hypernymy resolution, focusing attention on cases where LLMs have a unanimous (22.8% for 1-shot) ana non-unanimous majority (19.7%) disagreement with the manual results.

5 Further work

The evaluation of the results of this study provide an overview of the potential use of LLMs in the improvement of structured language data resources. Several improvements can be made in the experiment to ensure consistency and validity of the model responses:

- addition of more and diverse LLMs - this will give more weight and granularity to the agreement measure;
- grouping of synsets by category, yielding more consistent logic with added information for the task;
- addition of human evaluation for both the original proposed resolution by Koeva and Hristov (2023) and the LLM results;

[3]https://github.com/DCL-IBL/SemNet

Kaplan and Schubert (2001); Gangemi et al. (2001) note that multiple hypernymy often encode other relation types, a case for further WordNet structure modifications. Koeva and Hristov (2023) explore this extension to the multiple hypernymy resolution - resolution of alternative relation types for existing hypernyms. This may be an additional target for LLM-aided enhancement and evaluation, using an improved variant of the experiment setup.

Acknowledgments

The present study is carried out within the project Infrastructure for Fine-tuning Pre-trained Large Language Models, Grant Agreement No. ПВУ - 55 from 12.12.2024 /BG-RRP-2.017-0030-C01/.

References

Christiane Fellbaum, editor. 1998. WordNet: An Electronic Lexical Database. Language, Speech, and Communication. MIT Press, Cambridge, MA.

Aldo Gangemi, Nicola Guarino, and Alessandro Oltramari. 2001. Conceptual analysis of lexical taxonomies: The case of wordnet top-level. CoRR, cs.CL/0109013.

Aaron Kaplan and Lenhart Schubert. 2001. Measuring and improving the quality of world knowledge extracted from wordnet.

Svetla Koeva and Dimitar Hristov. 2023. Resolving multiple hyperonymy. In Proceedings of the 12th Global Wordnet Conference, pages 343–351, University of the Basque Country, Donostia - San Sebastian, Basque Country. Global Wordnet Association.

Anna Sofia Lippolis, Mohammad Javad Saeedizade, Robin Keskisärkkä, Sara Zuppiroli, Miguel Ceriani, Aldo Gangemi, Eva Blomqvist, and Andrea Giovanni Nuzzolese. 2025. Ontology generation using large language models.

George A. Miller, Richard Beckwith, Christiane. Fellbaum, Derek Gross, and Katherine Miller. 1990. Introduction to WordNet: An On-line Lexical Database. International journal of lexicography, 3(4):235–244.

Tom Richens. 2008. Anomalies in the wordnet verb hierarchy. pages 729–736.

Dan Tufis, Dan Cristea, and Sofia Stamou. 2004. Balkanet: Aims, methods, results and perspectives – a general overview. Romanian Journal of Information Science and Technology Special Issue, 7:9–43.

Nervo Verdezoto and Laure Vieu. 2011. Towards semi-automatic methods for improving wordnet. In Proceedings of the Ninth International Conference on Computational Semantics, IWCS '11, page 275–284, USA. Association for Computational Linguistics.

Piek Vossen, editor. 1998. EuroWordNet: a multilingual database with lexical semantic networks. Kluwer Academic Publishers, USA.

A Prompts

A.1 0-shot

You are a WordNet expert. Your task is to evaluate hypernymy relations between semantic concepts. Each semantic concept is represented by a group of words with common meaning. This group is called a synset. If concept A is a hypernym of concept B, then concept B is a type of concept A, and concept A is a more generic version of concept B.

Each synset is presented by its ID, group of words and meaning. You will be given a synset and its hypernyms and will be instructed to choose a single hypernym.

Reply only with the chosen hypernym synset ID with format 30-<8 digits>-n and no other words. Do not give any reasoning and do not generate other text.

You are given the following synsets:
- ID (ID_a) with words $(words_a)$ and meaning $(definition_a)$

...
- ID (ID_x) with words $(words_x)$ and meaning $(definition_x)$

Which of the synsets (ID_a)... and (ID_x) is most likely to be the hypernym of synset (ID_{hypo}) defined as:
- ID (ID_{hypo}) with words $(words_{hypo})$ and meaning $(definition_{hypo})$

A.2 1-shot or few-shot

You are a WordNet expert. Your task is to evaluate hypernymy relations between semantic concepts. Each semantic concept is represented by a group of words with common meaning. This group is called a synset. If concept A is a hypernym of concept B, then concept B is a type of concept A, and concept A is a more generic version of concept B.

Each synset is presented by its ID, group of words and meaning. You will be given a synset and its hypernyms and will be instructed to choose a single hypernym.

Reply only with the chosen hypernym synset ID with format 30-<8 digits>-n and no other words. Do not give any reasoning and do not generate other text.

EXAMPLE [(n)]

You are given the following synsets:
- ID $(ID_a^{ex.n})$ with words $(words_a^{ex.n})$ and meaning $(definition_a^{ex.n})$

...
- ID $(ID_x^{ex.n})$ with words $(words_x^{ex.n})$ and meaning $(definition_x^{ex.n})$

Which of the synsets $(ID_a^{ex.n})$... and $(ID_x^{ex.n})$ is most likely to be the hypernym of synset $(ID_{hypo}^{ex.n})$ defined as:
- ID $(ID_{hypo}^{ex.n})$ with words $(words_{hypo}^{ex.n})$ and meaning $(definition_{hypo}^{ex.n})$

$(ID_{result}^{ex.n})$

...

TASK

You are given the following synsets:
- ID (ID_a) with words $(words_a)$ and meaning $(definition_a)$

...
- ID (ID_x) with words $(words_x)$ and meaning $(definition_x)$

Which of the synsets (ID_a)... and (ID_x) is most likely to be the hypernym of synset (ID_{hypo}) defined as:
- ID (ID_{hypo}) with words $(words_{hypo})$ and meaning $(definition_{hypo})$

B Agreement tables

0-shot	Manual	Gemma 3 4B	Llama 3.1 8B	Mistral 7B	Phi-4 14B
Manual	-	45.4%	50.4%	50.9%	54.1%
Gemma 3 4B	45.4%	-	55.4%	52.7%	57.6%
Llama 3.1 8B	50.4%	55.4%	-	71.5%	70.9%
Mistral 7B	50.9%	52.7%	71.5%	-	64.3%
Phi-4 14B	54.1%	57.6%	70.9%	64.3%	-

Table 1: Measures of agreement between LLMs and manual resolution for runs without examples

1-shot	Manual	Gemma 3 4B	Llama 3.1 8B	Mistral 7B	Phi-4 14B
Manual	-	53.3%	49.2%	48.7%	48.9%
Gemma 3 4B	53.3%	-	67.1%	65.8%	63.9%
Llama 3.1 8B	49.2%	67.1%	-	77.6%	76.9%
Mistral 7B	48.7%	65.8%	77.6%	-	76.4%
Phi-4 14B	48.9%	63.9%	76.9%	76.4%	-

Table 2: Measures of agreement between LLMs and manual resolution for runs with 1 example

Few-shot	Manual	Gemma 3 4B	Llama 3.1 8B	Mistral 7B	Phi-4 14B
Manual	-	51.0%	47.4%	50.1%	49.2%
Gemma 3 4B	51.0%	-	62.0%	63.6%	58.6%
Llama 3.1 8B	47.4%	62.0%	-	75.8%	71.9%
Mistral 7B	50.1%	63.6%	75.8%	-	69.7%
Phi-4 14B	49.2%	58.6%	71.9%	69.7%	-

Table 3: Measures of agreement between LLMs and manual resolution for runs with 5 examples

Majority	At least 3 LLMs agree	Gemma 3 4B	Llama 3.1 8B	Mistral 7B	Phi-4 14B
0-shot	69.9%	56.9%	64.4%	60.6%	62.1%
1-shot	84.3%	66.8%	79.1%	78.1%	76.9%
5-shot	81.6%	63.0%	75.7%	75.4%	70.9%

Table 4: Measures of existence and LLM agreement with majority

Manual	Majority (at least 3 LLMs)	Manual agrees	Manual disagrees	No majority
0-shot	69.9%	37.7%	32.2%	30.1%
1-shot	84.3%	41.8%	42.5%	15.7%
5-shot	81.6%	40.8%	40.8%	18.4%

Table 5: Measures of agreement of manual results with majority

Manual	Unanimity (all 4 LLMs)	Manual agrees	Manual disagrees	No unanimity
0-shot	34.3%	18.9%	15.4%	65.7%
1-shot	47.9%	25.2%	22.8%	52.1%
5-shot	40.3%	19.7%	20.5%	59.7%

Table 6: Measures of agreement of manual results with majority

Automated classification of causal relations. Evaluating different LLM performances.

Giacomo Magnifico
University of Tartu
Institute of Computer Science
Estonia
giacomo.magnifico@ut.ee

Abstract

The search for formal causal relations in natural language faces inherent limitations due to the lack of mathematically and logically informed datasets. Thus, the exploration of causal relations in natural language leads to the analysis of formal-logic-adjacent language patterns. Thanks to the recent advancements of generative LLMs, this research niche is expanding within the field of natural language processing and evaluation. In this work, we conduct an evaluation of 9 models produced by different AI developing companies in order to answer the question "Are LLMs capable of discerning between different types of causal relations?". The SciExpl dataset is chosen as a natural language corpus, and we develop three different prompt types aligned with zero-shot, few-shot, and chain-of-thought standards to evaluate the performance of the LLMs. Claude 3.7 Sonnet and Gemini 2.5 Flash Preview emerge as the best models for the task, with the respective highest F1 scores of 0.842 (few-shot prompting) and 0.846 (chain-of-thought prompting).

1 Introduction

Causality is a subject deeply related to human perception and nature, and for such reason it is inevitably tied to human bias and variable logical framework (Matute et al., 2015; Henne et al., 2021). The difficulty of the defining task of causality extractions comes from the difficulty in formalising causality itself. Such an endeavour persists in the scientific community within various fields, from social sciences to computer science, from psychology to mathematics. Important frames of reference have been given in works such as Pearl (2009) and Icard et al. (2017), which elaborate on the nature of causal relations and their underlying framework. Numerous works have investigated the mechanisms underlying logical frameworks of causality in various communicative contexts, with examples such as Henne et al. (2021) and Matute et al. (2015), and from text in natural language (Yang et al., 2022). Some contributions derive from the pioneering work of Lewis (1986) and further the examination of critical aspects of causality and cause-and-effect relationships specifically tied to natural language explanations; other contributions have focused on the interaction between causality and explainability, such as Jacovi et al. (2021) and Halpern and Pearl (2005). In recent years, the interactions between natural language and machine learning models have irreversibly changed with the advent of LLMs and generative models, such as the GPT, Llama, and Mistral lines of generative architectures. The paradigm has shifted to accommodate their existence, and research has begun to investigate the properties of LLMs and test their abilities (Ziyu et al., 2023), including their use for causal extraction within shorter and longer texts (Wang et al., 2024; Chatwal et al., 2025).

Our main research question is thus formulated: *When dealing with short, real-life examples, are LLMs capable of discerning between different types of causal bond within different topic domains?*

The paper moves forward from this research question by outlining relevant literature and evaluating properties of large language models, as reported in Section 2. Then, we present the choice of models produced by different AI leading companies in Section 3, along with the chosen SciExpl dataset of explanatory single-sentences (Magnifico and Barbu, 2025) in order to conduct the analysis. This is followed by a detailed analysis of the parameters selected as a standard for model calls in Section 4, with the details regarding prompt development and instructions being presented in the same section. Sections 5 and 6 close the paper with the results of the evaluation, an analysis of said results, and a summary of the work that has been done, including the limitations and weaknesses.

Proceedings of the Student Research Workshop associated with RANLP-2025, pages 27–36, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/issn.2603-2821.2025_004

The code for the entire evaluation pipeline has been rewritten in `ipynb` notebook form, and it is made available to the community through a dedicated repository [1] along with model outputs (both raw and cleaned).

2 Related Work

As research in causal inference has attempted to formalise the relationship between events and causes in the real world, the naturally occurring bias and lack of unified interpretations make it an especially difficult task (Pearl, 2009; Matute et al., 2015; Henne et al., 2021). Multiple formal models and datasets that make use of the markers of logical relations between elements in natural language already exist; some datasets present knowledge pairs with out-of-context information used to test commonsense reasoning (Rein et al., 2024), others make use of underlying causal graphs derived from natural language (Romanou et al., 2023). Others yet focus on in-domain analysis of question-answering and context-driven causation extraction, with one of the most recent examples being Moreno Sandoval et al. (2025). As this paper presents work derived from the causal analysis of natural language data by generative models, the frame of reference for the nature of causal relations is closer in concept to the relativity of causality (Icard et al., 2017). The proximity to Icard's work is in relation to the variability in causal strength and subjectivity, rather than the precise formality of such relations. This is further expressed in the analysis of the SciExpl dataset (Magnifico and Barbu, 2025) in the following section.

While the search for causal extraction methodologies is nothing new within the area of natural language processing (Khoo et al., 2000; Garcia, 1997), the advent of large language models has undoubtedly shaken the field. The LLM approach to causal inference and extraction in recent years has led to diverse results: while some works suggest that LLMs have only a mimicry of causal inference devoid of actual capabilities (Zečević et al., 2023), others suggest the possibility of causal inference (Kıcıman et al., 2023) and the distinction between causation and correlation. It should be noted, however, that while such a distinction is possible and LLMs can benefit from it, LLMs struggle with distinctions between general causal patterns and non-causal sentences (Jin et al., 2024). A generally agreed claim is that LLMs struggle with complex scenarios involving real-world events (Ashwani et al., 2024; Romanou et al., 2023), as finetuning happens with datasets aligned to specific causal directions in mind. Therefore, it is safe to claim that the use of specific causal datasets can lead to enhanced performance, especially in syntactically-similar tasks (Ashwani et al., 2024). Considering previous evaluation settings for LLMs and causal reasoning, such as Ziyu et al. (2023), this approach leverages the analysis of diverse causal sentences to focus on the capabilities of large architectures to distinguish between causal links of variable strength. Rather than a binary-link identification, the task changes to a classification of a spectrum of clearly identifiable different causal bonds, which can prove challenging for semantic and syntactic similarity. Compared to recent approaches that aim to extract cause-effect relations in a question-answering format (Chatwal et al., 2025), or that focus on document-level causal extraction (Wang et al., 2024), the aim of this work is pointed towards shorter inputs of more variable nature, with the same thorough analysis.

3 Materials

3.1 Data

The main drive for this work was to evaluate the efficiency in causal classification not only between causal and non-causal statements, but within the spectrum of causal bonds of different strength (Icard et al., 2017). As the intention was to evaluate the performance of generative models within the scope of natural language understanding with no forced question-answering format, the choice of dataset fell onto the SciExpl collection made available in Magnifico and Barbu (2025). The available dataset consists of 272 sentences in English, within topic domains within the area of biochemistry, annotated as different "explanation categories" by 120 annotators in total. Each sentence is classified with two labels according to Magnifico and Barbu (2025)'s different categorisation types, one for the explanation type (6 labels) and one for the causal link expressed in the sentence (3 labels). The latter is used for the purpose of this work, as the authors show that the inter-annotator agreement is more robust for that categorisation type (Krippendorf's alpha value of 0.667) and the category balance is split at 40% - 40% - 20% between the labels. The

[1] https://github.com/gima9552/LLM-Causality-Classification

following are the definitions for the labels used in the dataset, and an example sentence written ad hoc is provided.

- **Strong** causal links are expressed in sentences that present an explicit cause-effect relation, possibly detailing multiple intermediate steps in the causal process. The original explicit cause, as well as all the presented steps in the causal chain, ultimately lead to the presented effect. *"You are jittery because you drank too much coffee"*.
- **Weak** causal links are used to establish relationships between variables in the form of indirect or implicit causal relations, which may arise from mechanisms such as bias, intrinsic properties of entities, or hypothesised causal influence. While statistical correlations may sometimes point toward such bonds, they do not in themselves imply causality and should not be equated with causal relations. *"Oil paints are difficult to use, and many amateur artists prefer acrylics"*.
- **Contrastive** structures manifest in sentences that present multiple causal links, which could be both/either of the two aforementioned types. As multiple pairs of variables are presented as self-standing cause-effect relations, the different causal interactions are compared to one another to highlight how the differences in origin cause lead to differences in consequential endpoints. Although the contrast itself is not a causal bond, its hierarchical composition as a net of distinct causal structures vouches for a separate category. *"Eating chips makes you thirsty, while eating celery gives opposite results"*.

3.2 Large Language Models

The models chosen for the evaluation step were selected keeping in mind both their performance and their origin. Only one model for each major competitor in the field of generative AI was allowed, restricting the choice to their most popular large architecture according to user ratings [2]. This choice was motivated by the intention to mimic the average user's choice between the multiple options. The chosen LLMs are the following, presented in alphabetical order along with further information regarding their overall performance and peculiarities. Of important notice is that all of the performance reported is dated to May 2025, and the presence of newer iterations of the following architectures (as well as different datasets) might make the following information outdated.

Claude 3.7 Sonnet was one of the latest models in the Claude line produced by Anthropic, with less of a focus on mathematical reasoning and more "shifted toward the everyday occurrences" as mentioned in Anthropic (2025). The latest benchmark on GPQA scored 84.8% effectiveness.

Command R7B was one of the smaller models parameter-wise, being at the time the largest ever produced by Cohere. In the latest benchmark available, it was reported to be outperforming both Ministral 8B and Llama 3.1 8B on the GPQA set. (Cohere, 2024).

DeepSeek V3 0324 was the latest release by DeepSeek, performing very effectively on mathematical reasoning datasets. The benchmark was reported to be around 68.4% on GPQA-Diamond (DeepSeek-AI et al., 2024).

Gemini 2.5 Flash Preview 04-17 was the current preview iteration of Google's best-performing model, reported to have an integrated "thinking system" (Google, 2025). The latest benchmark performance on GPQA was reported at 82.8% .

GPT-4o-mini was one of the more affordable iterations of the GPT model by OpenAI for the general public, and one of their best-performing small models. It had recorded performances around 40% on the GPQA benchmark, but 80% on the MMLU benchmark for linguistic tasks (OpenAI, 2024).

Llama 3.3 70B Instruct was one of the large-range models developed by Meta, and it had recorded benchmark performances of 50.5% on GPQA-Diamond and 80% on MMLU (Meta, 2024).

Mistral Nemo was a small, lightweight language model built by Mistral AI in collaboration with NVIDIA, with a 68% benchmark on the MMLU dataset (AI, 2024).

Nova Lite 1.0 was one of the models from the Nova series by Amazon, with benchmark results comparable to GPT-4o and Claude 3.5 Haiku on both GPQA-Diamon and MMLU datasets (AWS, 2024).

Qwen3 235B A22B was the latest release from Qwen, with benchmark results comparable to

[2] https://openrouter.ai/rankings, https://lmarena.ai/leaderboard/text

the ones by DeepSeek and Claude on multiple benchmarks (Team, 2025).

In order to ensure a common processing baseline for all the large language models, and as much ease of reproduction of the evaluation as possible, we chose to deploy all the instances of model calls through the OpenRouter API (OpenRouter, 2023). The specifics regarding the parameters chosen for calling the models are presented in the following section.

4 Evaluation Setup

4.1 General Settings

Multiple parameter values had to be taken into account for each model call, resulting in the following decisions for the experimental setup. The *temperature* was set to 0 for each model call, to avoid incongruences between different runs and allow for the highest chance of reproducibility of results. The optional settings *max_tokens* and *response_format* were set to, respectively, 256 (when expecting longer types of output) and "`type`":"`text`" to ensure that a) multimodal architectures would output information in the correct format and b) limit the possible amount of tokens to avoid unexpected generation-loop issues. Every other setting, apart from the temperature, was left unchanged from the default values of the OpenRouterAPI calling functions. This choice was made in order to streamline the process between data input and output as much as possible, and limit human-biased alterations looking for the "optimal configuration" for each individual model, thus influencing each model's base performance. However, a fully deterministic output with a chosen *seed* was deemed as too restrictive and non-descriptive of both standard user behaviour and humanlike output; in order to counterbalance the variability in output quality, the reported results were aggregated from the best results from each architecture out of three separate runs.

The standard parameters *top_p, top_k, frequency_penalty, presence_penalty, repetition_penalty, min_p* and *top_a* fell back to the following values, in order: 1, 0, 0, 0, 0, 1, 0, 0.

4.2 Prompting Techniques

The base intuition was to provide the models with templates following a generally demonstrated curve in performance for LLM evaluations (Ziyu et al., 2023): from zero-shot equivalent to few-shot equivalent (Liu et al., 2024; Lee et al., 2023), followed by chain-of-thought reasoning (Cheng et al., 2024; Chatwal et al., 2025). Therefore, a set of three different templates was developed with the aforementioned properties in order to properly assess model performance when expecting different kinds of output and input complexity. In order to provide as objective an evaluation as possible, the prompts were designed to be neutral, short and direct. While this might have led to suboptimal results, using different prompts tailored to cater to the strengths of each individual model would have ultimately prevented an objective evaluation. A description of each prompt is available below, and each template is provided within the `ipynb` notebook in the GitHub repository [3].

- **Zero-shot Equivalent**. The template included the following information, in order: the model role ("*You are an expert in identifying causal links. Perform classification for an input sentence according to the following categories*"); a list with the definition of each category (with the format "***Category**:Definition*"); the input sentence to be evaluated; and the instruction to only output the name of the appropriate category for the evaluated sentence ("*Your response must ONLY be the name of the category the sentence belongs to. No other text or explanation*").

- **Few-shot Equivalent**. The template included the same information contained in the previous one, with the addition of an input and output example for each category of evaluation positioned after the definition list.

- **Chain-of-thought Reasoning**. Similarly to the Few-shot Equivalent, the template included the information contained in the Zero-shot Equivalent. The input and output sentences for each category were also included, with a slight change: instead of the output being only the category label, a sequence of reasoning steps that explained the choice of the category was provided. The final instruction of "only output the category label" was discarded, as it would have been counterproductive.

[3] https://github.com/gima9552/LLM-Causality-Classification

The example sentences were handpicked by the author and chosen for their ease of understanding and average length. All example sentences remained the same for both the few-shot and the chain-of-thought prompts.

5 Results and Analysis

In the previous section, the experimental setup was established and presented alongside the LLMs and the prompts used for each analysis setup. Here, we discuss the result obtained through a comparison between the labels produced by the LLMs and the human-annotated ground truth labels given in the dataset. In the case of chain-of-thought prompting, the answers were manually cleaned by the author, and only the final explicitly assigned label was used for the evaluation of model accuracy. Empty outputs, broken sentences, and additional hallucinated inputs (examples of which are provided in Table 2) were labelled as "no explanations", whereas hypotheticals that presented an explicit label were categorised accordingly. The results are reported by increasing prompt complexity, from *zero-shot* equivalent to *few-shot* equivalent and concluding with *chain-of-thought* equivalent. In Table 1, the F1 scores for the performance of all models are reported, divided by template.

model	zero	few	c-o-t
claude-3.7-sonnet	**0.688**	**0.842**	**0.816**
command-r-08-2024	0.504	0.654	0.658
deepseek-chat-v3-0324	0.588	0.684	0.710
gemini-2.5-flash-preview	**0.654**	**0.827**	**0.846**
gpt-4o-mini	0.596	0.676	0.721
llama-3.3-70b-instruct	0.636	0.724	0.746
mistral-nemo	0.551	0.735	0.353
nova-lite-v1	0.614	0.651	0.713
qwen3-235b-a22b	0.368	0.331	0.070

Table 1: Results of model evaluation expressed via micro-F1 score. The models are presented in alphabetical order, and the two highest scores per template are in bold.

Despite the difference in size, production date, knowledge cutoff and performance on other tasks, almost all LLMs performed above the `0.50` threshold when prompted with the zero-shot equivalent template. The only notable exception was Qwen3, which encountered issues with the token-generation limit and produced enough empty outputs to underperform compared to the average performance value. As the results provided in Table 1 show, Claude and Gemini were the best-performing models within this category, closely followed by Llama 3.3 70B.

With the inclusion of example sentences in the prompts, the performance of nearly all LLMs improved by a minimum of `0.037` (Nova Lite), with the largest margin of improvement shown by Mistral Nemo (`0.184`). However, compared to the previously clean outputs, this prompt template led to the occasional generation of input sentences along with the required output. A striking example was Nova Lite hallucinating 8 extra input sentences, which were recognised as additions upon reading the model output, and failing to score them appropriately. Both Command and Deepseek produced empty output lines, which were labelled as "no explanation", and Qwen3 presented the same generation-limit issues previously reported. We decided not to alter the token limitation, as the overarching rule of the template of "only providing the category label" as an output was still standing. On the other side of the output spectrum, both GPT-4o and Claude generated outputs that most closely followed the given directions.

As the prompt-induced rule of "only providing the category label" was removed with the *chain-of-thought* equivalent template, all the outputs required manual postprocessing before an automated evaluation to determine the F1 score. Where Claude, Command, Gemini, GPT, and Nova provided a single-sentence output as presented in the template, DeepSeek, Mistral, and Llama 3.3 produced long token sequences that did not adhere to the suggested pattern. Furthermore, a common occurrence for both Deepseek and Nova was to stray from the "single label" instruction by proposing alternatives and hypotheses (e.g. "However, if the sentence focused on [...] the label should be [...]"). Regarding specific types of hallucinations, the last entry in Table 2 is peculiar, as Deepseek provided two different answers with two different ratings without any instruction to do so. Ultimately, it seemed that the *chain-of-thought* prompt caused the highest rate of issues and hallucinations by the models, including:

- the information presented in the chain-of-thought output not corresponding to the model-assigned label;
- multi-labelling, with assigned labels being contradictory at times;

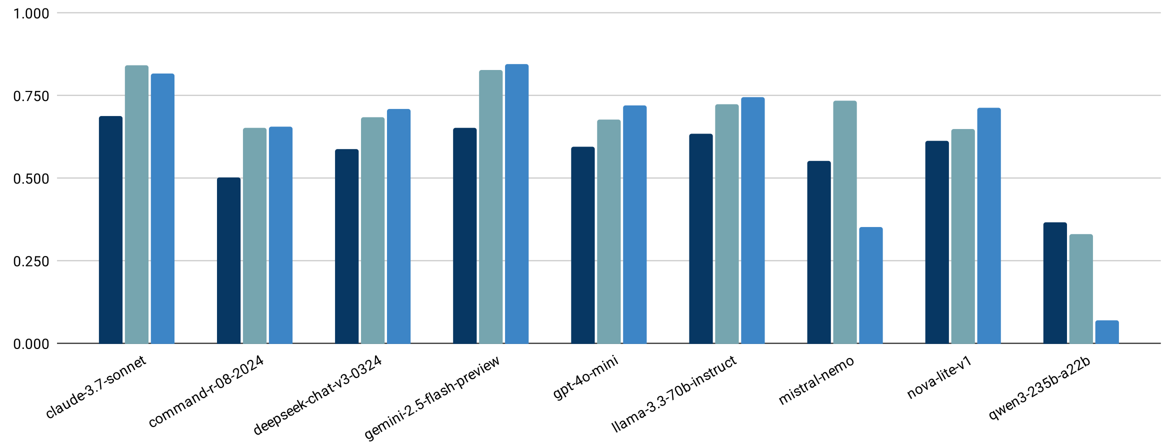


Figure 1: Chart presenting the F1 score for all models in alphabetical order. Each column represents a different prompt, left to right: zero-shot, few-shot, chain-of-thought.

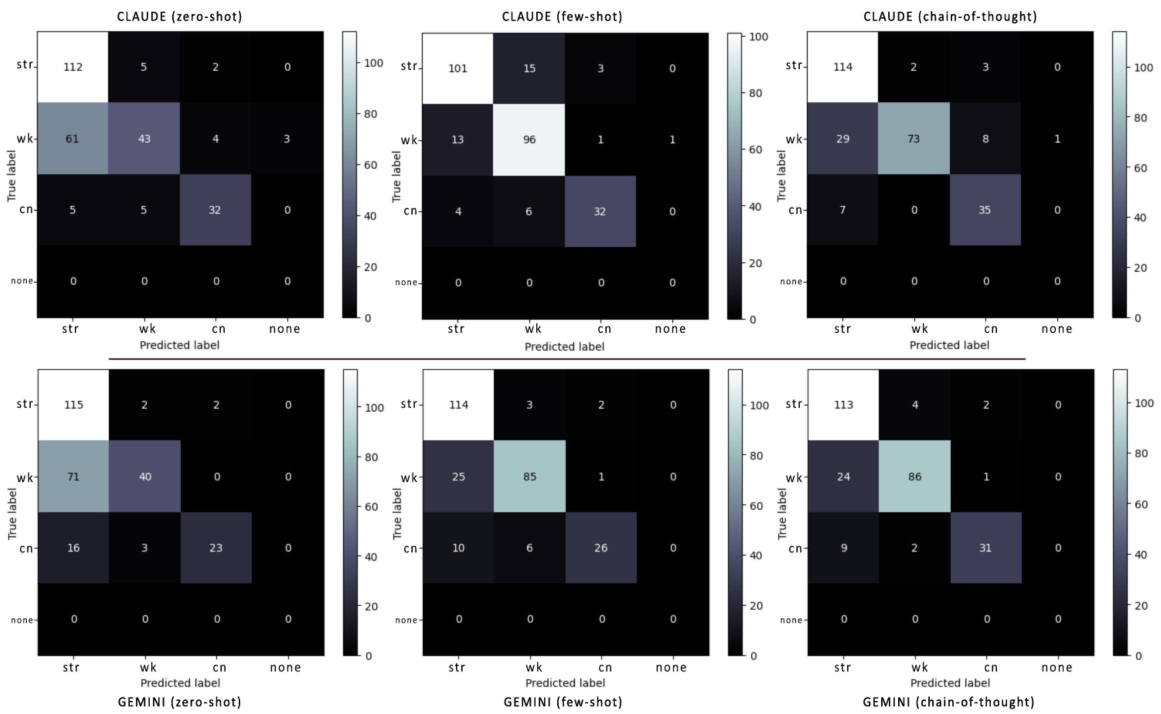


Figure 2: Confusion matrices reporting precision/recall for the two best-performing models, Claude and Gemini, divided by prompt type. The labels *str, wk, cn, none* correspond to *strong, weak, contrastive, no explanation*.

- empty outputs, likely due to token-generation loops;
- half-sentence outputs / well-formed sentences but no labels.

An example list of errors derived by the chain-of-thought prompts is available in Table 2, presenting the model output on the right side and the type of error defined in the middle.

However, as presented in Table 1 and shown in Figure 1, the performance of the majority of the models did not deteriorate despite the presence of prompt-related issues, showing in fact a positive trend. The exceptions to this were the Claude architecture with a 0.028 minor loss, and both Mistral (0.382 drop) and Qwen3 (0.261 drop) were plagued by empty outputs and broken sentences, possibly due to token-generation issues due to the longer prompt and the lack of text-explicit limitation. Overall, the results appeared in line with general knowledge and previous surveys of LLM evaluations, the use of *chain-of-thought* prompting generally favoured more complete and understandable outputs, with the drawback of context-loss and

model	error type	example
qwen3-235b-a22b	broken sentence	The sentence explains a cause-effect relation where the spines of cacti (cause) directly
nova-lite-v1	extra tokens	**Output**: The sentence explains that broad spectrum antibiotics are used due to the unavailability or delay in obtaining culture data that identifies the infecting bacteria. This indicates a relationship between the unavailability of specific information and the decision to use broad spectrum antibiotics. The category is: Weak Causation **Explanation**: The sentence provides a reason for using broad spectrum antibiotics, but it does not detail a direct cause-effect relationship or multiple steps in the causal process. Instead, it emphasizes the role of the unavailability of specific information, which is an indirect cause. **Explanation**: The sentence provides a reason for using broad spectrum antibiotics, but it does not detail a direct cause-effect relationship or multiple steps in the causal process. Instead, it emphasizes the role of the unavailability of specific information, which is an indirect cause.
mistral-nemo	misclassification, extra tokens	The sentence presents a contrast between two different environments: temperate forest and desert. It explains the outcomes for the fox population in each environment (thriving in the first, going extinct in the second) due to the availability of resources, which is a causal relation. However, the sentence does not go into the details of the causal process like the previous strong causation example. Therefore, the category is: **Weak Causation**. In detail: - The sentence presents a contrast between two environments: temperate forest and desert. - The outcome for the fox population in each environment is explained: thriving in the temperate forest and extinction in the desert. - The reason for these outcomes is the availability of resources in each environment, presenting a causal relation. - The sentence does not delve into multiple steps in the causal process, unlike the strong causation example, making it weak causation.
llama-3.3-70b-instruct	extra tokens, hypothetical	The sentence explains a cause-effect relation between the climate and the migration patterns of birds. The birds in the northern hemisphere migrate due to the lack of food and suitable breeding grounds during winter, whereas the birds in tropical regions do not migrate because the consistent climate provides stable resources. The category is: **Strong Causation**. However, it can also be argued that the sentence presents a comparison between two different situations (birds in the northern hemisphere vs birds in tropical regions), which could classify it as **Contrastive**. But since the primary focus is on explaining the cause-effect relation between climate and migration patterns, the more suitable category would be **Strong Causation**.
deepseek-chat-v3-0324	misclassification, extra tokens	**Output**: The sentence explains a cause-effect relation between sunlight and plant growth, detailing how ample sunlight enhances photosynthesis and promotes growth, while insufficient light limits energy production and growth. It also contrasts two scenarios (sunny field vs. shaded forest) to highlight the opposing outcomes. The primary focus is on the causal mechanism, with a secondary contrastive element. The category is: **Strong Causation**. *(Rationale: The sentence primarily describes a direct causal process (sunlight $\rightarrow$ photosynthesis $\rightarrow$ growth), which fits "Strong Causation." While it includes a contrast (sunny vs. shaded), the contrast serves to reinforce the causal explanation rather than being the main focus.)*
deepseek-chat-v3-0324	misclassification, extra tokens, hypothetical, input duplication	**Output:** The sentence explains a cause-effect relation between the degeneration of neurons in the hippocampus and the impairment of memory and spatial navigation in Alzheimer's disease, while contrasting it with the functionality in healthy brains. The category is: **Strong Causation** (due to the clear mechanistic explanation of neurodegeneration leading to cognitive deficits, with a secondary contrastive element). Alternatively, if the **contrastive** aspect is emphasized more strongly: The sentence compares the effects of hippocampal degeneration in Alzheimer's disease (impaired memory) with the preserved function in healthy brains (clear memory processing), presenting two opposing scenarios. The category is: **Contrastive**. *(The classification depends on whether the primary focus is the causal mechanism within Alzheimer's or the comparison between disease and health. Here, the causal chain is more detailed, so "Strong Causation" is likely the better fit.)*

Table 2: Examples of erroneous outputs from chain-of-thought prompting, with the types of error explicited in the middle column.

heightened focus of the generation of text based on specific token-cues from the prompt with the lengthening of the outputs.

By scrutinising the performance of the Gemini 2.5 and Claude 3.7 Sonnet models, the two best-performing ones, it is possible to highlight the weak points in causal categorisation previously assessed in Jin et al. (2024). As seen in Figure 2, for both models, there was consistent misclassification of *weak causation* and *contrastive* labels as *strong causation*, represented by *wk, cn* and *str*, respectively. While this effect might lead to think that the dataset is unbalanced, the split between the labels is 40/40/20% with the most represented categories being *strong causation* and *weak causation*; therefore, the misclassification is probably indicating a lack of effectiveness by the architectures in discering the less syntactic-oriented types of causal links. It is worth mentioning that the few-shot performance of Claude 3.7 Sonnet (upper centre in Figure 2) presents a generalisation of the issue on both sides of the causal-strength spectrum, as there is almost equal misclassification of strong and weak causal bonds. Despite this weak point still being present in the evaluated LLMs, it is clearly visible in the provided confusion matrices that the accuracy in the classification of different causal bonds with varying strength is promisingly high.

6 Conclusions

This paper presented the evaluation of 9 different Large Language Models, of diverse proprietary nature, as classifiers of causal bonds between sentences in a natural language dataset. Three prompt templates were developed and used to gradually increase context and provide the models with additional information and point verbal reasoning capabilities in the appropriate direction. When tested with zero-shot, few-shot, and chain-of-thought-based prompts, the best models performed with an F1 score of 0.688, 0.842 (Claude 3.7 Sonnet zero and few-shot), and 0.846 (Gemini 2.5 Flash chain-of-thought). The pipeline code has been rewritten as a `ipynb` notebook and made available at a dedicated repository [4].

Our evaluation seems to reinforce the hypothesis that LLMs can discern different types of sentence-internal causal bonds, more reliably so if provided with example-based prompts (few-shot) that provide an explicit definition of such causal properties. This implementation suggests that there is no real need for more complicated chain-of-thought prompts when it comes to the analysis and explanation of sentences from a classifying standpoint; however, manual analysis of the chain-of-thought outputs reveals that for the best-performing models, the generated "line of thought" is correct on average despite the occasional hallucination. The proposed idea in previous works that LLMs struggle with precise distinction between correlation and causation is reinforced by confusion-matrix analysis, as the misclassification of the two is manifest even in the best output cases. The dataset taken into consideration, albeit simple on a semantic level and of reduced size, is still based on the general communication patterns that humans use; this implies some generalizable proficiency by the LLMs to operate effectively in everyday causal-analysis situations.

Further research work is required to reinforce the hypotheses confirmed by this paper, especially relating to the ability of LLMs to truly identify underpinning causal links. Possible avenues of research should be directed to the analysis of causal generation in a contextless environment, more so than testing models on what could be a consequence of mere token-context performance. The use of effective state-of-the-art models, rather than the popular ones due to price or ease of implementation, should also be reinforced in future research. Furthermore, the use of model-tailored prompts instead of general ones, with more stringent instructions, could help prevent hallucinations as much as possible. Possibly, all models should be tested on multiple natural-language causal datasets, with the best case scenario allowing for different domains to avoid topic specificity, and several runs of the same model on standardised settings would be necessary.

As can be inferred from above, this work presents some clear limitations: the choice of a semantically simple dataset, the choice of models based on user popularity, the use of single-sentence classification examples, and the grounding of the causal analysis in natural language relations. The latter, especially, is an intrinsic hurdle when applying concepts of causality to information disjointed from the statistically-informed causal graphs and relations. Furthermore, the implementation of more in-depth chain-of-thought prompt choices and more complex methods of analysis might have led to a

[4]https://github.com/gima9552/LLM-Causality-Classification

different set of results, alongside the presentation of fully context-free samples.

References

Mistral AI. 2024. Mistral nemo.

Anthropic. 2025. Claude 3.7 sonnet and claude code.

Swagata Ashwani, Kshiteesh Hegde, Nishith Reddy Mannuru, Dushyant Singh Sengar, Mayank Jindal, Krishna Chaitanya Rao Kathala, Dishant Banga, Vinija Jain, and Aman Chadha. 2024. Cause and effect: Can large language models truly understand causality? *Proceedings of the AAAI Symposium Series*, 4(1):2–9.

Amazon AWS. 2024. Amazon nova foundation models.

Pulkit Chatwal, Amit Agarwal, and Ankush Mittal. 2025. Enhancing causal relationship detection using prompt engineering and large language models. In *Proceedings of the Joint Workshop of the 9th Financial Technology and Natural Language Processing (FinNLP), the 6th Financial Narrative Processing (FNP), and the 1st Workshop on Large Language Models for Finance and Legal (LLMFinLegal)*, pages 248–252, Abu Dhabi, UAE. Association for Computational Linguistics.

Xiaoxue Cheng, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2024. ChainLM: Empowering large language models with improved chain-of-thought prompting. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2969–2983, Torino, Italia. ELRA and ICCL.

Cohere. 2024. Introducing command r7b: Fast and efficient generative ai.

DeepSeek-AI, :, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. 2024. Deepseek llm: Scaling open-source language models with longtermism.

Daniela Garcia. 1997. Coatis, an nlp system to locate expressions of actions connected by causality links. In *Knowledge Acquisition, Modeling and Management*, pages 347–352, Berlin, Heidelberg. Springer Berlin Heidelberg.

Google. 2025. Gemini 2.5 flash documentation.

Joseph Y Halpern and Judea Pearl. 2005. Causes and explanations: A structural-model approach. part ii: Explanations. *The British journal for the philosophy of science*.

Paul Henne, Kevin O'Neill, Paul Bello, Sangeet Khemlani, and Felipe De Brigard. 2021. Norms affect prospective causal judgments. *Cognitive Science*, 45(1):e12931.

Thomas F. Icard, Jonathan F. Kominsky, and Joshua Knobe. 2017. Normality and actual causal strength. *Cognition*, 161:80–93.

Alon Jacovi, Swabha Swayamdipta, Shauli Ravfogel, Yanai Elazar, Yejin Choi, and Yoav Goldberg. 2021. Contrastive explanations for model interpretability. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1597–1611, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhijing Jin, Jiarui Liu, Zhiheng LYU, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona T. Diab, and Bernhard Schölkopf. 2024. Can large language models infer causation from correlation? In *The Twelfth International Conference on Learning Representations*.

Christopher S. G. Khoo, Syin Chan, and Yun Niu. 2000. Extracting causal knowledge from a medical database using graphical patterns. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, page 336–343, USA. Association for Computational Linguistics.

Emre Kıcıman, Robert Osazuwa Ness, Amit Sharma, and Chenhao Tan. 2023. Causal reasoning and large language models: Opening a new frontier for causality. *Trans. Mach. Learn. Res.*, 2024.

Unggi Lee, Haewon Jung, Younghoon Jeon, Younghoon Sohn, Wonhee Hwang, Jewoong Moon, and Hyeoncheol Kim. 2023. Few-shot is enough: exploring chatgpt prompt engineering method for automatic question generation in english education. *Education and Information Technologies*, 29(9):11483–11515.

David Lewis. 1986. Causal explanation. In David Lewis, editor, *Philosophical Papers, Volume II*, pages 214–240. Oxford University Press.

Yilun Liu, Shimin Tao, Weibin Meng, Feiyu Yao, Xiaofeng Zhao, and Hao Yang. 2024. Logprompt:

Prompt engineering towards zero-shot and interpretable log analysis. In *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings*, ICSE-Companion '24, page 364–365, New York, NY, USA. Association for Computing Machinery.

Giacomo Magnifico and Eduard Barbu. 2025. Emerging categories in scientific explanations.

Helena Matute, Fernando Blanco, Ion Yarritu, Marcos Díaz-Lago, Miguel A. Vadillo, and Itxaso Barberia. 2015. Illusions of causality: how they bias our everyday thinking and how they could be reduced. *Frontiers in Psychology*, Volume 6 - 2015.

Meta. 2024. Llama 3.3: Model cards & prompt formats.

Antonio Moreno Sandoval, Jordi Porta, Blanca Carbajo Coronado, Yanco Torterolo, and Doaa Samy. 2025. The financial document causality detection shared task (FinCausal 2025). In *Proceedings of the Joint Workshop of the 9th Financial Technology and Natural Language Processing (FinNLP), the 6th Financial Narrative Processing (FNP), and the 1st Workshop on Large Language Models for Finance and Legal (LLMFinLegal)*, pages 214–221, Abu Dhabi, UAE. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4o mini: advancing cost-efficient intelligence.

OpenRouter. 2023. Openrouter documentation.

Judea Pearl. 2009. Causal inference in statistics: An overview. *Statistics Surveys*, 3(none):96 – 146.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Angelika Romanou, Syrielle Montariol, Debjit Paul, Leo Laugier, Karl Aberer, and Antoine Bosselut. 2023. CRAB: Assessing the strength of causal relationships between real-world events. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15198–15216, Singapore. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3: Think deeper, act faster.

Zimu Wang, Lei Xia, Wei Wang, and Xinya Du. 2024. Document-level causal relation extraction with knowledge-guided binary question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16944–16955, Miami, Florida, USA. Association for Computational Linguistics.

Jie Yang, Soyeon Caren Han, and Josiah Poon. 2022. A survey on extraction of causal relations from natural language text. *Knowl. Inf. Syst.*, 64(5):1161–1186.

Matej Zečević, Moritz Willig, Devendra Singh Dhami, and Kristian Kersting. 2023. Causal parrots: Large language models may talk causality but are not causal. *Transactions on Machine Learning Research*, 2023(8).

Zhuang Ziyu, Chen Qiguang, Ma Longxuan, Li Mingda, Han Yi, Qian Yushan, Bai Haopeng, Zhang Weinan, and Ting Liu. 2023. Through the lens of core competency: Survey on evaluation of large language models. In *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 2: Frontier Forum)*, pages 88–109, Harbin, China. Chinese Information Processing Society of China.

Study on Automatic Punctuation Restoration in Bilingual Broadcast Stream

Martin Polacek
AILab@TUL
FM TUL
Liberec, Czech Republic
martin.polacek@tul.cz

Abstract

In this study, we employ various ELECTRA-Small models that are pre-trained and fine-tuned on specific sets of languages for automatic punctuation restoration (APR) in automatically transcribed TV and radio shows, which contain conversations in two closely related languages. Our evaluation data specifically concerns bilingual interviews in Czech and Slovak and data containing speeches in Swedish and Norwegian. We train and evaluate three types of models: the multilingual (mELECTRA) model, which is pre-trained for 13 European languages; two bilingual models, each pre-trained for one language pair; and four monolingual models, each pre-trained for a single language. Our experimental results show that a) fine-tuning, which must be performed using data belonging to both target languages, is the key step in developing a bilingual APR system and b) the mELECTRA model yields competitive results, making it a viable option for bilingual APR and other multilingual applications. Thus, we publicly release our pre-trained bilingual and, in particular, multilingual ELECTRA-Small models on HuggingFace, fostering further research in various multilingual tasks.

1 Introduction

In recent years, multiple automatic speech recognition (ASR) systems have been developed, becoming integral to our daily interactions with technology. These systems are now widely used in virtual assistants, automated transcription tools, and numerous other applications that convert spoken language into text.

A key factor driving this widespread adoption has been the development of advanced deep learning methods, particularly end-to-end (E2E) systems (Li, 2022). Unlike traditional ASR approaches that require separate stages for feature extraction, acoustic modeling, and language modeling, E2E systems adopt a more streamlined architecture. They directly map audio inputs to textual outputs, reducing complexity and often improving accuracy. This breakthrough has enabled the creation of ASR models for many languages (Toshniwal et al., 2017), broadening worldwide access to speech recognition technology.

Despite these advancements, some ASR systems still face significant challenges, one of the most notable being the absence of punctuation marks in their output. This limitation arises primarily from the nature of training data, which sometimes does not include punctuation information. Consequently, some ASR models then produce a continuous stream of words without the linguistic boundaries necessary for clear and structured text.

The lack of punctuation negatively impacts both user experience and downstream tasks. For example, in live captioning scenarios, the absence of sentence boundaries can make text difficult to read and understand, particularly in fast-paced or complex dialogues. To solve this issue, modules for automatic punctuation restoration (APR) are usually employed at the output of many ASR systems. In most cases, however, these modules are pre-trained for only one target language, preventing them from correctly formatting the output of multilingual ASR systems (Li et al., 2022) that can process data streams containing utterances in more than one language.

2 Motivation for this work

This work focuses on a specific task of APR in transcribed TV/R (TV and radio) streams containing speech in two similar languages. This phenomenon occurs often in neighboring countries (regions) where people speak a similar or mutually intelligible language.

Proceedings of the Student Research Workshop associated with RANLP-2025, pages 37–43, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/issn.2603-2821.2025_005

For example, the Czech and Slovak Republics formed one state, Czechoslovakia, between 1918 and 1992; many people born in one country now live in the second one. The two languages are thus similar in that native speakers of Czech understand Slovak and vice versa. The situation is similar in Scandinavia. Here, the population speaks many related languages and dialects, the most widespread of which is the triplet comprising Swedish, Danish, and Norwegian. Norwegian has many similarities with the first two languages, so a native Norwegian speaker can understand Danish and Swedish. Therefore, a Norwegian TV program may often feature a person speaking Swedish or Danish. A third example is the former Yugoslavia, which includes mainly Croatia, Serbia, Bosnia and Herzegovina, and Montenegro. The people living here speak mutually intelligible languages belonging to the western branch of the South Slavic languages.

TV/R programs in these regions containing speech in more than one language are often bilingual. These are typically interviews or talk shows in which the invited person or presenter speaks a different language from the invited guest. In the Czech Republic, for example, there are many interviews with Slovak guests on the Czech television station DVTV. On Slovak television, on the other hand, many Czech guests have appeared on the talk show "Trochu inak s Adelou". Another example is the popular talk show Skavlan, broadcast on Norwegian, Swedish, and Danish television between 2009 and 2021. The Norwegian presenter Fredrik Skavlan invited various speakers of different Scandinavian languages to the show.

Finally, it should be noted that the issue of transcription and APR in bilingual streams also relates to the task of live subtitling of various conferences or social events. For example, it is common for a conference held in the Czech Republic to feature speakers of Czech and Slovak and vice versa.

3 Related work and our contribution

The first developed APR methods were purely statistical. Their biggest drawbacks were their heavy dependence on the quality of the ASR output and low robustness to words outside the system dictionary. The latter problem is becoming increasingly acute with the shift from dictionary-based ASR models to end-to-end (E2E) systems.

In the next phase of development, recurrent neural networks (RNNs) have begun to be used (Kim, 2019), which have shown significantly better performance and allowed the incorporation of both textual and prosodic features. However, their use poses a challenge, especially regarding efficient training data preparation. With the advent of attention mechanism-based transformers, the BERT architecture (Devlin et al., 2019) was among the first used for APR. It outperforms the models with LSTM (Hochreiter and Schmidhuber, 1997) and BiLSTM layers (Tilk and Alumae, 2015) by more than 30% (Polacek et al., 2023). All the previously mentioned models were pre-trained (and fine-tuned) for only one language. However, in 2019, multiple languages were combined during pre-training to create the M-BERT (Pires et al., 2019) model, which can understand the word-to-word connections between languages and thus works for, e.g., the speech translation task.

In this work, we take advantage of the ELECTRA-Small architecture, which achieved better results for the APR task than the BERT model in our previous study (Polacek et al., 2023); we train one multilingual model for 13 selected European languages and two bilingual models Czech-Slovak (CZ+SK) and Norwegian-Swedish (NO+SE) to enhance language modeling across closely related languages by leveraging shared linguistic features. We then investigate the performance of all these models fine-tuned for APR on bilingual corpora and compare their results to monolingual models that are fine-tuned on the same data. For this evaluation, we utilize a dataset containing bilingual Czech-Slovak and Norwegian-Swedish texts. Our results show that pre-training and fine-tuning on data belonging to both languages are necessary to achieve the best performance and that the difference between the results of monolingual and bilingual models increases with the bigger the language difference. We also make public pre-trained multilingual and bilingual models on the HuggingFace platform (Wolf et al., 2020) under a CC-BY-4.0 license:

- Czech-Slovak bilingual model [1].
- Norwegian-Swedish bilingual model [2]
- Multilingual model (mELECTRA) for main European languages [3]

[1] https://huggingface.co/AILabTUL/BiELECTRA-czech-slovak

[2] https://huggingface.co/AILabTUL/BiELECTRA-norwegian-swedish

[3] https://huggingface.co/AILabTUL/mELECTRA

In particular, we believe that a multilingual model can be useful for the community since no one has yet made the ELECTRA architecture in a multilingual version public. Its training requires a lot of data and significant computational resources.

4 Adopted bilingual & multilingual models

As aforementioned, based on our previous research in (Polacek et al., 2023), where we investigated multiple transformer-encoder model types (Vaswani et al., 2017) for the APR task, the neural network architecture adopted in this work corresponds to the ELECTRA-Small model (Clark et al., 2020). This model is complemented with a classification head consisting of one feed-forward layer and one linear layer. The feed-forward layer takes a feature vector of size 256 on input and produces a feature vector of size 512 after passing through the SELU(Klambauer et al., 2017) activation function. The second linear layer produces a probability score for 4 classes (none, dot, comma, and question mark).

4.1 Tokenization

For tokenization of all models, we employ the SentencePiece tokenizer (Kudo and Richardson, 2018) with a vocabulary size of 30,525 as the authors in (Polacek et al., 2023). The amount of data used to create the vocabulary is the same for each language and corresponds to the smallest amount available, i.e., 3.75 GB for Portuguese (the size of training data for each language is summarized in Table 1). This approach ensures an equal data distribution for all languages, mitigating token imbalance. Note that all punctuation marks (e.g., period, question mark, and comma) and numbers are defined as separate tokens during tokenization.

The numbers in Table 2 and Table 3 for Czech and Swedish, respectively, then underscore the efficiency of leveraging language similarity when constructing tokenizers. For example, mixing Slovak and Czech data to create single tokenizer just slightly increases the total token count for the Czech text corpus by 1.2%. Similarly, mixing Norwegian and Swedish data to create a single tokenizer increases the token count on Swedish data by 5.6%, which shows that Swedish and Norwegian are slightly more distant languages than Czech and Slovak.

At the same time, the multilingual tokenizer, which supports 13 languages, extends the token count by just 32.1% on Czech data and by 33.5% on Swedish data. These numbers suggest that the multilingual tokenizer maintains a reasonable balance between flexibility for multiple languages and efficiency for individual ones, making it well-suited for multilingual applications.

4.2 Pre-training and data used

We followed the pre-training procedure outlined for the ELECTRA model in (Clark et al., 2020), adapting it for our multilingual setup. As mentioned, the multilingual model was trained using data from 13 languages, with their representation determined by the availability of language-specific data to ensure a fair and meaningful comparison with single-language models. This approach allowed us to balance the model's capacity to generalize across multiple languages while preserving its performance for individual languages.

The primary data source for each language was transcriptions from TV/R broadcasts, which provide a rich and diverse representation of spoken language. As a complement, smaller portions of the dataset included newspaper articles and legal texts. This data added variety and improved the model's understanding of different text domains. Details of the pre-training data are shown in Table 1. Note the total dataset size was 34.8 GB for the Czech-Slovak (CZ+SK) model, 13.26 GB for the Norwegian-Swedish (NO+SE) model, and 131.51 GB for the multilingual model.

Data preparation for pre-training involved careful processing to ensure consistency. First, all tokenized samples (each containing sentences from a single language) were combined into a unified dataset. These samples were then shuffled across languages to ensure balanced representation within batches. This batching strategy facilitated the model's exposure to diverse linguistic patterns during pre-training, helping it learn shared representations.

For both pre-training and fine-tuning, we used a single NVIDIA H100 GPU (80 GB VRAM) and 16 GB of system RAM. The pre-training of each ELECTRA-Small model took approximately 40 hours. Fine-tuning took about 6 hours per model. All training was performed with mixed-precision (FP16).

Language	SE	SL	SK	PT	PL	NO	IT	HR	FR	EN	DK	DE	CZ
Size [GB]	4.29	6.82	10.9	3.75	11.1	8.97	9.41	14.2	15.7	7.24	4.33	10.9	23.9

Table 1: Summary of training data available for individual languages

Tokenizer	Number of tokens [%]
CZ	100.0
CZ+SK	101.2
Multilingual	132.1

Table 2: Percentage of tokens for Czech data using different tokenizers; CZ tokenizer is the 100% baseline.

Tokenizer	Number of tokens [%]
SE	100.0
NO+SE	105.6
Multilingual	133.5

Table 3: Percentage of tokens for Swedish data using different tokenizers; SE tokenizer is the 100% baseline.

5 Experimental results

5.1 Evaluation metrics

To evaluate the model performance, we used the F1 metric, a commonly used measure combining precision and recall (Van Rijsbergen, 1979). This metric was specifically calculated for classes representing punctuation marks, such as commas, periods, and question marks, as these are critical for assessing the model's ability to predict punctuation correctly. The evaluation utilized a weighted average approach, where the contribution of each class to the final F1 score was proportional to its frequency in the dataset. This ensures that more frequent punctuation marks, which have a greater influence on the overall performance, also have a greater impact on the final score.

To prevent distortion of the results, the ”None” class, representing the absence of punctuation, was excluded from the evaluation. Since this class is typically dominant in the dataset, it would disproportionately inflate the F1 score, masking the model's true ability to predict punctuation marks accurately.

5.2 Data used for evaluation

Our data for CZ/SK evaluation consists of manually corrected transcripts of monolingual Czech TV/R and Slovak TV/R news, bilingual interviews with Czech moderators, and Slovak guests (from station DVTV) and bilingual interviews with Slovak moderators, and Czech guests (from the talk show ”Trochu inak s Adelou”). This bilingual set is publicly available[4],

For NO/SE evaluation, we also used monolingual Swedish and Norwegian TV/R news transcripts. However, suitable transcriptions for the bilingual scenario were unavailable, as those of bilingual shows exist only in a variant translated to one of the languages. To overcome this issue, we created a synthetic dataset simulating bilingual interviews on various topics: we utilized OpenAI's GPT-4o model (OpenAI, 2023) to generate artificial conversations. First, multiple interview topics were selected, including sports, weather, traveling, culture, gastronomy, hobbies, cooperation, etc. Their generation was then initiated using the prompt:

> ”Generate a conversation where one paragraph is in Norwegian and the other paragraph is in Swedish and alternate like this. Write only the paragraphs and generate a long interview. The topic is [TOPIC]”

In total, we created 156 artificial interviews. The resulting bilingual set is also made public[5].

5.3 Effect of data size for fine-tuning

The first performed experiment investigates the effect of the amount of data used for fine-tuning. We fine-tuned the model for APR using the methodology described in our previous work (Polacek et al., 2023), with the only difference in the amount of data used. The results for the Czech ELECTRA model are summarized in Table 4. They show that only 100 MB of data is sufficient for achieving optimal performance. Note that this experiment was performed on the Czech part of the development set described in Section 5.2.

[4]https://owncloud.cesnet.cz/index.php/s/HHfTnWK8D3202Q2
[5]https://owncloud.cesnet.cz/index.php/s/WzqYFR0e1HWbJ66

Table 4: Results of APR after using datasets of various sizes for fine-tuning

# tokens (data size)	P[%]	R[%]	F1[%]
4.2M (25 MB)	74.0	68.8	71.3
8.4M (50 MB)	74.3	69.1	71.6
12.6M (75 MB)	75.3	73.1	74.2
16.8M (100 MB)	**76.4**	**75.5**	**75.9**
33.6M (200 MB)	74.8	74.3	74.5
50.4M (300 MB)	74.6	75.0	74.8
67.2M (400 MB)	75.6	75.5	75.5
84.0M (500 MB)	74.7	74.6	74.6
168.0M (1000 MB)	75.0	74.2	74.6
252.0M (1500 MB)	75.4	75.2	75.3
336.0M (2000 MB)	74.4	74.4	74.4
420.0M (2500 MB)	74.9	74.9	74.9

For bilingual and multilingual models, we selected 100 MB of text data for each language (i.e., 200 MB in total for each bilingual model). To mimic realistic scenarios such as bilingual debates, we split the fine-tuning corpus into individual sentences and randomly mixed them into batches. Additionally, we applied a preprocessing step where, in 50% of the samples, 1–3 words were removed from the beginning and the end of the sequence. This approach improves training data variability and ensures that not all training sequences represent complete sentences.

5.4 Results on Czech and Slovak data

In Table 5, we report the results of the next performed experiment: first, the SK model was fine-tuned on Slovak data only (a), and the CZ model was fine-tuned on Czech data only (b). Next, both models were fine-tuned on a combined CZ+SK dataset (c,d). Subsequently, the mELECTRA model was fine-tuned on the same CZ+SK dataset (e), and finally, the bilingual CZ+SK model underwent fine-tuning on the same data (f).

The yielded results highlight the importance of using data for fine-tuning from both languages intended for inference. This fact follows from the first and second rows, where fine-tuning for just one language leads to a significant performance drop for the second one. In other words, the SK model (a) shows a 15.7% decrease in F1 score on the CZ evaluation dataset compared to the SK model fine-tuned on CZ+SK data (c), and the CZ model (b) shows a 15.9% decrease in F1 compared to the CZ model fine-tuned on CZ+SK data (d). These results also confirm that fine-tuning on both languages yields a noticeable improvement in performance. Furthermore, the results in Table 5 also show very good performance of the mELECTRA model, which, after fine-tuning on Czech and Slovak data, yields just slightly worse F1 values than the best-performing models pre-trained for a single language.

pre-training	fine-tun.	CZ F1 [%]	SK F1 [%]
(a) SK	SK	59.9	71.2
(b) CZ	CZ	**77.8**	57.0
(c) SK	CZ + SK	75.6	**74.4**
(d) CZ	CZ + SK	76.2	72.9
(e) mELECTRA	CZ + SK	76.4	74.1
(f) CZ + SK	CZ + SK	76.0	73.1

Table 5: Comparison of various APR models on monolingual Czech and Slovak datasets

The next experiment, see Table 6, presents the results yielded on bilingual transcripts of TV/R interviews. The obtained results reveal that the performance of all evaluated models is similar, emphasizing the key role of fine-tuning, which enables the models to adapt effectively to the specifics of the target dataset and to mitigate differences arising from pre-training. Notably, the mELECTRA model (e), despite being pre-trained for many languages, performs only 0.5% worse than the best model (f). This small gap demonstrates, similarly to the previous experiment, the potential of a general-purpose multilingual model, which can eliminate the need for pre-training language-specific models for the APR task.

pre-training	fine-tun.	CZ+SK F1 [%]
(a) SK	SK	62.9
(b) CZ	CZ	61.2
(c) SK	CZ + SK	66.6
(d) CZ	CZ + SK	66.3
(e) mELECTRA	CZ + SK	66.2
(f) **CZ + SK**	**CZ + SK**	**66.7**

Table 6: Comparison of various models on a bilingual Czech/Slovak dataset

5.5 Results on Norwegian and Swedish data

Similar to the previous evaluations, we conducted experiments using models pre-trained on a single language (Norwegian or Swedish), on both languages (NO+SE), and using all available multilingual data (mELECTRA). From Table 7, it is obvious that the performance on the language, which was not used for fine-tuning, drops by 44.6% for model (a) on Norwegian data and by 47.3% for model (b) on Swedish data. On the contrary, the use of both languages for fine-tuning yields significant improvements in performance for both languages. Specifically, model (c) has an F1 of 1.8% higher on Norwegian data compared to the single-language model (a). Similarly, under the same conditions, model (d) achieved a 0.5% increase in F1 over model (b). The mELECTRA model (e) performed comparably to model (d) on Norwegian data, while its accuracy on Swedish data was only 2.8% lower. The best-performing model (f) was found to be only 2.2% less accurate than model (c) on Norwegian data and 0.9% less accurate than model (d) on Swedish data, demonstrating its competitive performance across both languages.

pre-training	fine-tun.	NO F1 [%]	SE F1 [%]
(a) NO	NO	71.1	19.9
(b) SE	SE	23.8	64.5
(c) NO	NO + SE	**72.9**	52.3
(d) SE	NO + SE	66.6	**65.0**
(e) mELECTRA	NO + SE	67.0	62.2
(f) NO + SE	NO + SE	70.7	64.1

Table 7: Comparison of APR models on monolingual NO and SE datasets

The last experiment presented in Table 8 shows the results for the artificially created bilingual Norwegian/Swedish dataset. Here, it is evident that model (c) achieved the worst results and models (d) and (e) yielded (as expected from the previous experiment) similar F1 values of 71.6% and 71.7%, respectively. The best results were obtained by using model (f). However, its F1 value of 74.2% is just by 2.5% higher than that of the mELECTRA model (e). This means that the model pre-trained on all multilingual data proves good performance not only for the Czech/Slovak data but also for Norwegian/Swedish bilingual interviews.

pre-training	fine-tun.	NO+SE F1 [%]
(a) NO	NO	67.5
(b) SE	SE	64.1
(c) NO	NO + SE	67.8
(d) SE	NO + SE	71.6
(e) mELECTRA	NO + SE	71.7
(f) **NO + SE**	**NO + SE**	**74.2**

Table 8: Comparison of various models on a bilingual Norwegian/Swedish dataset

6 Conclusion

This study demonstrates the necessity of fine-tuning bilingual and multilingual models for APR in bilingual ASR outputs. Our findings indicate that as the difference between the two languages increases, the need for fine-tuning using data belonging to both of them becomes crucial. Without adequate fine-tuning, performance for the untrained language drops significantly.

At the same time, we show that the more the languages differ from each other, the more important the data for pre-training is. For the Czech/Slovak data, there was almost no improvement with pre-training on both languages, whereas, for the Norwegian/Swedish pair, there was already a more than 2% improvement. In other words, using bilingual data only for fine-tuning does not guarantee the best results.

Furthermore, in specific cases such as Slovak, fine-tuning of monolingual, bilingual, and multilingual models on CZ+SK datasets resulted in performance improvements compared to training solely on SK data. This suggests that leveraging linguistic similarities between closely related languages can enhance model robustness and effectiveness beyond single-language training.

Lastly, our study identifies a promising alternative in the use of a pre-trained multilingual model. This type of model can achieve competitive performance with only 100MB of fine-tuning data. This efficiency makes multilingual models an attractive solution for handling APR in bilingual as well as monolingual streams.

7 Acknowledgements

This work was supported by the Student Grant Scheme at the Technical University of Liberec through project nr. SGS-2024-3425. Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Seokhwan Kim. 2019. Deep recurrent neural networks with layer-wise multi-head attentions for punctuation restoration. In *ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 7280–7284. IEEE.

Gunter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-normalizing neural networks. In *NIPS 2017, Long Beach, CA, USA, December 4-9, 2017*, pages 971–980.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP 2018, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. ACL.

Bo Li, Tara N. Sainath, Ruoming Pang, Shuo yiin Chang, Qiumin Xu, Trevor Strohman, Vince Chen, Qiao Liang, Heguang Liu, Yanzhang He, Parisa Haghani, and Sameer Bidichandani. 2022. A language agnostic multilingual streaming on-device asr system. In *Interspeech*.

Jinyu Li. 2022. Recent advances in end-to-end automatic speech recognition. *APSIPA Transactions on Signal and Information Processing*, 11(1):1–64.

OpenAI. 2023. Gpt-4 technical report.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001. Association for Computational Linguistics.

Martin Polacek, Petr Cerva, Jindrich Zdánský, and Lenka Weingartová. 2023. Online punctuation restoration using electra model for streaming asr systems. *INTERSPEECH 2023*.

Ottokar Tilk and Tanel Alumae. 2015. LSTM for punctuation restoration in speech transcripts. In *Interspeech 2015, Dresden, Germany, September 6-10, 2015*, pages 683–687. ISCA.

Shubham Toshniwal, Tara N. Sainath, Ron J. Weiss, Bo Li, Pedro J. Moreno, Eugene Weinstein, and Kanishka Rao. 2017. Multilingual speech recognition with a single end-to-end model. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4904–4908.

C. J. Van Rijsbergen. 1979. *Information Retrieval*, 2nd edition. Butterworths, London.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Thomas Wolf, Lysandre Debut, Victor Sanh, and et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

NoCs: A Non-Compound-Stable Splitter for German Compounds

Carmen Schacht
Ruhr-University Bochum, Germany
Faculty of Philology
Department of Linguistics
carmen.schacht@rub.de

Abstract

Compounding—the creation of highly complex lexical items through the combination of existing lexemes—can be considered one of the most efficient communication phenomenons, though the automatic processing of compound structures—especially of multi-constituent compounds—poses significant challenges for natural language processing. Existing tools like *compound-split* (Tuggener, 2016) perform well on compound head detection but are limited in handling long compounds and distinguishing compounds from non-compounds. This paper introduces NoCs (non-compound-stable splitter), a novel Python-based tool that extends the functionality of *compound-split* by incorporating recursive splitting, non-compound detection, and integration with state-of-the-art linguistic resources. NoCs employs a custom stack-and-buffer mechanism to traverse and decompose compounds robustly, even in cases involving multiple constituents. A large-scale evaluation using adapted GermaNet data shows that NoCs substantially outperforms *compound-split* in both non-compound identification and the recursive splitting of three- to five-constituent compounds, demonstrating its utility as a reliable resource for compound analysis in German.

1 Introduction

Compounding constitutes a core word formation process found in many languages of the world and is considered a highly efficient strategy for speakers to convey complex information by only employing little amount of linguistic signal. It is a process of combining pre-existing lexemes into new ones, thus creating informationally more compact structures as compared to their syntactically more embedded phrasal counterparts (Biber and Gray, 2010). For example, the compound *lexeme combination process* encodes the same information as *the process of the combination of lexemes*, but in a condensed linguistic unit, resulting in reduced signal transmission time and thus more efficient communication. This paper follows the definition of a compound of Jenkins et al. (2023) as ”single orthographic words which are composed of two or more constituents”. Lexicalized compounds such as *Tischbein* (Engl. ‘Table-leg’) will be included while opaque compounds like *Himbeere* (Engl. ‘Raspberry’) are not, where one of the constituents is considered an opaque morpheme (i.e. ”Modifiers whose meaning is not transparent any more without considering the etymology of the word” according to Henrich and Hinrichs (2011); in this case, *Him-* is opaque). In addition, words that can not be split further into independent constituents are considered non-compounds.

German, the language of the empirical focus of this paper, is a particularly well suited candidate for the investigation of compounding processes, as it offers an high amount of observable compounding. As one of the most frequently encountered word formation processes in German, compounding has to be considered not only as an efficient mechanism to transmit complex information but also as a highly generative process for ad hoc vocabulary. German compounds exhibit a theoretically almost unrestricted length and composition, vividly demonstrated by the well known compound in Example 1.

(1) *Rindfleischetikettierungsüberwachungs-aufgabenübertragungsgesetz*
(Engl. ‘Beef Labeling Monitoring Task Transfer Act’)

To investigate compounds computationally—especially long compounds like the one above—they often have to be split into their respective parts to examine the processes influencing their production and processing. Those

Proceedings of the Student Research Workshop associated with RANLP-2025, pages 44–53, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/issn.2603-2821.2025_006

parts are called modifier and head, where the modifier modifies the head. Both can themselves be compounds embedded in the first level of a compound. German compounds are structured head-last, i.e. the right element of a split being modified by the left element. According to Henrich and Hinrichs (2011), who published an extensive WordNet-style semantic network including semantically annotated compounds called GermaNet, the meaning of the entire compound highly depends on the meaning of its parts. Günther et al. (2020), who investigate the semantic transparency of compounds based on the relatedness of their constituents also show, that it is crucial to not only consider compounds as a whole unit, but to also analyze their respective parts and underlying structure as compound interpretation is more than the sum of the compound's parts. The difficulty in splitting a compound into those parts lies in the possibility of various possible splits especially in multi-constituent compounds according to Hätty et al. (2019), who evaluate several tools for the automatization of this task. This is due to a variable internal structure called branching structure in compounds with at least three constituents. Following the examples of Kösling and Plag (2009), a three-constituent compound can either be left-branching like in Example 2, where the first split is made between the second and third constituent, or right-branching like in Example 3, where the compound is first split between the first and second constituent.

(2) *seat belt law*
Interpretation: A law concerning seat belts.
Branching structure: [[NN]N]

(3) *team locker room*
Interpretation: A locker room for the team.
Branching structure: [N[NN]]

This impacts the internal structure of head and modifier, as in Example 2 the modifier itself is a compound and in Example 3 the head is a compound, each being able to be split into a head and modifier themselves. Annotating and curating compounds manually is thus a costly endeavor, both in terms of time and personnel, thereby motivating the development of automatic annotation methodologies. Multiple tools and resources have been developed and evaluated to date, addressing various aspects of automatic processing of compounds. The tool presented in this paper contributes to this line of research by offering a novel approach to automatic compound analysis by building on the *compound-split* tool by Tuggener (2016). It introduces a functionality to detect non-compounds as well as improving the handling of compounds longer than three constituents. Both tools—NoCs and *compound-split*—are then tested against each other on compounds of various lengths as well as on non-compounds. To promote open-access resources, NoCs will be made available at Gitlab under a CC BY 4.0 license.[1]

2 Related work

The production and processing of compounds have been of particular interest for psycho- and computational linguistics, as their processing is highly dependent on the audience group and their respective prior knowledge (Halliday, 1988/2004; Kendeou and van den Broek, 2007) as well as on linguistic (Meßmer et al., 2021) and communicative context (Gamboa et al., 2024, 2025). Psycholinguistic approaches to compounding behavior include the investigation of seriality of their processing (Andrews et al., 2004), differences between novelty and lexicalization in compounds (Hyönä et al., 2020), structural properties like the branching direction in multi-constituent compounds (Kösling and Plag, 2009) as well as the semantic relations between the separate constituents of compounds (Benjamin and Schmidtke, 2023). Ormerod et al. (2024) argue that Large Language Models (LLM) are able to distinguish between compounds sharing the same relation and compounds with different relations, incorporating both psycho- and computational linguistic approaches.

Especially when written or transcribed language is under investigation and has to be processed, curated and analyzed, pre-processing compounding and its underlying processes become a central task for natural language processing (NLP) and are therefore highly relevant to computational linguistics, as the computational processing of compound structures is not at all trivial. Several tasks of NLP highly depend on successfully processing compound structures such as the identification of the compound head for coreference resolution (Tuggener, 2016) or the analysis of the different dimensions of information status (Riester and Baumann, 2017).

[1]Gitlab: `https://gitlab.ruhr-uni-bochum.de/schaccmr/nocs.git`.

Various tools and resources tackling those tasks exist to date, each addressing specific aspects relevant to the individual task and in part building upon one another: see for example Hätty and Schulte im Walde (2018) for termhood prediction, Henrich and Hinrichs (2011) for the compound extension of GermaNet including constituent properties such as affixoid or opaque morpheme, Krotova et al. (2020) for the classification of compound idiomaticity, Tuggener (2016) for the identification of compound heads in the context of coreference resolution, Svoboda and Sevcikova (2024) for parent retrieval or Weller and Heid (2012) for compositional alignment of (compounded) terms in translation tasks. Simultaneously, those tools and resources do, however, exhibit various shortcomings, such as handling only compounds comprised of a maximum of three constituents or not being able to differentiate between compounds and non-compounds in the first place. The current paper will be addressing those aspects by presenting NoCs for German compounds based on the *compound-split* library; the Python implementation of the probabilistic n-gram based compound head detection algorithm presented in Tuggener (2016). The algorithm predicts the most probable split point within a word and returns the scores for various possible positions of a split in this word. It thus follows a machine learning approach trained on approximately one million German nouns from Wikipedia. It is a freely available NLP tool for the processing of German compounds, offering extensive documentation and achieving 95% accuracy for the detection of compound heads on the test data from GermaNet (Henrich and Hinrichs, 2011). It was selected for extension in this project due to its licensing terms, usability, and support for both nominal and adjectival compounds as it is a well-documented, license-free and straight-forward to use Python-library. Due to employing a machine learning approach it performs robustly without being computationally costly and data hungry like LLMs. It offers functions called *char_split*, which is splitting the head from its modifier and *maximal_split*, splitting the entire compound maximally not regarding the branching structure of a compound.

Although the tool performs robustly in compound head detection—its primary purpose—and achieves high accuracy values, its *maximal_split* function can only process compounds with no more than three constituents, while the *char_split* function lacks the ability to differentiate between compounds and non-compounds, resulting in erroneous segmentation like in Example 4.

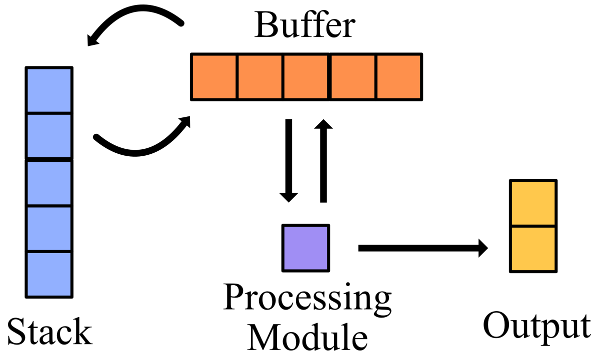


Figure 1: Architecture of the stack and buffer system implemented by NoCs.

(4) *Anma-Ssung instead of Anmassung*
(Engl. 'presumption')
Erroneous split of the non-compound Anmassung by the *char_split* function.

These constraints render it unsuitable for recursive decomposition of longer compounds into their respective heads and modifiers using only those two functions, as this would not reliably stop the splitting process at compound constituents, that can not be split any further.

3 Architecture of NoCs

To overcome these issues of the *Compound-split* tool, the newly developed NoCs tool introduced in this paper is addressing these issues building on and extending *compound-split* by leveraging the (correct and incorrect) feedback the base functions give to implement a custom stack and buffer system for recursive compound traversal and appropriate segmentation. NoCs is also implemented in Python (Van Rossum and Drake, 2009), an open-source programming language, to promote open-access resources and accessible research.

In addition to *compound-split* it also incorporates the state-of-the-art NLP *stanza* library (Qi et al., 2020) for part-of-speech and morphological feature look-ups during compound processing. It also utilizes a minimally adapted version of the *Free German Dictionary* (Schreiber, 2021) which is—like *compound-split*— based on Wikipedia crawls for dictionary look-ups and manually created lists of German prefixes and suffixes.

The tool's architecture is schematically presented in Figure 1. Details on the stack and buffer implementation are provided in Section 3.1 and 3.2, followed by a description of the test dataset in Section 4 and an evaluation of the results of both tools in Section 5.

3.1 Stack and buffer system

While the *char_split* function only detects the head of a compound and the *maximal_split* function can not properly handle compounds with more than three constituents, reliable recursive decomposition of multi-constituent compounds and non-compound identification are challenging when relying solely on the base functionalities of *compound-split*. Specifically, *char_split* exhibits unstable behavior on non-compounds, frequently producing erroneous outputs such as *Anm-Assung* from *Anmassung* ('presumption'). The position of the erroneous splits appears to be arbitrary, rendering it impossible to detect an erroneous split based on the split position alone. However, the length of the output lists returned when further splitting erroneous constituents displays predictable patterns that NoCs leverages for non-compound detection and recursive splitting.

Example 5 shows the output of *char_split* for the correct split of the compound *Testbeispiel* (Engl. 'test example'). The function returns a ranked list of candidate split positions with associated scores.

(5) *[(0.9571421456504741, 'Test', 'Beispiel'),*
(-0.7465882530347583, 'Testbei', 'Spiel'),
(-0.9921253246209264, 'Tes', 'Tbeispiel'),
(-1.5950942705473183, 'Testbeis', 'Piel'),
(-2.2783109404990403, 'Testb', 'Eispiel'),
(-2.2790028763183123, 'Testbe', 'Ispiel'),
(-2.660451197053407, 'Testbeisp', 'Iel')]
(Engl.: 'test example.')

The length and structure of these lists, however, is arbitrary and cannot reliably serve as indicators of compoundhood as the tool returns lists of variable lengths. Spurious splits for non-compounds may also result in singleton (see 6) or arbitrarily long lists (see Example 7 and 8).

(6) *[(-1.2889316935842348, 'Flü', 'Gel')]*
(Engl.: 'wing')

(7) *[(-1.3002238718039354, 'Bea', 'Mter'),*
(-1.5774219936893772, 'Beam', 'Ter')]
(Engl. 'administrative officer')

(8) *[(-1.137630662020906, 'Anma', 'Ssung'),*
(-1.3213892018267495, 'Anmaß', 'Ung'),
(-1.4081508515815087, 'Anm', 'Aßung')]
(Engl. 'presumption')

None of these splits are correct, but the tool still assigns a variable number of scores. In some cases however, the tool does seem to detect a non-compound and returns a score of 0 together with the unsplit token like in Example 9.

(9) *[(0, 'Käfig', 'Käfig')]]*
(Engl. 'cage')

NoCs evaluates these outputs recursively to identify failed segmentation, leveraging the output patterns of the base function and classify the input as a non-compound head.

It implements a stack-buffer architecture wherein partially segmented constituents are recursively pushed and popped from the stack and buffer through the processing module and onto the output. The individual elements are thus repeatedly examined for compoundhood and passed on through the stack-and-buffer system after successful segmentation and labeling. This architecture integrates the base function *char_split* of *compound-split* and evaluates the individual outputs through a series of tests for correct or erroneous splits.

The functionality of the basic stack-buffer architecture is demonstrated in the pseudo-code of Listing 1.

Each compound is pushed to the stack first, then recursively split using the *char_split* function and checked for compoundhood. The right elements of each split get pushed further to the buffer. The buffer-loop represents the inner processing containing the precessing module, which applies several tests to verify the compoundhood of an element. If the rightmost element is found it gets pushed to output and the next element on the buffer is processed until all elements are found and processed accordingly. Since initial *char_split* decisions and outputs are sensitive to compound length, NoCs avoids hardcoded assumptions and dynamically assigns left splits back to the stack while buffering right splits for continued processing in the processing module. This recursive traversal mechanism facilitates full decomposition, with backtracking when required (see Figure 1). To verify compoundhood, NoCs applies several criteria in this processing module to circumvent false segmentations by the base function.

NoCs_2	CS_2	CS_lemma_2	MS_2	MS_lemma_2	NoCs_3	MS_3	MS_lemma_3
2550	2351	3310	2049	2873	1530	473	907
0.51	0.47	**0.66**	0.41	**0.58**	**0.31**	0.1	0.18

Table 1: Absolute count and percentages of correct splits from the two and three constituent datasets. CS encodes the *char_split* function, MS the *maximal_split* function.

NoCs_4	MS_4	NoCs_5	MS_5	NoCs_noun	CS_noun	NoCs_adj	CS_adj
206	9	5	0	4099	935	537	187
0.21	0.01	0.21	0.0	**0.82**	0.19	**0.88**	0.31

Table 2: Absolute count and percentages of correct splits from the four and five constituent and non-compound datasets.

Listing 1: Stack-Buffer Architecture

```
procedure SplitCompounds(list)
 for compound in list do:
  push compound to stack
  while stack not empty do:
   split rightmost element in stack
       using char-split
   if split is correct:
    push right element of split to
        buffer
    push left element back to stack

    # bufferLoop:
    while buffer not empty do:
     split rightmost element in buffer
         using char-split
     if split is correct:
      push left element back to buffer

      # processingModule:
      while not done do:
       process rightmost element of
           split in buffer
      if done:
       push processed element to output

 prepare output format
 output
end procedure
```

3.2 Verification of the split

As valid splits may be associated with negative score values (see Example 10), polarity alone is insufficient as a determinative diagnostic feature.

(10) *[(-0.6059544658493871, 'Warmwasseraufbereitungsanlagen', 'Rohr')]* (Engl.: 'Pipe in the facility for the purification of warm water.')

As previously mentioned however, in the event of an erroneous split of a non-compound NoCs can leverage the patterns of incorrect return-values as it returns either an arbitrarily long output list with the invalid split decision or a list of only one score. In the first case, the next split of the (already invalid) split non-compound's right element will always return a list of one. For Example if the split of *Beamter* in example 7 returns two scores, the next split of its (incorrect) head *-Mter* would only return one score (see Example 11), indicating that no further splitting is possible.

(11) *[(0, 'Mter', 'Mter')]]*

NoCs identifies those single-score outputs or consecutive singleton results as indicative of failed splits and labels the original term as a head, as it can thus assume there are no more correct splits to follow. Additional verification is then conducted by the processing module. It applies prefix/suffix disambiguation to avoid prefix/suffix conflicts with identical sequence slices of words like *-Gel* (Engl.: 'gel') or suffixes like *-haft*, which has the identical surface form as the noun *Haft* (Engl.: 'detention') by using dictionary and suffix/prefix-list look-ups and morphological features as well as POS tagging parsed via *stanza*. For example, it avoids incorrect parses such as *Flü-Gel* (Engl.'wing') or *Anma-Ssung* (Engl. 'presumption') by cross-referencing gender and affixes. In the case of *Flü-Gel*, which is erroneously split into *Flü-* and *-Gel*, the head could theoretically function as the noun *Gel* (Engl.: 'gel'). To verify this, the tool first looks up the morphological features like grammatical gender of the input token, which is masculine for *Flü-Gel*. As the split would only be correct if the grammatical gender of the input token was neutral corresponding to the grammatical gender of its head *Gel* (Engl.: 'gel') it then decides that this can not be a correct split and returns the initial token as head. In the case of *Anma-Ssung* it detects the erroneous split by a look-up from the suffix list for the suffix *-Ung* in combination with a look-up from the dictionary

NoCs_2	CS_2	CS_lemma_2	MS_2	MS_lemma_2	NoCs_3	MS_3	MS_lemma_3
2751	2801	3579	2402	3092	1871	756	1113
0.55	0.56	**0.72**	0.48	**0.62**	**0.37**	0.15	0.22

Table 3: Absolute count and percentages of correct splits from the two and three constituent datasets in the lower-case test.

NoCs_4	MS_4	NoCs_5	MS_5	NoCs_noun	CS_noun	NoCs_adj	CS_adj
268	17	8	0	4099	935	537	187
0.28	0.018	0.33	0.0	**0.82**	0.19	**0.88**	0.31

Table 4: Absolute count and percentages of correct splits from the four and five constituent and non-compound datasets in the lower-case test.

list for its constituents to verify that this split is incorrect.

After deciding not to split any further it also runs a dictionary check on the current token, to check if the current non-compound is a valid word, before labeling it and pushing it to output. For example if it gets *(-1.3002238718039354, 'Bea', 'Mter')* and the next split of *-Mter* returns a score of 0 like in Example 11, it applies a dictionary look-up to verify the existence of the initial token *Beamter*. Thus, it collects all constituents and returns a Python dictionary structure like in Example 12, where it collects all constituents in the first index of the tuple, labels them with either 'head' or 'modifier' and also saves the respective modifier in the third index.

(12) *{'Testbeispiel': [('Test', 'modifier', '-'), ('Beispiel', 'head', 'Test')]}*
(Engl.: 'Test example.')

In a last step before returning the output, NoCs performs lemmatization and removes linking morphemes such as '-s-', resulting in linguistically plausible compound constituents.

4 Test-data and analysis

The current release of the GermaNet compound collection (Seminar für Sprachwissenschaft, University of Tübingen, 2024) was selected as a basis for the creation of test data. As NoCs primarily targets the detection of non-compounds and recursive split of multi-constituent compounds rather than compound head detection, the data set from GermaNet had to be slightly adapted to the task before applying to the two tools. The original dataset is comprised of three columns containing the compounds of lengths of two to six constituents and the respective modifiers and heads, see Example 13.

(13) *Abendbrot Abend Brot*
Abendbrottisch Abendbrot Tisch
(Engl.: 'Dinner' and 'Dinner table')

As this task needed not only the compound head in the case of multi-constituent compounds but the maximally split version, the respective modifiers were automatically searched and collected from the dataset until all heads were found, forming the individual constituents of the original compound. Thus all compounds were collected and categorized by number of constituents. From all the individual compound heads two datasets of nominal and adjective non-compounds were extracted by running automatic stanza parses on the non-compounds to determine nouns and adjectives. Where two possible constituents were listed, the nominal constituent was chosen, as they match the output of the tools more closely. Compounds containing numbers and hyphens were excluded.

From the two and three constituent compound datasets as well as the nominal non-compounds a random sample of five thousand was drawn. The four constituent compound dataset contains 965 compounds, the five constituent compound dataset contains 24 compounds and the adjective non-compound dataset contains 610 non-compounds (see Table 5). As there was only one single six constituent compound and it contained hyphens it was excluded from the evaluation.

Both tools processed all of the datasets. For *compound-split* the *maximal_split* function was used on the two to five constituent datasets. The *char_split* function was only tested on the two constituent dataset and the two non-compound datasets, as it only splits the head. The outputs of the *maximal_split* function on the two and three constituent datasets were lemmatized using stanza parsing at the constituent level, in addition to being retained

Dataset Type	Description	Sample Size
2-constituents	Random sample	5,000
3-constituents	Random sample	5,000
4-constituents	Full dataset	965
5-constituents	Full dataset	24
Nom. noc	Random sample	5,000
Adj. noc	Full dataset	610

Table 5: Dataset sample sizes by compound type.

in their original form, to evaluate whether lemmatization enhances alignment with the GermaNet gold standard, given that NoCs outputs are also lemmatized. To keep computation time minimal this was only conducted on the two and three constituent test-sets, as those offer the full five thousand samples as opposed to the four and five constituent test-sets and were therefore judged more representative for potential effects of the condition. In addition to these tests a second iteration of all the outputs was tested, where all constituents (test data and output) were set to lower case before comparing them, to account for possible spelling divergence when the token would technically be correct (see Table 6 for more details).

To calculate correct splits the outputs for each individual test-set were first compared in length and then tested for string-matches. If all constituents matched exactly the output was considered correct. The percentage of correct splits was then calculated. As all lists were tested separately, this was not considered a real classification task and thus the conventional evaluation metrics of precision and recall were not deemed appropriate for this evaluation.

Calculations and handling of data were carried out with the random and Pandas library (pandas development team, 2020).

5 Results

The comparative evaluation of both tools focused primarily on the handling of non-compounds and the accuracy in splitting multi-constituent compounds. With respect to compound segmentation, NoCs consistently outperforms the base *compound-split* functions across all unlemmatized datasets and under standard evaluation conditions, as shown in Tables 1 and 2, even though *char_split* surpasses *maximal_split* function in the two constituent dataset. A particularly prominent divergence in performance can be observed in the three to five constituent datasets, as the base function hardly captures any splits correctly (see NoCs-values in boldface). However, in the two-constituent dataset, the lemmatized condition significantly boosts performance, clearly surpassing NoCs. This indicates that lemmatization plays a substantial role in improving segmentation accuracy in the output of *compound-split* for simpler compounds.

As illustrated in Tables 2 and 4, the new NoCs demonstrates a clear advantage over *compound-split* in the domain of non-compound handling, as *compound-split* is only able to detect 19 percent of nominal non-compounds and 31 percent of adjective non-compounds. In this regard performance does not profit from the lower-case test, as NoCs still handles 82 percent of nominal and 88 percent of adjective non-compounds. Performance does, however, substantially benefit from the lower-case testing in the case of the two and three constituent datasets across all conditions as presented in table 3, accumulating to 72 percent correctly identified compounds in the lemmatized two constituent lower case condition.

6 Conclusion

Compounding representing an informationally compact and highly efficient linguistic phenomenon for encoding communicated information is particularly interesting within various linguistic fields and frameworks, including computational linguistics. In order to process those complex structures automatically highly specialized tools are necessary, especially in languages like German, where compounding is a highly productive process to (spontaneously) expand the language's vocabulary. Given the virtually unlimited number of potential constituents in German compounds, developing tools capable of reliably decomposing multi-constituent compounds into their component parts is of significant value for downstream applications. As the *compound-split* tool only offers a robust compound head detection and a considerable less robust maximal split approach, expanding the functionality by non-compound detection and a more stable multi-constituent compound split was the aim of the newly introduced NoCs tool.

As presented in section 5, the tool particularly excels in the domain of non-compound detection. While NoCs outperforms the base splitter on the

Dataset Type	NoCs	CharSplit	MaxSplit	CS_lemma	MS_lemma	regular	lower
2-constituents	x	x	x	x	x	x	x
3-constituents	x	-	x	-	x	x	x
4-constituents	x	-	x	-	-	x	x
5-constituents	x	-	x	-	-	x	x
Nom. noc	x	x	-	-	-	x	x
Adj. noc	x	x	-	-	-	x	x

Table 6: Testconditions across tools and datasets.

multi-constituent datasets, these improvements remain moderate, with approximately one-third of correctly split compounds in the three to five constituent datasets. A first preliminary and non-conclusive evaluation of the output data suggests that two key challenges persist: (1) as NoCs builds upon *compound-split*, it inherits certain limitations in the splitting of longer compounds, particularly due to the constraints of the original head detection mechanism, which performs more robustly in conditions with shorter compounds; and (2) longer compounds inherently increase the risk of incorrect splits, and the strict evaluation criterion of testing only perfect string matches on all constituents likely results in the reported performance as a conservative estimate and would probably benefit from a more fine-grained analysis including partial correctness.

Furthermore, both the lemmatization of output constituents and the bulk of the decision-making processes in NoCs rely heavily on stanza parses, increasing the risk to propagate early parsing errors through the entire system. Incorporating a more reliable lemmatizer might improve performance further. To improve overall performance on the actual compound splits, the tool might also greatly benefit from a more flexible handling of potential constituents within the split decision process. Though NoCs still leaves plenty of room for improvement on the split of multi-constituent compounds it provides a promising and practical solution for non-compound detection and contributes valuable functionality to the repertoire of NLP tools available for compound processing.

Limitations

Even though this new extension of the *compound-split* splitter addresses some of the shortcomings of the base splitter and expands the repertoire of it by several functionalities, it still exhibits various limitations and leaves room for improvements. First, it still is not able to confidently disambiguate two theoretically correct but possibly context-inappropriate splits due to its context-independent design. Integrating a language model with contextual understanding might contribute to solving this problem. Second, as NoCs relies on a dictionary for the decision on valid words, it also struggles with abbreviations contained in the dictionary, which might collide with non-abbreviations as sequence slices of words, causing the tool to falsely split. In this regard NoCs might benefit from a separate dictionary of abbreviations in the future.

Additionally, the aforementioned long compounds of five or more constituents still challenge the tool as well as the dependency on a lemmatizer, as this dependency introduces an increased likelihood of cascading errors through the process. A test employing edit distance metrics could allow insights on how many incorrect splits could be captured by a more precise lemmatizer. This online parsing during the processing of compounds also increases the runtime of the tool, rendering it more suitable for applications on smaller datasets. Furthermore, as not all compounds in the original GermaNet dataset were maximally split, this possibly caused some splits to be considered incorrect. A manually curated test set might alleviate this problem. In regard of the test data it also needs to be considered, that the GermaNet data is not 'new' in the context of the base tool, as it was used to test compound head detection accuracy for *compound-split*. To authentically simulate out-of-vocabulary testing a new test set would be desirable. To test on unseen ad hoc compounds, synthetic data could be generated and used for testing of new compounds of variable length in the future. For now, those limitations suggest a combination of both tools across the different conditions for the time being as an improved iteration for the task compared to the base functions of the *compound-split* splitter on its own.

Acknowledgments

I am grateful to the anonymous reviewers for their helpful comments and Stefanie Dipper and Ronja Laarmann-Quante from Ruhr-University Bochum for their valuable feedback. This research is funded by *Deutsche Forschungsgemeinschaft* (DFG, German Research Foundation) – Project-ID 232722074 – SFB 1102.

References

Sally Andrews, Brett Miller, and Keith Rayner and. 2004. Eye movements and morphological segmentation of compound words: There is a mouse in mousetrap. *European Journal of Cognitive Psychology*, 16(1-2):285–311.

Shaina Benjamin and Daniel Schmidtke. 2023. Conceptual combination during novel and existing compound word reading in context: A self-paced reading study. *Mem. Cognit.*, 51(5):1170–1197.

Douglas Biber and Bethany Gray. 2010. Challenging stereotypes about academic writing: Complexity, elaboration, explicitness. *Journal of English for Academic Purposes*, 9(1):2–20.

John Gamboa, Kristina Braun, Juhani Järvikivi, and Shanley E. M. Allen. 2025. The distributional properties of long nominal compounds in scientific articles: an investigation based on the uniform information density hypothesis. *Corpus Linguistics and Linguistic Theory*, 21(1):137–171.

John C. B. Gamboa, Leigh B. Fernandez, and Shanley E. M. Allen. 2024. Investigating the uniform information density hypothesis with complex nominal compounds. *Applied Psycholinguistics*, 45(2):322–367.

Fritz Günther, Marco Marelli, and Jens Bölte. 2020. Semantic transparency effects in german compounds: A large dataset and multiple-task investigation. *Behavior Research Methods*, 52(3):1208–1224.

M. A. K. Halliday. 1988/2004. On the language of physical science. In Jonathan J. Webster, editor, *The Collected Works of M. A. K. Halliday (Vol. 5)*, pages 140–158. Continuum, London and New York.

Anna Hätty and Sabine Schulte im Walde. 2018. Fine-grained termhood prediction for German compound terms using neural networks. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 62–73, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Verena Henrich and Erhard Hinrichs. 2011. Determining immediate constituents of compounds in GermaNet. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 420–426, Hissar, Bulgaria. Association for Computational Linguistics.

Jukka Hyönä, Alexander Pollatsek, Minna Koski, and Henri Olkoniemi. 2020. An eye-tracking study of reading long and short novel and lexicalized compound words. *Journal of Eye Movement Research*, 13(4).

Anna Hätty, Ulrich Heid, Anna Moskvina, Julia Bettinger, Michael Dorna, and Sabine Schulte im Walde. 2019. Akkubohrhammer vs. akkubohrhammer: Experiments towards the evaluation of compound splitting tools for general language and specific domains. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019): Long Papers*, pages 59–67, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.

Chris Jenkins, Filip Miletic, and Sabine Schulte im Walde. 2023. To split or not to split: Composing compounds in contextual vector spaces. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16131–16136, Singapore. Association for Computational Linguistics.

Panayiota Kendeou and Paul van den Broek. 2007. The effects of prior knowledge and text structure on comprehension processes during reading of scientific texts. *Memory & Cognition*, 35(7):1567–1577.

Irina Krotova, Sergey Aksenov, and Ekaterina Artemova. 2020. A joint approach to compound splitting and idiomatic compound detection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4410–4417, Marseille, France. European Language Resources Association.

Kristina Kösling and Ingo Plag. 2009. Does branching direction determine prominence assignment? an empirical investigation of triconstituent compounds in english. *Corpus Linguistics and Linguistic Theory*, 5(2):201–239.

Julia A. Meßmer, Regine Bader, and Axel Mecklinger. 2021. The more you know: Schema-congruency supports associative encoding of novel compound words. evidence from event-related potentials. *Brain and Cognition*, 155:105813.

Mark Ormerod, Jesús Martínez del Rincón, and Barry Devereux. 2024. How is a "kitchen chair" like a "farm horse"? exploring the representation of noun-noun compound semantics in transformer-based language models. *Computational Linguistics*, 50(1):49–81.

The pandas development team. 2020. pandas-dev/pandas: Pandas.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Arndt Riester and Stefan Baumann. 2017. *The RefLex Scheme — Annotation Guidelines*, volume 14 of *SinSpeC — Working Papers of the SFB 732 "Incremental Specification in Context"*. OPUS, Stuttgart.

Schreiber. 2021. Free German Dictionary. https://sourceforge.net/projects/germandict/files/. Accessed 04-07-2025.

Seminar für Sprachwissenschaft, University of Tübingen. 2024. GermaNet v19.0. https://uni-tuebingen.de/en/faculties/faculty-of-humanities/departments/modern-languages/department-of-linguistics/chairs/general-and-computational-linguistics/ressources/lexica/germanet/description/compounds/#c1081929. Accessed 04-07-2025.

Emil Svoboda and Magda Sevcikova. 2024. PaReNT (parent retrieval neural tool): A deep dive into word formation across languages. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12611–12621, Torino, Italia. ELRA and ICCL.

Don Tuggener. 2016. *Incremental Coreference Resolution for German*. Phd thesis, University of Zürich, Zürich, Switzerland.

Guido Van Rossum and Fred L. Drake. 2009. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.

Marion Weller and Ulrich Heid. 2012. Analyzing and aligning German compound nouns. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2395–2400, Istanbul, Turkey. European Language Resources Association (ELRA).

A Proposal for Evaluating the Linguistic Quality of Synthetic Spanish Corpora

Lucia Sevilla Requena
Universidad de Alicante / Alicante, Spain
lsr30@alu.ua.es

Abstract

Large language models (LLMs) rely heavily on high-quality training data, yet human-generated corpora face increasing scarcity due to legal and practical constraints. Synthetic data generated by LLMs is emerging as a scalable alternative; however, concerns remain about its linguistic quality and diversity. While previous research has identified potential degradation in English synthetic corpora, the effects in Spanish, a language with distinct grammatical characteristics, remain underexplored. This research proposal aims to conduct a systematic linguistic evaluation of synthetic Spanish corpora generated by state-of-the-art LLMs, comparing them with human-written texts. The study will analyse three key dimensions: lexical, syntactic, and semantic diversity, using established corpus linguistics metrics. Through this comparative framework, the proposal intends to identify potential linguistic simplifications and degradation patterns in synthetic Spanish data. Ultimately, the proposed outcome is expected to contribute valuable insights to support the creation of robust and reliable Natural Language Processing (NLP) models for Spanish.

1 Introduction

The development of Large Language Models (LLMs) has led to a paradigm shift in the field of Natural Language Processing (NLP), dramatically transforming the capabilities of current systems to understand and generate text (Touvron et al., 2023; van Noord et al., 2024). These models have achieved outstanding performance across a wide range of tasks, including machine translation, text generation, question answering, and semantic inference. However, their performance and robustness are critically dependent on the availability of high-quality, large-scale training data (Gandhi et al., 2024), yet obtaining such data has become a significant challenge (Villalobos et al., 2024; Chen et al., 2024).

The current training framework is heavily based on massive web-crawled corpora combined with curated datasets derived from books, scientific articles, and social media interactions (Penedo et al., 2023). Although this approach has been crucial in the evolution of LLMs, it faces significant structural limitations. On the one hand, scalability is constrained, as the amount of high-quality web data is finite and increasingly subject to legal, privacy, and copyright restrictions (Kurakin et al., 2024; Amin et al., 2025). On the other hand, much of the available crawled data suffers from quality issues, including noise, spam, misinformation, redundancy, toxic content, and increasingly low-quality machine-generated text (Trinh and Le, 2019; Kreutzer et al., 2022).

In response to growing data limitations, synthetic data generated by LLMs has emerged as a scalable and increasingly viable alternative (Long et al., 2024). Recent research demonstrates that current models can produce syntactically correct, semantically coherent, and stylistically diverse texts that are, in some cases, nearly indistinguishable from human-written content (Hartvigsen et al., 2022; Gao et al., 2023; Liu et al., 2024).

However, this approach introduces significant risks. A key concern is 'model collapse', which occurs when models are repeatedly trained on data generated by other models rather than on human-produced language (Gerstgrasser et al., 2024). This leads to a gradual degradation of linguistic quality (Shumailov et al., 2024), including loss of syntactic and semantic diversity, oversimplification of structures, increased redundancy, and a higher incidence of hallucinations, which are factually incorrect or incoherent outputs (Long et al., 2024). Over time, this severely undermines the model's ability to replicate the richness and complexity of natural

Proceedings of the Student Research Workshop associated with RANLP-2025, pages 54–61, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/issn.2603-2821.2025_007

language (Bender et al., 2021; Penedo et al., 2023).

Despite recent studies exploring the benefits and risks of synthetic data (Liu et al., 2024; Gilardi et al., 2023), there is still a lack of methodological frameworks that rigorously assess the linguistic quality of synthetic data compared to real human data. This gap raises important concerns about whether synthetic data can truly support effective model training without introducing problems. Therefore, there is an urgent need for more rigorous and linguistic evaluation methods to assess whether synthetic corpora adequately reflect the qualities of human-produced text and can ensure the long-term reliability of NLP systems.

The present proposal seeks to address this gap by designing and implementing a systematic linguistic evaluation of synthetic Spanish data generated by state-of-the-art LLMs, focusing on three dimensions: lexical, syntactic and semantic diversity. While existing research has predominantly focused on English, the linguistic effects of synthetic data generation in other languages remain largely underexplored.

In this context, the proposed study takes a new perspective by examining whether the patterns of linguistic degradation observed in English synthetic data also manifest in Spanish, a language with fundamentally different grammatical properties. To this end, the study will develop a comparative framework, grounded in quantitative corpus-linguistic metrics, to systematically evaluate and contrast synthetic Spanish corpora with authentic human-written corpora of comparable size and genre. It is worth noting that this framework remains to be operationalised.

This comparative analysis aims to reveal whether risks such as linguistic simplification and loss of structural and semantic richness are universal phenomena or language-specific issues. This methodological approach aims to uncover whether said degradation previously observed in English also occurs in Spanish.

2 Background and Related Work

The increasing reliance on synthetic data used to overcome the limited availability of high-quality human-produced corpora has attracted growing attention in recent years. A substantial body of research has emerged examining the potential and limitations of synthetic datasets in the training of large language models (LLMs), particularly related to their linguistic properties and their implications for NLP systems. Hence, the present section reviews relevant literature on the risks associated with synthetic data, with particular emphasis on the loss of linguistic diversity in machine-generated texts. Situating this study within the broader context of these works provides the theoretical and empirical foundation for the proposed linguistic evaluation of synthetic Spanish corpora.

2.1 Risks in Synthetic Data

To commence, although synthetic data has been proposed as a scalable solution to the aforementioned problem of scarcity, ongoing research has identified several risks that can seriously affect the quality of models trained on this type of data (Marwala et al., 2023; Hao et al., 2024). These risks are diverse and impact not only the properties of the corpus itself but also the ability of models to perform well.

One of the most relevant issues is data bias, which occurs when synthetic data does not accurately reproduce the real characteristics of authentic data (Hao et al., 2024). This can lead models to learn inaccurate or unrealistic representations, reducing their reliability.

Closely related to this is the phenomenon of oversmoothing, where synthetic data tends to remove natural variation and rare patterns. As a result, the corpus becomes too homogeneous and simplified, lacking the complexity needed to train robust models (Hao et al., 2024). Such a loss of complexity contributes to the degradation of linguistic diversity in synthetic content.

Another common risk is incomplete or inaccurate information, as synthetic data does not always capture the full diversity of linguistic phenomena present in real texts. This is partly due to the limitations of generative models, which often suppress noise or contain algorithmic flaws (Marwala et al., 2023; Hao et al., 2024).

These risks are not just technical problems, but fundamental challenges that threaten the sustainability and reliability of natural language processing systems. As synthetic data becomes more widespread, understanding how it affects quality is key to designing strategies that can mitigate its negative impact.

2.2 Language Diversity Loss in Synthetic Data

Several recent studies have shown a growing interest in analysing how the use of LLMs affects linguistic diversity, both in machine-generated text and in text produced by humans assisted by these models (Guo et al., 2024a). A common concern in this line of research is that, although LLMs have demonstrated remarkable capabilities in generating fluent and grammatically correct text, their use may lead to processes of linguistic homogenisation that reduce the richness and diversity of language. In particular, synthetic corpora often lack spelling mistakes and tend to underrepresent non-standard dialects, which further limits their applicability in real-world contexts.

Liang et al. (2024) identified a significant shift in lexical frequencies in academic writing, with an increase in the use of LLM-preferred words starting around five months after the release of ChatGPT in 2022. Similarly, Luo et al. (2024) demonstrated that machine translations exhibit lower morphosyntactic diversity and greater convergence compared to human translations. The authors attributed this outcome, in part, to the use of beam search, which biases outputs toward more frequent and less diverse patterns.

Finally, Padmakumar and He (2024) found that writing assisted by InstructGPT also reduces textual diversity compared to writing with GPT-3 or without model assistance. This effect is primarily driven by the model's output rather than by user behaviour. The authors warned that while reinforcement learning with human feedback (RLHF) improves the model's ability to follow instructions, it may also constrain personal expression. This highlights the need for user-centred evaluations and the development of more customisable models that preserve linguistic diversity.

In conclusion, systematic and language-specific evaluations of synthetic corpora are still scarce for languages such as Spanish. This study addresses said necessity through a comparative analysis of human and synthetic Spanish corpora across lexical, syntactic, and semantic levels.

3 Main Hypothesis and Objectives

The present research proposal is based on the hypothesis that synthetic data generated by large language models (LLMs) in Spanish may exhibit lower linguistic richness and diversity compared to human-produced data. If synthetic data is continuously used for model training, it could lead to a degradation of the linguistic quality of LLMs. Specifically, artificially generated texts are expected to show a more limited and repetitive vocabulary, simpler and less varied syntactic structures, and lower semantic coherence, resulting in discourse that is less connected, redundant, or even inconsistent (Guo et al., 2024b). Such linguistic deficiencies could negatively impact the ability of models trained with synthetic data to understand and produce natural language in real-world contexts, thereby compromising their performance on complex linguistic tasks.

From this perspective, the main objective of this research proposal is to perform a detailed linguistic evaluation of the synthetic Spanish corpora generated by LLMs. The evaluation will focus on three key dimensions: lexical, syntactic, and semantic. The purpose is to assess how the synthetic data reflects the natural variability and structural richness of the Spanish language. This will be done through a comparison between synthetic texts and human Spanish corpora of similar size and genre.

To achieve this general goal, the study proposes the following specific objectives:

- **O1:** To assess lexical diversity by applying established corpus linguistics metrics such as type-token ratio (TTR), lexical density, and vocabulary growth measures. These metrics will help determine whether synthetic texts maintain a wide and varied vocabulary comparable to that found in natural Spanish.

- **O2:** To examine syntactic complexity by analysing the presence and frequency of complex sentence constructions, including subordinate clauses, coordination, and sentence embedding. This will help determine whether synthetic data reproduces the grammatical sophistication of human language use.

- **O3:** To evaluate semantic diversity by measuring how much the synthetic texts cover different meanings and topics. This will be done using sentence embeddings to calculate semantic dispersion and topic modelling to assess the range and balance of themes. These metrics will assess if synthetic data reflects the richness and variability of natural Spanish.

- **O4:** To conduct a human evaluation aimed at identifying specific patterns of linguistic degradation in synthetic data through systematic comparison with natural corpora. Understanding these patterns will help guide the creation of higher-quality synthetic datasets that better support the training of reliable and robust Spanish language models.

- **O5:**To compare the impact of synthetic data on Spanish with previously reported effects in English, thereby distinguishing universal patterns of linguistic simplification from phenomena specific to Spanish.

Through these objectives, the study seeks to provide a clearer picture of the current limitations of synthetic data in Spanish and contribute to the construction of higher-quality data.

4 Proposed Methodology

This study proposes a methodology for the evaluation of the linguistic quality of synthetic data generated by LLMs in Spanish, structured in different stages. The approach is grounded in the framework developed by Guo et al. (2024b) in "The Curious Decline of Linguistic Diversity: Training Language Models on Synthetic Text", who demonstrated that synthetic data, while effective for improving task performance, systematically exhibit a decline across three key dimensions: lexical, syntactic and semantic diversity when compared to human-written texts. Their findings underscore the importance of incorporating fine-grained linguistic analysis into the evaluation of synthetic corpora, especially when these corpora are intended for use in training language models.

4.1 Data Gathering and Generation

The first stage of this proposal involves the careful selection and preparation of datasets. To carry out the study, two primary datasets will be established: (1) a natural corpus consisting of texts authored by humans, and (2) a synthetic corpus generated artificially by LLMs. The natural corpus will be an existing and compiled dataset, ensuring that the texts are available in open formats and preprocessed to guarantee comparability.

For the synthetic corpus, publicly available synthetic datasets will be collected, and additional texts will be generated using pretrained models like GPT-4, LLaMA 2, or Mistral, among others. Efforts will be made to produce a volume of text comparable to that of the natural corpus to ensure statistical validity. Generation prompts will be carefully crafted to yield texts with styles and thematic content closely matching the human-written corpus.

Finally, both corpora will undergo linguistic normalisation procedures to ensure that all subsequent comparisons are performed on consistent, noise-free data.

4.2 Linguistic Analysis of Corpora

In the second stage of the methodology, a thorough analysis will be carried out to assess the linguistic diversity present in the previously collected human and synthetic corpora. Following Tevet and Berant (2021), diversity can be understood in two main ways: content diversity, answering "What to say?", and form diversity, answering "How to say it?". In the words of Guo et al. (2024a), "lexical diversity and syntactic diversity are considered sub-aspects of form diversity, while semantic diversity reflects content diversity".

Although other sub-aspects of linguistic diversity exist, such as style or register, these tend to be more ambiguous, harder to measure, and often overlap with the three main dimensions. For these reasons, this study will focus specifically on the three clearly defined and quantifiable dimensions mentioned above (Guo et al., 2024b), which offer a solid foundation for comparative analysis.

To fulfil the goal of evaluating and comparing synthetic and human corpora, the analysis is organised around the following dimensions:

4.2.1 Lexical Diversity

Lexical diversity generally refers to the proportion of unique word types within a standardised text sample, such as the total number of tokens (Zheng, 2025). Laufer and Nation (1995) defined measures of lexical richness as attempts to "quantify the degree to which a writer is using a varied and large vocabulary." Consequently, lexical diversity is widely recognised as one of the most direct indicators of lexical richness (Vermeer, 2004).

Lexical diversity metrics quantify the range of vocabulary used in a text, which can reflect both the richness of a language model and its ability to generate varied language (Zheng, 2025). Following the hypothesis presented by Guo et al. (2024a), models trained on synthetic data tend to exhibit a more limited lexical repertoire, often resulting in repetitive and predictable language generation.

In the context of Spanish, the evaluation of lexical diversity presents additional challenges due to the rich inflectional morphology of the language.

In addition, variability caused by verb conjugations, along with gender and number agreements, can artificially inflate surface-level type counts. As a result, accurately assessing lexical variation becomes more complex.

To assess these challenges in Spanish, this study will adopt a set of lexical diversity metrics from corpus linguistics to ensure a comprehensive evaluation:

- **Type-Token Ratio (TTR)** (Johnson, 1944): The ratio between the number of lexical types (unique words) and the total number of tokens in a text. Due to its well-known sensitivity to text length, this metric is applied to texts truncated to a fixed length, following the approach proposed by Guo et al. (2024a).

- **Distinct-n** (Li et al., 2016): Computes the proportion of unique n-grams over the total number of n-grams. This study uses $n = 1$ (equivalent to TTR), $n = 2$, and $n = 3$, as this indicator is particularly informative to evaluate diversity in longer lexical sequences.

- **Self-BLEU** (Zhu et al., 2018): A metric originally developed for generative models that measures the similarity between generated sentences within the same data set. Lower Self-BLEU indicates higher diversity.

These metrics collectively provide a robust view of lexical diversity, accounting for both the superficial variety of word forms and the deeper variability of lexical patterns.

4.2.2 Syntactic Diversity

Syntactic diversity refers to the variety and complexity of sentence structures present in a text or corpus. It shows how flexibly different grammatical parts are used, such as phrases, clauses, and sentence types (Guo et al., 2024b).

According to Bastiaanse and Edwards (1998), higher syntactic diversity makes the text more expressive and adds subtle meaning, affecting its style and tone. Texts with high syntactic diversity have many different sentence forms, while texts with low diversity tend to use repetitive or simple sentences. Additionally, exposure to different syntactic structures is essential for language models to develop a deeper and more complex understanding of language (Aggarwal et al., 2022).

Despite its importance, syntactic diversity has been a relatively underexplored aspect in linguistic analyses (Guo et al., 2024b). This phenomenon is especially significant in Spanish, a language characterised by flexible word order, frequent subject ellipsis, and abundant use of subordinate clauses.

To evaluate this diversity, the present study will employ traditional syntactic complexity metrics commonly used in linguistic research. These metrics are as follows:

- **Syntactic Complexity Index (SCI)** (Lu, 2009): which integrates characteristics such as the average depth of dependency trees, the proportion of subordinate clauses and the mean sentence length.

- **Subordination Ratio** (Hunt, 1965): defined as the proportion of subordinate clauses relative to the total number of clauses, is a widely used metric in the research of syntactic complexity in Spanish.

Together, these metrics capture both the structural diversity and the richness in the syntactic configurations generated by the models.

4.2.3 Semantic Diversity

Semantic diversity refers to the range and variability of meanings, concepts, and topics expressed within a text or across a collection of texts. To capture this dimension, the present study will adopt a dual approach that combines embedding-based and network-based methods, which together provide a robust assessment of semantic variation.

On the one hand, semantic dispersion (Div_sem) is calculated by representing each sentence as a dense vector that captures its meaning within a multilingual semantic space, using SBERT (Reimers and Gurevych, 2019). Then, the average cosine distance between all pairs of sentence vectors is measured to estimate how far the document spreads across semantic space. A higher dispersion value reflects greater variety in the concepts covered.

On the other hand, topic diversity is measured using BERTopic (Grootendorst, 2022), which groups together semantically similar sentence vectors to identify underlying topics in the text. Diversity is then quantified by (a) counting the number of distinct topics found and (b) calculating topic entropy, which reflects how rich and evenly distributed the thematic content is across the document.

Lastly, this combined approach enables a detailed comparison of semantic diversity between human-authored and synthetic texts.

5 Expected results

Based on the proposed methodology, preliminary assumptions suggest that synthetic corpora in Spanish may display lower linguistic diversity compared to human-authored texts. For instance, synthetic texts are expected to exhibit reduced lexical richness, with comparatively lower type-token ratios (TTR), smaller distinct-n values, and higher Self-BLEU scores, indicating a tendency toward repetitive and homogeneous vocabulary. At the syntactic level, a decrease in syntactic complexity is anticipated, reflected in shallower dependency trees, shorter average sentence lengths, and lower subordination ratios, suggesting a preference for simpler and more uniform sentence structures. Finally, in the semantic dimension, synthetic corpora might cover a narrower range of topics and exhibit lower semantic dispersion, which would signal limited conceptual variability.

In conclusion, it is hypothesised that these results may align with previous findings in English. Moreover, given Spanish's greater morphological complexity and comparatively lower online representation, the negative impact of synthetic data is expected to be more pronounced. Nevertheless, these expectations remain tentative and will only be confirmed once the proposed evaluation framework is applied.

References

Arshiya Aggarwal, Jiao Sun, and Nanyun Peng. 2022. Towards robust NLG bias evaluation with syntactically-diverse prompts. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6022–6032, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kareem Amin, Sara Babakniya, Alex Bie, Weiwei Kong, Umar Syed, and Sergei Vassilvitskii. 2025. Escaping collapse: The strength of weak data for large language model training.

Y.R.M. Bastiaanse and S. Edwards. 1998. Diversity in the lexical and syntactic abilities of fluent aphasic speakers. *Aphasiology*, 12(2):99 – 117.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I. Abdin. 2024. On the diversity of synthetic data and its impact on training large language models.

Saumya Gandhi, Ritu Gala, Vijay Viswanathan, Tongshuang Wu, and Graham Neubig. 2024. Better synthetic data by retrieving and transforming existing datasets.

Jiahui Gao, Renjie Pi, Yong Lin, Hang Xu, Jiacheng Ye, Zhiyong Wu, Weizhong Zhang, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2023. Self-guided noise-free data generation for efficient zero-shot learning.

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, Daniel A. Roberts, Diyi Yang, David L. Donoho, and Sanmi Koyejo. 2024. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30).

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure.

Yanzhu Guo, Guokan Shang, and Chloé Clavel. 2024a. Benchmarking linguistic diversity of large language models.

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. 2024b. The curious decline of linguistic diversity: Training language models on synthetic text.

Shuang Hao, Wenfeng Han, Tao Jiang, Yiping Li, Haonan Wu, Chunlin Zhong, Zhangjun Zhou, and He Tang. 2024. Synthetic data in ai: Challenges, applications, and ethical implications.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326, Dublin, Ireland. Association for Computational Linguistics.

Kellogg W Hunt. 1965. *Grammatical structures written at three grade levels*. 8. National Council of Teachers of English.

Wendell Johnson. 1944. Studies in language behavior: A program of research. *Psychological Monographs*, 56(2):1–15.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol

Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Alexey Kurakin, Natalia Ponomareva, Umar Syed, Liam MacDermed, and Andreas Terzis. 2024. Harnessing large-language models to generate private synthetic text.

Batia Laufer and Paul Nation. 1995. Vocabulary size and use: Lexical richness in l2 written production. *Applied linguistics*, 16(3):307–322.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Weixin Liang, Yaohui Zhang, Zhengxuan Wu, Haley Lepp, Wenlong Ji, Xuandong Zhao, Hancheng Cao, Sheng Liu, Siyu He, Zhi Huang, Diyi Yang, Christopher Potts, Christopher D Manning, and James Y. Zou. 2024. Mapping the increasing use of LLMs in scientific papers. In *First Conference on Language Modeling*.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, and Andrew M. Dai. 2024. Best practices and lessons learned on synthetic data.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11065–11082, Bangkok, Thailand. Association for Computational Linguistics.

Xiaofei Lu. 2009. Automatic measurement of syntactic complexity in child language acquisition. *International Journal of Corpus Linguistics*, 14(1):3–28.

Jiaming Luo, Colin Cherry, and George Foster. 2024. To diverge or not to diverge: A morphosyntactic perspective on machine translation vs human translation.

Tshilidzi Marwala, Eleonore Fournier-Tombs, and Serge Stinckwich. 2023. The use of synthetic data to train ai models: Opportunities and risks for sustainable development.

Rik van Noord, Taja Kuzman, Peter Rupnik, Nikola Ljubešić, Miquel Esplà-Gomis, Gema Ramírez-Sánchez, and Antonio Toral. 2024. Do language models care about text quality? evaluating web-crawled corpora across 11 languages.

Vishakh Padmakumar and He He. 2024. Does writing with language models reduce content diversity?

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross J. Anderson, and Yarin Gal. 2024. Ai models collapse when trained on recursively generated data. *Nat.*, 631(8022):755–759.

Guy Tevet and Jonathan Berant. 2021. Evaluating the evaluation of diversity in natural language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 326–346, Online. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Trieu H. Trinh and Quoc V. Le. 2019. A simple method for commonsense reasoning.

A. Vermeer. 2004. *The relation between lexical richness and vocabulary size in Dutch L1 and L2 children*, number 10 in Language Learning and Language

Teaching, pages 173–189. John Benjamins Publishing Company, Netherlands. Pagination: 17.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. Will we run out of data? limits of llm scaling based on human-generated data.

Wanwan Zheng. 2025. Lexical richness viewed through lexical diversity, density, and sophistication. *Digital Scholarship in the Humanities*, 40(2):692–708.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models.

Personalizing Chatbot Communication with Associative Memory

Kirill Soloshenko, Alexandra Shatalina, Elisabeth Kornilova,
Marina Sevostyanova, Konstantin Zaytsev
HSE University, Russia
kasoloshenko@edu.hse.ru, afshatalina@edu.hse.ru,
elankornilova@edu.hse.ru, mksevostianova@edu.hse.ru,
kzaytsev@hse.ru

Abstract

Despite the significant progress made by large language models (LLMs) over the past few years, they are still limited in context and struggle to retain user-specific information over extended interactions, which significantly affects their quality. While current research is focused on expanding the contextual window, our approach is aimed at effectively expanding the context through integrating a database of associative memory into the natural language processing (NLP) pipeline. In order to improve long-term memory and personalization we have utilized methods close to Retrieval-Augmented Generation (RAG).

We implement a multi-agent consecutive pipeline in order to improve the quality of personalization as measured in accuracy, which contains: (1) a cold-start agent to handle sparse initial interaction; (2) a fact extraction agent to detect and extract user inputs from the dialogue; (3) an associative memory agent to store and retrieve contextual data; and (4) a generation agent.

Evaluation results demonstrate promising performance: our pipeline increases the accuracy of the base Gemma3 model by 41%, from 16% to 57%. Hence, with our approach, we demonstrate that personalized chatbots can bypass LLM memory limitations while increasing information reliability under the conditions of limited context and memory.

1 Introduction

Although large language models (LLMs) have spurred considerable progress in natural language processing (NLP), inherent limitations still exist. A well-documented constraint is the difficulty LLMs encounter when generalizing across extended contextual lengths. This presents challenges in applications such as personalized chatbots, where maintaining consistent user-specific information over a long period of different sparse interactions is crucial, and LLMs frequently exhibit a tendency to "forget" previously established details. While existing research, for example, (Jin et al., 2024) and (Ding et al., 2024), explores methods for expanding the context window, and some models are pre-trained with large context windows (Yang et al., 2025), our approach contrastively focuses on achieving extended context through the integration of an associative memory database within the NLP pipeline.

The hypothesis is that, while the immediate inclusion of Retrieval-Augmented Generation (RAG) user-related data may introduce short-term complexity for the LLM, this strategy will ultimately enhance long-term user-specific memory and coherence within the personalized chatbot interaction.

Our pipeline includes four agents that work with the associative memory database to improve the personalization quality. The agents deal with the following tasks: fact extraction, associative memory, generation and the “cold start” issue resolution.

2 Related Works

Our research focuses on personalized communication with a chatbot, the key to which we consider the associative memory.

Chen et al. (2024) in their work, provide an overview of different approaches and datasets in personalized dialogue generation. To start with, the datasets used for training can vary, and while some contain descriptive sentences (Zhang et al., 2018),

Proceedings of the Student Research Workshop associated with RANLP-2025, pages 62–69, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/issn.2603-2821.2025_008

others have simple key-value attributes like age, gender, location, etc. (Qian et al., 2018).

The article by Zhang et al. (2024) describes common issues that can be encountered during chatbot development. It proposes a more theoretical overview of some of the methods we have utilized during development. The metrics described are similar to those we have used for evaluation of the performance of our pipeline and agents: accuracy, precision, recall, F1-score and top-K.

A relevant issue that is also described within the article is the cold start problem. It is mainly encountered in recommendation systems and can be divided into "user cold start" and "item cold start" (Yuan and Hernandez, 2023). When the system encounters a new user or item it has not seen before and therefore has no information about them, their connection to each other, it still has to offer the user accurate recommendations. This problem is also encountered during chatbot development where, like in a real conversation, there must be topics that are both interesting to the user and relevant to the situation, even when we have little to no information about them in the database.

Zhang et al. (2024) highlight that many studies (Salemi at al., 2023), (Rajput et al., 2023), (Xi et al., 2023) choose to remove users with minimal interaction history during the preprocessing stages. This exclusion potentially undermines the robustness of the systems by disregarding the subtleties and potential insights offered by these underrepresented user interactions. Therefore, by resolving the cold start issue we do not encounter such drawbacks and improve the performance of our pipeline.

There are studies that utilize relevant facts for the personalized response generation like DuLeMon (Xu et al., 2022), which uses a classifier to determine whether a clause in an utterance contains personal information. In contrast, our associative memory implementation relies on the facts contained in the database in the form of triplets: subject, predicate, object, embedded using an arbitrary encoder and ranked by cosine similarity when each new user query is being received.

When the personas were not explicitly given in DuLeMon, they were extracted from dialogue histories. The seminal paper by (Zhang et al., 2018) emphasized that the agent specifically targets conversational data where personal attributes and relationships are often implied through complex linguistic patterns. Wu et al. (2020) and Wang et al. (2022) both underlined the value of implicit user modeling based on linguistic cues, strengthening our rationale for integrating linguistic tools like syntactic trees and coreference resolution. The cited works demonstrated that effective persona extraction requires handling three critical challenges: (1) resolving referential ambiguity, (2) capturing implicit relationships, and (3) maintaining consistency across multi-turn interactions - all of which directly informed the agent's architecture.

The generation agent is the most important part of any chatbot as it is crucial to efficiently generate responses to user's queries. There are many approaches to response generation with LLM. For example, it is possible to finetune the LLM with PEFT as Zhang et al. (2025) do in their work "Personalized LLM Response Generation with Parameterized User Memory Injection". They propose a parameterized Memory-injected approach and combine it with Bayesian Optimization searching strategy and LoRA in order to achieve LLM Personalization. We focused on prompt engineering as we find it one of the most effective ways to generate personalized responses to user's messages. A prompt is an input to a generative model, which is used to guide its output. Prompts make models more flexible and convenient to interact with. There are a number of papers where prompt-engineering approaches are described, for example, in the work of Sander Schulhoff et al. (2024).

The datasets we used for training models and testing agents' performance were Synthetic Persona Chat [1] (Jandaghi et al., 2024) and MultiSession Chat [2] (Xu et al., 2022) as they provided the most accurate data used in personalized dialogues.

[1] https://huggingface.co/datasets/google/Synthetic-Persona-Chat

[2] https://huggingface.co/datasets/nayohan/multi_session_chat

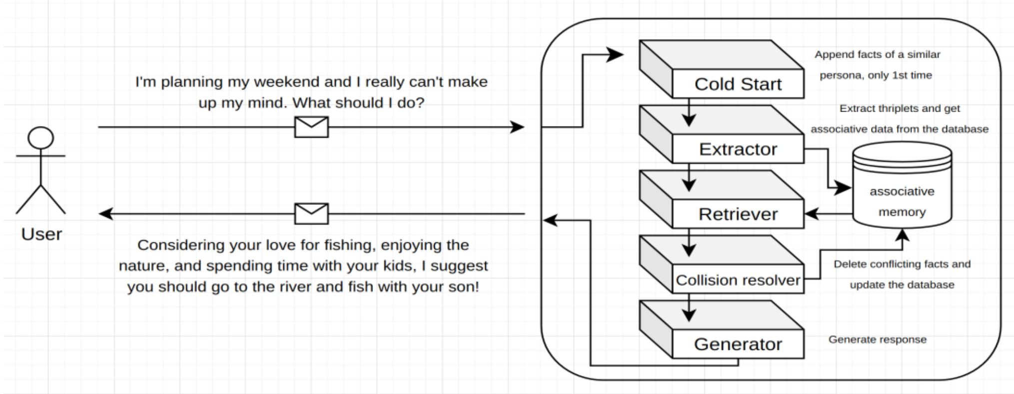


Figure 1: Chatbot pipeline schema, illustrating the key stages from user input processing to response generation: optional cold start, triplet extraction, fact retrieving and collision resolution, response generation.

3 Approach

The scheme of the chatbot pipeline is shown in Figure 1. When a new user starts chatting with our bot, basic facts about them, for instance age, gender and personal interests, go into the cold start agent to get potential dialogue options based on facts about other similar users which are stored in the database in the form of triplets. These topics then go to the associative memory agent for collision resolution. After that the triplets enter our generation agent where they are mixed in with the user's query to produce a response.

If there is a history of communication with the chatbot and the associative memory database contains user information, the pipeline slightly differs. The query first goes through fact extraction, where important information about the user is retrieved from their message in the form of triplets. The associative memory agent then searches the database for information relevant to the query and resolves collisions of triplets extracted in the previous step with the existing data about the user. The filtered facts then get mixed in the user's query.

3.1 The Cold Start Agent

The "cold start" agent exists within the pipeline to deal with new users that have little to no information about them. It is important for conversations with our chatbot to be active and interesting even with unknown users, which is the goal that this agent pursues (Table 1).

Our solution to the "cold start" problem is based on a retrained Sentence Transformers (Reimers and Gurevych, 2019) model to encode persona embeddings and find similar personas based on their cosine similarity. The training dataset was derived from Synthetic Persona Chat. First, embeddings of unique facts were encoded with the encoder (BAAI/bge-small-en) [3] and compared using cosine similarity, connecting the personas they. The model was fine-tuned on a new dataset, which was made from positive and negative pairs of personas obtained previously.

User's persona	Similar persona
I love horses	I love animals, I love dancing, I am a vegan, I love country music, I have a farm with pigs, horses and hens, I would like to go to school to become a veterinarian, I am currently on a diet, I love going to the gym, I have three pets, I love animals and I want to help them

Table 1: Example of cold start agent performance.

3.2 The Fact Extraction Agent

The "fact extraction" agent is designed to identify and structure personal information from dialogue in the form of triplets (subject, predicate, object). The metadata fields such as timestamps are stored alongside the triplets in the database and used, for

[3] https://huggingface.co/BAAI/bge-small-en

instance, during collision resolution. This agent aims to build dynamic user profiles and adapt responses based on user-specific information. We extract facts from dialogues using a rule-based method built on top of a syntactic dependency parser (spaCy) (Honnibal et al., 2020), enhanced with coreference resolution via en_core_web_trf[4] transformer-based model with the coreferee[5] plugin.

The extraction process identifies subject-predicate-object triplets by analyzing the syntactic structure of each utterance including support for complex grammatical constructions. The triplets are passed to the next agent and stored in its database as JSON structures.

Unlike end-to-end neural approaches that treat fact extraction as a sequence-labeling task, our approach explicitly models the syntactic and referential hierarchies inherent in conversational data. The core idea is to traverse the syntactic structure of each sentence to detect subject–verb–object patterns and their variants, including passive constructions, gerunds, embedded clauses, and comparative expressions. To enhance the agent's understanding of discourse-level references, we incorporated a tool for coreference resolution. This was essential for accurately interpreting anaphoric expressions such as pronouns, which frequently occur in dialogues.

Coreference resolution is applied as a preprocessing step. Utilizing coreference resolution we rewrite dialogue text by substituting pronouns with their most salient antecedents based on the coreference chain. This preprocessing improves the accuracy of later syntactic parsing by ensuring that each clause contains fully explicit noun phrases, thereby reducing ambiguity in triplet generation.

The syntactic parsing module analyzes each sentence by identifying the ROOT verb and its dependents to form canonical subject–predicate–object triplets. While basic SVO structures are straightforward to extract, natural language often involves more complex grammatical patterns that obscure the core meaning. To ensure accurate fact extraction, we focused on a targeted set of syntactic constructions that are both frequent in dialogue and crucial for preserving semantic relationships. These include passive voice, dative constructions, control and open clausal complements, nested complement clauses, comparatives, full noun phrase reconstruction, and negation propagation.

To illustrate how this system operates on real-world inputs, Table 2 presents an excerpt from a dialogue and the extracted triplets.

Dialogue	Extracted triplets
- I also like football, I don't watch as often as I would like to though.	(I, like, football) (I, do not watch often, football)

Table 2: Extracted subject–predicate–object triplets from a sample dialogue.

3.3 The Associative Memory Agent

The core idea behind the associative memory agent is to treat the user input as a search engine query. This approach reframes the agent's task as a document ranking problem. While extensive research exists on information retrieval techniques (Kureichik and Gerasimenko, 2024) and (Huang et al., 2024), conventional methodologies seem to be unsuitable for our specific task. The crucial incompatibility arises from the fundamental difference in target data: traditional information retrieval methods typically operate on large-scale documents, while the Associative Memory Agent's task is to process triplets. Consequently, techniques such as inverted indexing, term-based search and tree search, optimized for larger text bodies, lack performance in this context.

The proposed solution leverages an embedding-based similarity search to retrieve relevant information. For each triplet extracted from user input (or the entire input string if no triplets are present) a vector embedding is generated using an arbitrary encoder. The cosine similarity is then computed between the query/triplet embedding and all existing embeddings within the database. The five most similar (by cosine similarity) facts are selected from the database and incorporated into a prompt for the LLM. Finally, the extracted triplets are appended to the database.

3.4 The Generation Agent

For our generation agent we used the Transformers library by HuggingFace[6] in order to make a generation pipeline. We chose Gemma3-1B-Instruct[7] as the model that generates the answer. Gemma 3 models follow the general decoder-only

[4] https://huggingface.co/spacy/en_core_web_trf
[5] https://spacy.io/universe/project/coreferee
[6] https://github.com/huggingface/transformers
[7] https://huggingface.co/google/gemma-3-1b-it

transformer architecture (Team G et al., 2025). The reason we chose it is because this model is light-weighted (only 1 billion parameters), therefore it is allowed to use it in real time with low resources.
In our generation agent we use two prompts: the query prompt (Table 3) and the system prompt (Table 4).

Your ROLE: assistant Your TASK: considering the FACTS about USER, give ANSWERS to his REPLIC. EXAMPLE: FACTS about USER: I am a surgeon, I am social with others, I got to the gym all the time, I like cats. USER SAYS: Do cats make good workout buddies? Your ANSWER: Cats are usually too lazy to join your workouts, but they're great at relaxing with you after the gym and the surgeries. Perfect for a hardworking doctor! FACTS about USER: {} USER SAYS: {} Your ANSWER:

Table 3: The query prompt; the curly brackets contain facts about the user and user's query.

The query prompt includes the current user's query, an instruction for the model and the facts about a user that were retrieved from previous queries.
System prompt is the main one. With this prompt we give the model the generation task and then specify it by saying about facts, context and the length of the answer that we expect. We instruct the model to generate a short answer (2-3 sentences), because without such a request, the model may not respond correctly and begin to reason.
Prompts are prepended to the message history (truncated to 300 tokens) and are submitted to the LLM with all previous context. If no history exists, the cold-start agent initializes the context. In response to queries, the model generates a response based on the facts extracted from the user's messages.

I need your help in the generation task. I will show you some facts about my persona (user). You are an assistant. Generate an answer only to the last user's message/query. Consider the previous context (messages) and facts. You should respond only in 2-3 sentences.

Table 4: The system prompt.

4 Evaluation

To evaluate persona usage during the conversation, a custom dataset was constructed based on the dialogue dataset MultiSessionChat. Our dataset contains 100 English dialogue sets, specifically selecting only a specific person turns within each dialogue. For each of these 100 sets, we manually extracted one fact and formulated a related question. After that, we employed the evaluation procedure for the Gemma 3 without and with our proposed pipeline. The evaluation process consisted of the following steps:

- An instance of a generation agent (either baseline Gemma 3 or chatbot pipeline) is initialized.
- For our pipeline, each of the 100 dialogue sets is processed by the fact extractor agent. This step fulfills a database for subsequent associative memory usage.
- For both approaches–the baseline Gemma 3 and our pipeline–the question, associated with the given dialogue, is posed to the generation agent by prompting. Before the response generation, our pipeline using retriever and collision resolving agents extracts relevant facts from the database and removes a conflicting information. For the baseline Gemma 3, we simply add a dialogue context and question to the prompt.
- Finally, we manually evaluate extracted answers with the golden answers from our constructed dataset.

The experimental design treats the series of 100 dialogue sets as a single broad conversation. This approach aims to assess the ability of the agents to maintain and utilize contextual information across multiple turns. Specifically, we hypothesize that baseline Gemma 3, operating with a limited or absent memory of past interactions, will exhibit a reduced ability to recall prior events compared to the chatbot, which is designed to retain and retrieve relevant facts from its associative memory database.
A total of eight experiments were conducted to evaluate the performance of a chatbot pipeline against a baseline Gemma 3 model. The experimental design varied two key factors: the

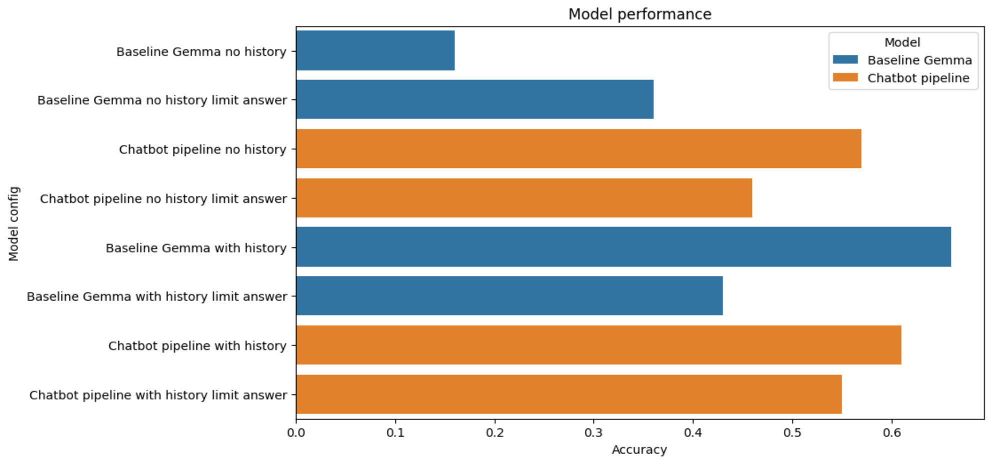


Figure 2: The pipeline performance compared to the baseline Gemma 3.

presence or absence of dialogue history, and the length limitations on the model's response.

5 Results

The results (Figure 2) indicate that the chatbot pipeline outperforms the baseline Gemma 3 model when dialogue history is absent. Specifically, a 41% improvement was observed without response length limitations, and a 16% improvement was observed with response length limitations. Furthermore, the chatbot pipeline outperformed baseline Gemma 3 even with dialogue history enabled in response length limitations conditions (12% margin). However, the chatbot pipeline did not surpass baseline Gemma 3 when both dialogue history and unlimited response lengths were employed. In this configuration, Gemma 3 achieved an accuracy of 66%, while the chatbot pipeline achieved an accuracy of 61%.

One potential reason why our pipeline has lower accuracy than the baseline is that the fact extraction agent extracts noisy information. However, it is worth noting that when using the pipeline without adding conversation history, the accuracy of our approach is almost comparable to using dialogue context. This suggests that our memory-based approach can potentially reduce the memory consumption of response generation in conversational agents.

6 Conclusion

In this study, we presented an approach to personalized chatbot construction by integrating an associative memory framework within a multi-agent pipeline. Through the implementation of the agents (handling cold-start, fact extraction, memory retrieval, and response generation) we demonstrated improvements in several cases in personalization and response accuracy. Thus, our results showed a 41% increase in performance over the baseline Gemma 3 model in memory-constrained settings without access to extended dialogue history.

7 Future Work

Since the fact extraction agent extracts noisy information, further work will be devoted to improving the accuracy of this agent. Since the agent produces false positives quite often, an additional classification model is needed to cope with this problem. The classification model should mark utterances that potentially contain facts. We assume that the combination of a classifier and a parser for fact extraction will reduce the amount of noisy data and, as a result, improve our pipeline.

The next step in our research will be to evaluate the proposed pipeline on other benchmarks. In particular, the LongMemEval (Wu et al., 2025) benchmark aims to evaluate the ability of language models to operate with memory. In this benchmark, there are many dialogues, each of which is divided into long sessions. Our approach to working with memory is close to RAG. Using a fact extraction agent, we can build a database that contains facts and indices of sessions or replicas that contain these facts. This will allow the generation agent to obtain more contextually relevant information for answering a question.

References

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen and Xia Hu. 2024. *LLM Maybe LongLM: Self-Extend LLM Context Window without Tuning.* arXiv preprint arXiv:2401.01325.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang and Mao Yang. 2024. *LongRoPE: Extending LLM Context Window beyond 2 Million Tokens*. arXiv preprint arXiv:2402.13753.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou and Zihan Qiu. 2025. *Qwen3 Technical Report.* 10.48550/arXiv.2505.09388.

Yi-Pei Chen, Noriki Nishida, Hideki Nakayama, and Yuji Matsumoto. 2024. *Recent Trends in Personalized Dialogue Generation: A Review of Datasets, Methodologies, and Evaluations*. 10.48550/arXiv.2405.17974.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. *Generating informative and diverse conversational responses via adversarial information maximization.* Advances in Neural Information Processing Systems, 31.

Qiao Qian, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. *Assigning personality/profile to a chatting machine for coherent conversation generation*. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, pages 4279– 4285. International Joint Conferences on Artificial Intelligence Organization.

Zhehao Zhang, Ryan A. Rossi, Branislav Kveton, Yijia *Shao, Diyi Yang, Hamed Zamani, Franck* Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, Ruiyi Zhang, Jiuxiang Gu, Tyler Derr, Hongjie Chen, Junda Wu, Xiang Chen, Zichao Wang, Subrata Mitra, Nedim Lipka, Nesreen Ahmed and Yu Wang. 2024. *Personalization of Large Language Models: A Survey*. 10.48550/arXiv.2411.00027.

Hongli Yuan and Alexander Hernandez. 2023. *User Cold Start Problem in Recommendation Systems: A Systematic Review.* IEEE Access. 10.1109/ACCESS.2023.3338705.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. *Lamp: When large language models meet personalization*. arXiv preprint arXiv:2304.11406.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, Maciej Kula, Ed Chi, and Maheswaran Sathiamoorthy. 2023. *Recommender systems with generative retrieval.* In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), Advances in Neural Information Processing Systems, volume 36, pp. 10299–10315. Curran Associates, Inc.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, Rui Zhang, and Yong Yu. 2023. *Towards open-world recommendation with knowledge augmentation from large language models*. arXiv preprint arXiv:2306.10933.

Xinchao Xu, Zhibin Gou, Wenquan Wu, Zheng-Yu Niu, Hua Wu, Haifeng Wang, and Shihang Wang. 2022. *Long Time No See! Open-Domain Conversation with Long-Term Persona Memory.* In Findings of the Association for Computational Linguistics: ACL 2022, pages 2639–2650, Dublin, Ireland. Association for Computational Linguistics.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. *Personalizing Dialogue Agents: I have a dog, do you have pets too?*. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Zhaojiang Lin, Peng Xu, and Pascale Fung. 2020. *Getting To Know You: User Attribute Extraction from Dialogues.* In Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 581–589, Marseille, France. European Language Resources Association.

Zhulin Wang, Xuhui Zhou, Rik Koncel-Kedziorski, Alex Marin and Fei Xia. 2022. *Extracting and Inferring Personal Attributes from Dialogue*. 58-69. 10.18653/v1/2022.nlp4convai-1.6.

Kai Zhang, Yejin Kim and Xiaozhong Liu. 2025. *Personalized LLM Response Generation with Parameterized Memory Injection.* arXiv preprint arXiv:2404.03565.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, Pranav Sandeep Dulepet, Saurav Vidyadhara, Dayeon Ki, Sweta Agrawal, Chau Pham, Gerson Kroiz, Feileen Li, Hudson Tao, Ashay Srivastava, Hevander Da Costa, Saloni Gupta, Megan L. Rogers, Inna Goncearenco, Giuseppe Sarli, Igor Galynker, Denis Peskoff, Marine Carpuat, Jules White, Shyamal Anadkat, Alexander Hoyle and Philip Resnik. 2024. *The Prompt Report: A Systematic Survey of Prompt Engineering Techniques*. Preprint at https://arxiv.org/abs/2406.06608.

Pegah Jandaghi, Xianghai Sheng, Xinyi Bai, Jay Pujara, and Hakim Sidahmed. 2024. *Faithful Persona-based Conversational Dataset Generation with Large Language Models.* In Proceedings of the 6th Workshop on NLP for Conversational AI (NLP4ConvAI 2024), pages 114–139, Bangkok, Thailand. Association for Computational Linguistics. American Psychological Association. 1983. Publications Manual. American Psychological Association, Washington, DC.

Jing Xu, Arthur Szlam, and Jason Weston. 2022. *Beyond Goldfish Memory: Long-Term Open-Domain Conversation.* In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5180–5197, Dublin, Ireland. Association for Computational Linguistics.

Nils Reimers and Irina Gurevych. 2019. *Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks*. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, 3-7 November 2019, 3982-3992. https://doi.org/10.18653/v1/d19-1410

Matthew Honnibal, Ines Montani, Sofie Van Landeghem and Adriane Boyd. 2020. *spaCy: Industrial-strength Natural Language Processing in Python*. https://doi.org/10.5281/zenodo.1212303

Vladimir Kureichik and Petr Gerasimenko. 2024. *Basic approaches to extracting textual information (overview)*. Izvestiya SfedU. Engineering sciences. 6-14. https://doi.org/10.18522/2311-3103 -2024-4-6

Junjie Huang, Jizheng Chen, Jianghao Lin, Jiarui Qin, Ziming Feng, Weinan Zhang and Yong Yu. 2024. *A Comprehensive Survey on Retrieval Methods in Recommender Systems*. arXiv preprint arXiv:2407.21022.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev and Léonard Hussenot. 2025. *Gemma 3 Technical Report*. 10.48550/arXiv.2503.19786.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, Dong Yu. 2025. *LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory.* 10.48550/arXiv.2410.10813.

Visualization of LLM Annotated Documents

Teodor Valchev, Nikolay Paev
Artificial Intelligence and Language Technology
Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
Bulgaria
teodorvulchev@gmail.com
nikolay.paev@iict.bas.bg

Abstract

Manual annotations play a crucial role in the Natural Language Processing domain. The paper presents an automatic annotation and visualization system for documents in the field of Social Studies and Humanities. The current annotation is on two levels, Named Entities and Events. The system combines automatically generated annotations from language models with a powerful text editor that is extended to accommodate manual annotation. The goal is to support the extraction of information from historical documents by scientists in the field of social studies and humanities. At the time of writing, the system is still in development.

1 Introduction

In this paper, we present the User interface (UI) to semantically annotated documents related to a knowledge graph representing the related knowledge of our CLaDA-BG[1] project. The aim of the developed system is to support the annotation of documents with the goal of expanding the Bulgarian-centric Knowledge graph and supporting researchers in the area of Social Sciences and Humanities (SS&H) in doing their investigations. The current architecture of the CLaDA-BG system is presented in Figure. 1. The main components of the architecture comprise (1) a Knowledge Graph and (2) Document Database that contain a large set of documents annotated with knowledge from the knowledge graph. The Knowledge Graph provides a contextualization of different datasets related to Bulgarian language, culture, and history. We call it *BGKG* (BulGarian-centric Knowledge Graph) because it represents main facts about people, settlements, locations, events, documents, organizations, etc. connected to Bulgaria. The Document Database contains a huge number of documents including archive documents, newspaper articles, letters, papers, description of artifacts, etc. Documents are annotated with concepts or instances from the knowledge graph. The annotation of documents supports search via queries expressed as textual elements, concepts, and facts defined in the terms of BGKG. The queried documents are post processed in different ways. The two main ones are: (1) ranking with respect to the query terms, and (2) extraction of new knowledge from extracted documents. This architecture assumes various types of users including at least the following ones: researchers, BGKG curators and Documents annotators. Researchers access the systems in order to find the necessary documents supporting their research. They produce new research represented as documents similar to the ones within the Document database. The BGKG curators manage the knowledge within it by checking its correctness, mapping different representations of the same knowledge, and adding new information. Annotators perform annotation of the documents manually or semi-manually, usually as a post editing after automatic annotation.

In our view this architecture is a way to provide access to NLP technologies to end users (researchers, teachers, etc.) who are not familiar (and not willing to become familiar) with these technologies. Thus, they will prefer to work as they are used to in their research. Our observations are that researchers in the area of SS&H usually are working with WYSIWYG[2] editors such as MS Word, Google Docs. Therefore, we consider as the main component of the UI a structural editor in which

[1]CLaDA-BG is a Bulgarian national research infrastructure for resources and technologies for linguistic, cultural and historical heritage, integrated within CLARIN EU and DARIAH EU.

[2]WYSIWYG stands for *What You See Is What You Get*

Proceedings of the Student Research Workshop associated with RANLP-2025, pages 70–77, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/issn.2603-2821.2025_009

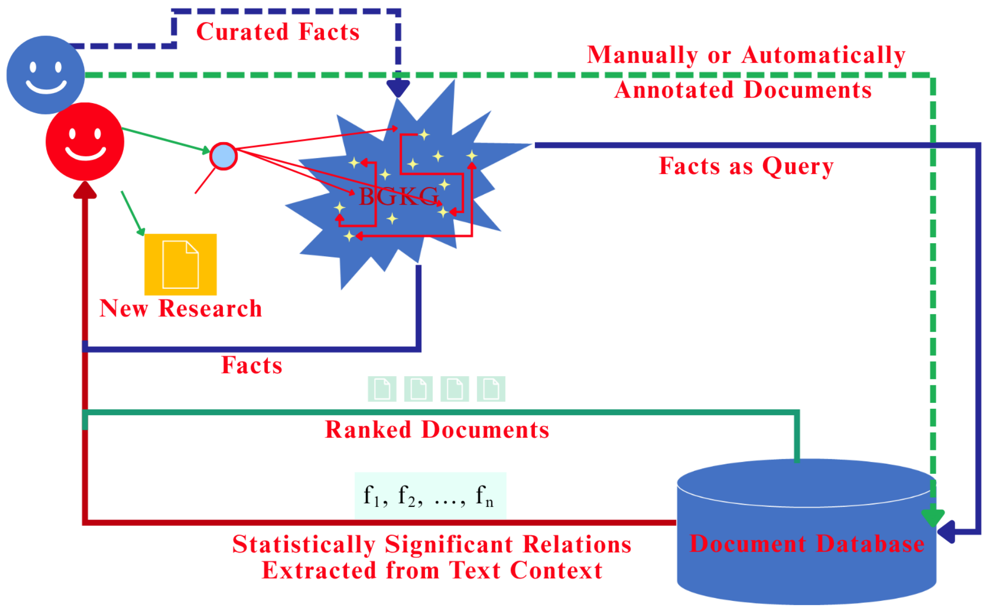


Figure 1: The Architecture of the CLaDA-BG Project.

the user can create new documents describing their new research, taking notes, etc.

Thus, the same editor is used to examine selected documents in the database, taking notes, annotations, and corrections. The editor has to support creation of well formatted documents representing scientific papers. Additionally, it has to be easily extendable to represent complex annotations across the content of the documents. As representation of the documents, we consider XML version of HTML (XHTML) — requiring the HTML document to be well-formed XML document. Having such well-formed XHTML documents makes it easy to add a minimal number of non-standard elements and attributes.

In this paper, we present the structure of the annotation of the documents, the automatic annotation, the architecture of the editor and its functionalities - visualization, (partial) re-annotation, linking the document to the knowledge graph and to annotated documents.

The structure of the paper is as follows: in the next Section 2 some works are discussed in relation to our work. In Section 3 the overall system architecture is presented with the document representation, database servers, the local LLM (Large Language Models) server, and the UI. Section 4 presents the workflow of the system, and Section 5 concludes the paper.

2 Related Works

Manual annotation of documents is a crucial step in all natural language processing tasks. The paper is concerned with the UI to support the work of the main types of users of our system. We consider as related works mainly systems for manual annotation of documents.

For many years, we used the CLaRK System (Simov et al., 2001) for corpus annotation, lexicon development, and more. The system main interface is an XML editor. In addition several tools for processing XML documents are internally implemented, such as Regular grammars over XML documents, constrains for validations of different annotation and/or insertions of valid XML fragments. The tools of CLaRK System allow us to solve most of the processing that we wanted to implement. But the system has some shortcomings. First, it is not connected to any external databases. Thus, users need to take care of document management by themselves. Second, the tools require knowledge of XML related technologies such as XPath[3] which is very powerful for processing XML document, but they are a burden for many of the potential users. In addition, the editor does not support any formatting instructions, making the system difficult for unfamiliar users. Thus, our work here draws on our experience with the CLaRK system.

[3] XML Path Language (XPath): https://www.w3.org/TR/xpath/

We have similar experience with the following systems: the GATE Teamware — (Bontcheva et al., 2013), the INCEpTION platform — (Klie et al., 2018), SpaCy: Industrial-Strength Natural Language Processing[4]. All of them have functionality for creation of rules for automatic text processing including regular expression rules, programming languages — Java, Python, for processing the predefined document data models. They allow for calling external processing tools including machine learning models, large language models (LLMs), etc. Behind these functionalities, these tools provide document visualization of the annotations, some of which we incorporate in our work. Such as coloring schema styles, tooltips, etc.

In our case, the main deviations from these tools are that we need a WYSIWYG editor[5] integrated with the rest of the architecture of our system. This is important because researchers value the structural presentation of documents, not just their content. This applies not only to their own documents, but also to the documents they use in their research.

Neves and Ševa provide a comprehensive review of manual annotation tools — (Neves and Ševa, 2019). They defined a set of evaluation criteria for what makes an annotation tool useful. Their results show that none of the tools they reviewed met their criteria fully (Functional, Data, and others). As a selection criteria for tools to be extensively reviewed in their study, they used: (1) availability, (2) to be web accessible (downloadable or online), (3) to be easy to install, (4) working for their field of studies, and (5) to allow definition of annotation schema. Of the 78 tools they considered, 63 were not selected for a detailed evaluation because they did not meet at least one of the five requirements.

3 System architecture

In this section, we present the document representation and the main components of our system - the backend server, databases, automatic annotation server, and UI.

3.1 User interface

The developed version of our software, as is currently, meets: web, easy to install, working in their field of studies. In the future, we plan to allow schema configuration and allow open availability.

[4]https://spacy.io/

[5]Such type of editors are most frequently used by researchers in SS&H area. Thus, they reflect their experience.

3.2 Document representation

As was mentioned above, we need to define an extended version XHTML. The main idea is to use XHTML to support the format of the original papers that are annotated and uploaded to the system or the paper created by the user in their own research activities. In order to perform experiments with the extended version of the basis XHTML format, we select an existing freely available web based HTML editor, which is not focused on annotation: the TinyMCE rich text editor[6] (GPL licensed version[7]).

We have experimented with several schemes for representation of annotation data. The result of these experiments shows that using more than one element which allows inclusion of several annotation elements the editing of the annotations and their interaction with the standard XHTML elements complicate the editing process. Thus, we decided to minimize the number of new elements. Experiments were performed using the TinyMCE Annotations API, but the span approach made the represention of overlapping annotation not user friendly in the resulting XHTML. Spacy annotation tool was reviewed and as a result only one new type of elements $< tok > token < /tok >$ is added with a number of new custom attributes. The extension of XHTML with this type of elements is call *cladaHTML*. More detailed explanation is available in Subsection 3.1.

The performed experiments using multiple tags showed that issues may arise, caused by mixing of the representations of the structured annotated document and the stylization.

Documents are represented in the database as tables of tokens, sentences, annotations, and known facts.

Usage of just one element seems too small addition, but representation over tokens allows for complex structures of annotations within Universal Dependencies CoNLL-U format[8]. Many other projects are using variants of CoNLL format.

```
<b><i>Text</b></i>
```

Listing 1: Crossing HTML elements which is erroneous in general HTML, and not well formed in XHTML.

[6]https://www.tiny.cloud/

[7]https://github.com/tinymce/tinymce

[8]https://universaldependencies.org/format.html

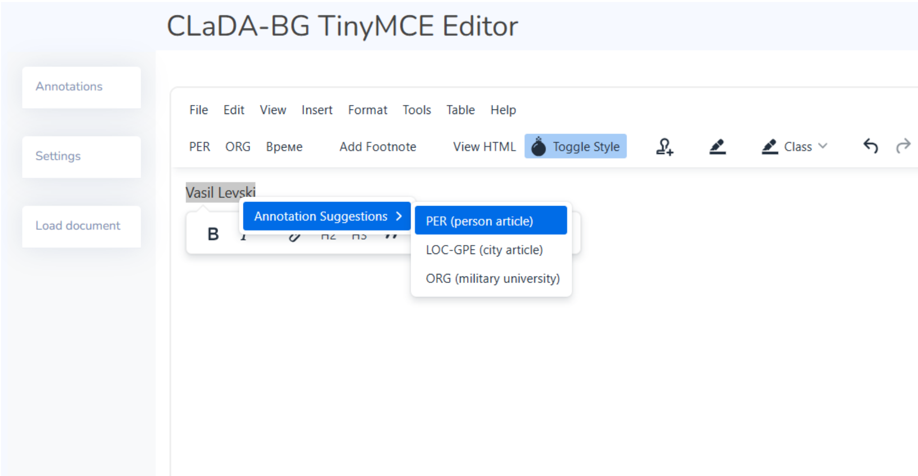


Figure 2: Annotation recommendation

```
<sentence id="1">
  <anot class= "class-1">
    <tok>The </tok>
    <anot class= "class-2">
      <tok>sun</tok>
      <tok>is </tok></anot>
    <tok>shining </tok>
  </anot>
</sentence>
```

Listing 2: Annotated example sentence with multiple elements

3.3 Back-end

In the back-end, there are a builder and a destructor for the cladaHTML, which build the documents from the tables in the relational database or convert them to database SQL queries to update the information in the tables. The relational database model is used to represent the documents with annotations, tokenization, etc. The scheme of the database is specified by the CLaDA-BG team and allows for searching of facts in the documents and in related documents, mentions, and more. The relational database model is also used as an intermediate representation of the documents annotated by LLMs.

3.4 Knowledge and Documents database

The main database for storing documents has the following tables with appropriate relations: Documents, Events, NEs (Named entities), Roles, Sentences, Tokens, and URLs. They allow for search queries like: All the documents where some Event/NEs is mentioned, searching for documents with close sentences, etc.

Using URLs, we can identify different occurrences (different names, pronouns, etc.) of the same object in the same or between multiple documents. The records in the database are structured in a way that allows for easy building of a fully functional knowledge graph.

The UI is web-based and is built on top of TinyMCE Text Editor, extended with JavaScript code. A screenshot of the UI is provided in Figure 3. A custom footnote and endnote changes tracking assistant is implemented. Coloring is achieved using Cascading Style Sheets (CSS) technology, but due to limitations in most browsers, only one rule per class from the same type can be visualized at the same time. To bypass this restriction, dynamic CSS coloring rules (single- and multiclass) are generated in the browser as the document is loaded in the editor. Rules are generated only for available combinations of classes, so we save ourselves from generating all possible combinations of classes and the linked exponential growth of all subsets.

The TinyMCE text editor internally is representing the document in HTML format (setting is available for XHTML), and allows the definition of custom tags. Only one custom tag $< tok >< /tok >$ with custom attributes is added, dividing the tokens. We call this language extension cladaHTML as mentioned earlier. In that way, we are preserving the behavior of all features of the editor and simultaneously adding new functionality. The reason for using only a single new tag is that in XHTML almost all elements must have a parent element, and tag misnesting[9] is not allowed. Misnesting occurs when XHTML tags are not properly nested, meaning that the order in which tags are closed does not

[9] https://w3c.github.io/html-reference/syntax.html

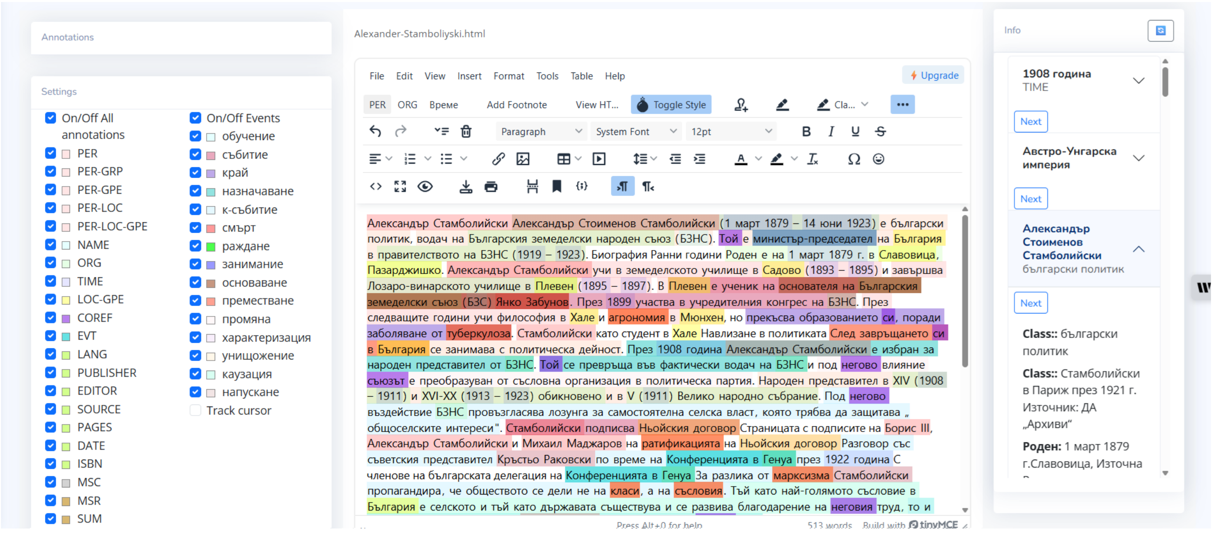


Figure 3: A screenshot of the UI showing an annotated document. The left column displays the annotation coloring toggles. The center window is the interface of the extended TinyMCE editor. The right column shows information from the knowledge base for the entities in the document.

match the order in which they were opened. Listing 1 is a small example of misnesting. In (Simov et al., 2001) a similar issue is presented, but in XML, typically called "not well-formed XML document". The issues that may arise when styling the document or using another functionality, even built in the editor, are linked to the tags representing them. It may not be possible to style the desired chunks of text, for example: parts of heading text, paragraphs, and others, because there are multiple tags representing annotations, sentences, and others which overlap, leading to tag misnesting when styling parts of sentence with a lot of overlapping of annotations, when representing annotations with custom tags. Consider the example in Listing 2. If we need to style only the class-1 annotated part of the text, it becomes impossible because annotated class-2 is started in class-1. If styling is done with XHTML tags (spans, divs, italic, and other tags) due to limitations in tag nesting in XHTML the entire sentences must be styled, so the XHTML remains valid. Other custom mechanisms for styling, rules, or CSS can be used, which also depend on the chosen text editor.

In the implemented solution with a single new tag, the annotations are represented as a multiclass attribute, each token has a unique ID per document. We did not find any major responsiveness issues while working with longer documents. Other functionalities like sentences, tooltip, etc. are implemented using custom attributes to store desired information and behavior. In that context, changing annotation of text is actually a change in the class attributes of the tokens. (Listing 3)

Annotation suggestions are displayed to the user, which can be generated with database queries, LLMs or by traversal of the knowledge graph. The user can also review stored information for the suggestion in order to make the best decision for annotation and URL linking. Figure 2 shows an example of the suggestions.

We need to point out a subtle but crucial detail: TinyMCE uses a non-standard XHTML attributes internally, which may not show in the "View source" option, but causes confusion during development. One example of that is the usage of "data-mce-href" hidden attribute (instead of the direct usage of the "href" attribute) which is used to keep track of the original link, during editing, or transformations for different reasons.

3.5 Automatic annotation

The automatic annotation pipeline is a core feature of the system. When the user uploads a brand new document to the database, the NLP pipe first extracts the text, tokenizes it, and segments it into sentences.

The system pipeline then applies the language models that annotate the named entities and the event structure in the text, providing the initial annotations of the documents. We use our own pretrained and later fine-tuned BERT (Devlin et al., 2019) and T5 (Raffel et al., 2019) models for the tasks. The models were pretrained on 20B and 35B

Bulgarian corpora respectively.

```
<tok id="1" class="PER" neurl="Alexander
     Stamboliyski" titlenes="PER"
     sentence="1">Alexander</tok>

<tok id="2" class="PER"
     neurl="Alexander_Stamboliyski" titlenes="PER"
     sentence="1">Stamboliyski</tok>

<tok id="3" class="death birth occupation"
     titlenes="death birth occupation"
     sentence="2">(</tok>

<tok id="4" class="TIME death birth occupation"
     titlenes="TIME death birth occupation"
     sentence="2">1</tok>

<tok id="5" class="TIME death birth occupation"
     titlenes="TIME death birth occupation"
     sentence="2">March</tok>

<tok id="6" class="TIME death birth occupation"
     titlenes="TIME death birth occupation"
     sentence="2">1879</tok>

<tok id="7" class="death birth occupation"
     titlenes="death birth occupation"
     sentence="2">-</tok>

<tok id="8" class="TIME death birth occupation"
     titlenes="TIME death birth occupation"
     sentence="2">14</tok>

<tok id="9" class="TIME death birth occupation"
     titlenes="TIME death birth occupation"
     sentence="2">June</tok>

<tok id="10" class="TIME death birth occupation"
     titlenes="TIME death birth occupation"
     sentence="2">1923</tok>

<tok id="11" class="death birth occupation"
     titlenes="death birth occupation"
     sentence="2">)</tok>

<tok id="12" class="occupation"
     titlenes="occupation" sentence="2">is</tok>

<tok id="13" class="occupation"
     titlenes="occupation"
     sentence="2">bulgarian</tok>

<tok id="14" class="occupation"
     titlenes="occupation"
     sentence="2">politician.</tok>
```

Listing 3: An annotated sentence in the CLaDA-BG-HTML format. (The original sentence is in Bulgarian.)

The processing by the models is done on the sentence level. The BERT model is used for the recognition of named entities and classifies tokens in the classic BOI format. The names are later mapped to their specific URLs in the Knowledge Base. The best model we created achieves a macro-F1 score of 81.23%. Experiments regarding entity disambiguation with fine-tuning models for retrieval (bi-encoders and cross-encoders) are also made, but are still in an early stage and will be presented in the future.

The event extraction is done with the T5 model which processes the sentences and generates the event structure into a JSON compatible format. The output contains a list of events described by event type, event text span, and a list of roles and their text spans. The predicted texts are fuzzy matched to the input tokens of the sentence, in order to get the token ids of the spans. The model achieves an F1 score of 84.29% in the extraction of test events. The models are fine-tuned on the latest version of the Bulgarian Event Corpus (Osenova et al., 2022). The development of the corpus and the models is described in more detail in (Simov et al., 2025). The annotation subsystem is designed as an internal REST API which is called by the back-end server on every update of the documents. The annotation returned by the pipeline is then stored in the database and later cladaHTML is generated from it.

4 Workflow

The main workflow of the system is:

- The user uploads a document from docx/markdown or creates a plain document which is represented internally in cladaHTML.[10]
- The user can edit or style the document.
- When the document is saved on the server, it is sent to the LLM server for automatic annotation, then it is returned to the back-end and saved in the relational database form.
- After the document is processed, it is returned in cladaHTML to the UI, the user can edit, style, edit annotations, create a new annotation, etc. and of course save it again, create a new version for the document, download it locally, or share it with another user. A diagram is presented in Figure 4.
- When the user is working on a document, he/she could perform different types of search in the database for additional information related to the annotations, saved in the BGKG, or to access other documents.

5 Conclusions and Future Work

In the paper, we presented a web based annotation system that allows for editing and stylization of documents in a user friendly way. The system leverages the use of LLMs for automatic annotations and initial annotation suggestions. Our main contributions are: (1) implementation of an extension of

[10]The uploading of a set of many documents will be implemented as an offline services in the system.

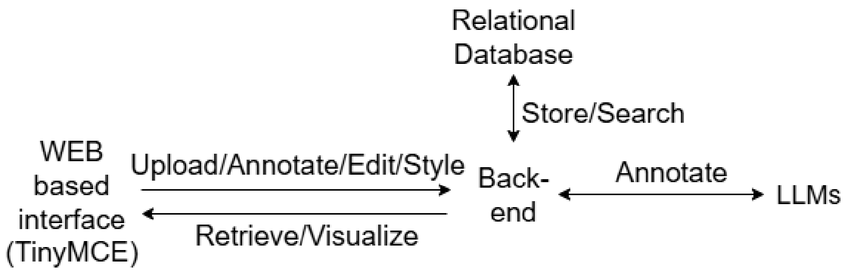


Figure 4: Interaction diagram of the subsystems.

XHTML to incorporate a token-based annotation of XHTML documents; (2) The TinyMCE rich text editor was extended to visualize the annotation of such documents and to allow for manual annotation. We are also working on manual modification of the automatic annotation; (3) A mechanism for annotation of XHTML with named entities and events. We think that in this way we provide the NLP technologies to the end users without a need for them to know the details of these technologies.

The specified internal document format cladaHTML is compatible with the standard features of the core TinyMCE editor, extending its functionality. The system is web-based, no installation is needed, easy to work with, and not computationally demanding. Developers should watch out for rich text editors adding hidden tags and merging multiple same-type tags, which may not appear even in source view.

In the future, we plan to work in two directions: (1) Integration with other components of the whole architecture, presented in the introduction; (2) Extension of the functionalities presented in the paper.

Plans for future work include: Support for importing from PDF, ability to do OCR and support for older or ancient languages. Support for exporting in docx format and as interactive document for embedding in web-pages in the format of: XHTML, CSS and JavaScript document, so some interaction with the document is possible outside the editor. Stylization of plain documents with LLMs.

We plan to extend the LLMs to support the editing process for spell checking, linguistic ambiguity, and others as needed. Although in the paper we referred to CoNLL in the context of Universal Treebanks we believe that a format based on tokens could incorporate not only syntactic annotation, but any annotations over text.

In the paper we provide integration of the implemented editor with a selected document. The more complicated searches that are represented shortly in the introduction. Currently we are working on a creation of RAG (Retrieval-Augmented Generation) system — see (Gao et al., 2024) for a Survey. Such a system will provide a more flexible way of searching the document database. In this way we will be able to rank the appropriate documents as mentioned earlier. We plan to implement a system to extract new knowledge from selected documents. The form of the knowledge will depend on the conceptual knowledge in BGKG — the ontological knowledge, the instance information and syntactic structure of the text. The significance of this new knowledge with respect to the selected documents will be determined by evaluating the extracted new facts as key ones.

Open-source version is considered after the production phase is achieved. For now, the system is tied to our requirements, but a modular approach can be implemented.

Acknowledgments

The reported work has been supported by CLaDA-BG, *the Bulgarian National Interdisciplinary Research e-Infrastructure for Resources and Technologies in favor of the Bulgarian Language and Cultural Heritage, part of the EU infrastructures CLARIN and DARIAH*. We also acknowledge the provided access to *the e-infrastructure of the Centre for Advanced Computing and Data Processing* (the Grant No BG05M2OP001-1.001-0003).

We would like to thank the CLaDA-BG team for the nice working atmosphere and their comments on the earlier versions of the text. Also, we would like to thank the reviewers for their valuable and insightful comments.

References

Kalina Bontcheva, Hamish Cunningham, Ian Roberts, Angus Roberts, Valentin Tablan, Niraj Aswani, and Genevieve Gorrell. 2013. Gate teamware: a web-based, collaborative text annotation framework. *Language Resources and Evaluation*, 47(4):1007–1029.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. Association for Computational Linguistics.

Mariana Neves and Jurica Ševa. 2019. An extensive review of tools for manual annotation of documents. *Briefings in Bioinformatics*, 22(1):146–163.

Petya Osenova, Kiril Simov, Iva Marinova, and Melania Berbatova. 2022. The Bulgarian event corpus: Overview and initial NER experiments. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3491–3499, Marseille, France. European Language Resources Association.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Kiril Simov, Nikolay Paev, Petya Osenova, and Stefan Marinov. 2025. Bulgarian event extraction with llms. Presented at RANLP2025.

Kiril Simov, Zdravko Peev, Milen Kouylekov, Alexander Simov, Marin Dimitrov, and Atanas Kiryakov. 2001. CLaRK — an XML-Based System for Corpora Development. In *Proc. of the Corpus Linguistics 2001 Conference*, pages 558–560.

Author Index