

Parsing the SYNTAGRUS Treebank of Russian

Joakim Nivre
Växjö University and
Uppsala University
joakim.nivre@vxu.se

Igor M. Boguslavsky
Universidad Politécnica
de Madrid
Departamento de
Inteligencia Artificial
igor@opera.dia.fi.upm.es

Leonid L. Iomdin
Russian Academy
of Sciences
Institute for Information
Transmission Problems
iomdin@iitp.ru

Abstract

We present the first results on parsing the SYNTAGRUS treebank of Russian with a data-driven dependency parser, achieving a labeled attachment score of over 82% and an unlabeled attachment score of 89%. A feature analysis shows that high parsing accuracy is crucially dependent on the use of both lexical and morphological features. We conjecture that the latter result can be generalized to richly inflected languages in general, provided that sufficient amounts of training data are available.

1 Introduction

Dependency-based syntactic parsing has become increasingly popular in computational linguistics in recent years. One of the reasons for the growing interest is apparently the belief that dependency-based representations should be more suitable for languages that exhibit free or flexible word order and where most of the clues to syntactic structure are found in lexical and morphological features, rather than in syntactic categories and word order configurations. Some support for this view can be found in the results from the CoNLL shared tasks on dependency parsing in 2006 and 2007, where a variety of data-driven methods for dependency parsing have been applied with encouraging results to languages of great typological diversity (Buchholz and Marsi, 2006; Nivre et al., 2007a).

However, there are still important differences in parsing accuracy for different language types. For

example, Nivre et al. (2007a) observe that the languages included in the 2007 CoNLL shared task can be divided into three distinct groups with respect to top accuracy scores, with relatively low accuracy for richly inflected languages like Arabic and Basque, medium accuracy for agglutinating languages like Hungarian and Turkish, and high accuracy for more configurational languages like English and Chinese. A complicating factor in this kind of comparison is the fact that the syntactic annotation in treebanks varies across languages, in such a way that it is very difficult to tease apart the impact on parsing accuracy of linguistic structure, on the one hand, and linguistic annotation, on the other. It is also worth noting that the majority of the data sets used in the CoNLL shared tasks are not derived from treebanks with genuine dependency annotation, but have been obtained through conversion from other kinds of annotation. And the data sets that do come with original dependency annotation are generally fairly small, with less than 100,000 words available for training, the notable exception of course being the Prague Dependency Treebank of Czech (Hajič et al., 2001), which is one of the largest and most widely used treebanks in the field.

This paper contributes to the growing literature on dependency parsing for typologically diverse languages by presenting the first results on parsing the Russian treebank SYNTAGRUS (Boguslavsky et al., 2000; Boguslavsky et al., 2002). There are several factors that make this treebank an interesting resource in this context. First of all, it contains a genuine dependency annotation, theoretically grounded in the long tradition of dependency grammar for Slavic languages, represented by the work of Tesnière (1959) and Mel'čuk (1988), among others. Secondly, with close to

© Joakim Nivre, Igor M. Boguslavsky, and Leonid L. Iomdin, 2008. Licensed under the *Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported* license (<http://creativecommons.org/licenses/by-nc-sa/3.0/>). Some rights reserved.

500,000 tokens, the treebank is larger than most other available dependency treebanks and provides a good basis for experimental investigations using data-driven methods. Thirdly, the Russian language, which has not been included in previous experimental evaluations such as the CoNLL shared tasks, is a richly inflected language with free word order and thus representative of the class of languages that tend to pose problems for the currently available parsing models. Taken together, these factors imply that experiments using the SYNTAGRUS treebank may be able to shed further light on the complex interplay between language type, annotation scheme, and training set size, as determinants of parsing accuracy for data-driven dependency parsers.

The experimental parsing results presented in this paper have been obtained using MaltParser, a freely available system for data-driven dependency parsing with state-of-the-art accuracy for most languages in previous evaluations (Buchholz and Marsi, 2006; Nivre et al., 2007a; Nivre et al., 2007b). Besides establishing a first benchmark for the SYNTAGRUS treebank, we analyze the influence of different kinds of features on parsing accuracy, showing conclusively that both lexical and morphological features are crucial for obtaining good parsing accuracy. All results are based on input with gold standard annotations, which means that the results can be seen to establish an upper bound on what can be achieved when parsing raw text. However, this also means that results are comparable to those from the CoNLL shared tasks, which have been obtained under the same conditions.

The rest of the paper is structured as follows. Section 2 introduces the SYNTAGRUS treebank, section 3 describes the MaltParser system used in the experiments, and section 4 presents experimental results and analysis. Section 5 contains conclusions and future work.

2 The SYNTAGRUS Treebank

The Russian dependency treebank, SYNTAGRUS, is being developed by the Computational Linguistics Laboratory, Institute of Information Transmission Problems, Russian Academy of Sciences. Currently the treebank contains over 32,000 sentences (roughly 460,000 words) belonging to texts from a variety of genres (contemporary fiction, popular science, newspaper and journal articles

dated between 1960 and 2008, texts of online news, etc.) and it is growing steadily. It is an integral but fully autonomous part of the Russian National Corpus developed in a nationwide research project and can be freely consulted on the Web (<http://www.ruscorpora.ru/>).

Since Russian is a language with relatively free word order, SYNTAGRUS adopted a dependency-based annotation scheme, in a way parallel to the Prague Dependency Treebank (Hajič et al., 2001). The treebank is so far the only corpus of Russian supplied with comprehensive morphological annotation and syntactic annotation in the form of a complete dependency tree provided for every sentence.

Figure 1 shows the dependency tree for the sentence *Наибольшее возмущение участников митинга вызвал продолжающийся рост цен на бензин, устанавливаемых нефтяными компаниями* (It was the continuing growth of petrol prices set by oil companies that caused the greatest indignation of the participants of the meeting). In the dependency tree, nodes represent words (lemmas), annotated with parts of speech and morphological features, while arcs are labeled with syntactic dependency types. There are over 65 distinct dependency labels in the treebank, half of which are taken from Mel'čuk's Meaning \Leftrightarrow Text Theory (Mel'čuk, 1988). Dependency types that are used in figure 1 include:

1. *предик* (predicative), which, prototypically, represents the relation between the verbal predicate as head and its subject as dependent;
2. *1-компл* (first complement), which denotes the relation between a predicate word as head and its direct complement as dependent;
3. *агент* (agentive), which introduces the relation between a predicate word (verbal noun or verb in the passive voice) as head and its agent in the instrumental case as dependent;
4. *квазиагент* (quasi-agentive), which relates any predicate noun as head with its first syntactic actant as dependent, if the latter is not eligible for being qualified as the noun's agent;
5. *опред* (modifier), which connects a noun head with an adjective/participle dependent if the latter serves as an adjectival modifier to the noun;



Figure 1: A syntactically annotated sentence from the SYNTAGRUS treebank.

6. предл (prepositional), which accounts for the relation between a preposition as head and a noun as dependent.

Dependency trees in SYNTAGRUS may contain non-projective dependencies. Normally, one token corresponds to one node in the dependency tree. There are however a noticeable number of exceptions, the most important of which are the following:

1. compound words like *пятидесятиэтажный* (fifty-storied), where one token corresponds to two or more nodes;
2. so-called phantom nodes for the representation of hard cases of ellipsis, which do not correspond to any particular token in the sentence; for example, *Я купил рубашку, а он галстук* (I bought a shirt and he a tie), which is expanded into *Я купил рубашку, а он купил_{PHANTOM} галстук* (I bought a shirt and he bought_{PHANTOM} a tie);
3. multiword expressions like *по крайней мере* (at least), where several tokens correspond to one node.

Syntactic annotation is performed semi-automatically: sentences are first processed by the rule-based Russian parser of an advanced NLP system, ETAP-3 (Apresian et al., 2003) and then edited manually by linguists who handle errors of the parser as well as cases of ambiguity

that cannot be reliably resolved without extra-linguistic knowledge. The parser processes raw sentences without prior part-of-speech tagging.

Morphological annotation in SYNTAGRUS is based on a comprehensive morphological dictionary of Russian that counts about 130,000 entries (over 4 million word forms). The ETAP-3 morphological analyzer uses the dictionary to produce morphological annotation of words belonging to the corpus, including lemma, part-of-speech tag and additional morphological features dependent on the part of speech: animacy, gender, number, case, degree of comparison, short form (of adjectives and participles), representation (of verbs), aspect, tense, mood, person, voice, composite form, and attenuation.

Statistics for the version of SYNTAGRUS used for the experiments described in this paper are as follows:

- 32,242 sentences, belonging to the fiction genre (9.8%), texts of online news (12.4%), newspaper and journal articles (77.8%);
- 461,297 tokens, including expressions with non-alphabetical symbols (e.g., 10, 1.200, \$333, +70С, #) but excluding punctuation;
- 31,683 distinct word types, of which 635 with a frequency greater than 100, 5041 greater than 10, and 18231 greater than 1;
- 3,414 sentences (10.3%) with non-projective

	POS	DEP	MOR	LEM	LEX
TOP	+	+	+	+	+
TOP-1	+				
HEAD(TOP)				+	+
LDEP(TOP)		+			
RDEP(TOP)		+			
NEXT	+		+	+	+
NEXT+1	+		+	+	+
NEXT+2	+				
NEXT+3	+				
LDEP(NEXT)		+			

Table 1: History-based features (TOP = token on top of stack; NEXT = next token in input buffer; HEAD(w) = head of w ; LDEP(w) = leftmost dependent of w ; RDEP(w) = leftmost dependent of w).

dependencies and 3,934 non-projective dependency arcs in total;

- 478 sentences (1.5%) containing phantom nodes and 631 phantom nodes in total.

3 MaltParser

MaltParser (Nivre et al., 2007b) is a language-independent system for data-driven dependency parsing, based on a transition-based parsing model (McDonald and Nivre, 2007). More precisely, the approach is based on four essential components:

- A transition-based deterministic algorithm for building labeled projective dependency graphs in linear time (Nivre, 2003).
- History-based feature models for predicting the next parser action (Black et al., 1992; Magerman, 1995; Ratnaparkhi, 1997).
- Discriminative classifiers for mapping histories to parser actions (Kudo and Matsumoto, 2002; Yamada and Matsumoto, 2003).
- Pseudo-projective parsing for recovering non-projective structures (Nivre and Nilsson, 2005).

In the following subsections, we briefly describe each of these four components in turn.

3.1 Parsing Algorithm

The parser uses the deterministic algorithm for labeled dependency parsing first proposed by Nivre (2003). The algorithm builds a labeled dependency graph in one left-to-right pass over the input, using a stack to store partially processed tokens and

adding arcs using four elementary actions (where TOP is the token on top of the stack and NEXT is the next token):

- **Shift**: Push NEXT onto the stack.
- **Reduce**: Pop the stack.
- **Right-Arc(r)**: Add an arc labeled r from TOP to NEXT; push NEXT onto the stack.
- **Left-Arc(r)**: Add an arc labeled r from NEXT to TOP; pop the stack.

Parser actions are predicted using a history-based feature model (section 3.2) and SVM classifiers (section 3.3). Although the parser only derives projective graphs, the fact that these graphs are labeled allows non-projective dependencies to be captured using the pseudo-projective approach of Nivre and Nilsson (2005) (section 3.4).

3.2 History-Based Feature Models

History-based parsing models rely on features of the derivation history to predict the next parser action (Black et al., 1992). The features used are all symbolic and defined in terms of five different node attributes:

- POS = part of speech (atomic)
- DEP = dependency type
- MOR = morphological features (set)
- LEM = lemma
- LEX = word form

Features of the type DEP have a special status in that they are extracted during parsing from the partially built dependency graph and are updated dynamically during parsing. The other four feature types (LEX, LEM, POS, and MOR) are given as part of the input to the parser and remain static during the processing of a sentence. Of these four feature types, all except LEX presupposes that the input has been preprocessed by a lemmatizer, tagger and morphological analyzer, respectively, but for the experiments reported below we use gold standard annotation from the treebank.

In order to study the influence of different features, we have experimented with different combinations of the five feature types, where the baseline model contains only POS and DEP features, while more complex models add MOR, LEM, and LEX features in different combinations. The exact

features included for each feature type are shown in table 1, where rows denote tokens in a parser configuration (defined relative to the stack, the remaining input, and the partially built dependency graph), and where columns correspond to feature types. The selection of features in each group was tuned on a development set as described in section 4.

3.3 Discriminative Classifiers

We use support vector machines (Vapnik, 1995) to predict the next parser action from a feature vector representing the history. More specifically, we use LIBSVM (Chang and Lin, 2001) with a quadratic kernel $K(x_i, x_j) = (\gamma x_i^T x_j + r)^2$ and the built-in one-versus-all strategy for multi-class classification. Symbolic features are converted to numerical features using the standard technique of binarization, and we split the set values of MOR features into their atomic components. In order to speed up training, we also divide the training data into smaller bins according to the feature POS of NEXT, and train separate classifiers on each bin.

3.4 Pseudo-Projective Parsing

Pseudo-projective parsing was proposed by Nivre and Nilsson (2005) as a way of dealing with non-projective structures in a projective data-driven parser. We projectivize training data by a minimal transformation, lifting non-projective arcs one step at a time, and extending the arc label of lifted arcs using the encoding scheme called HEAD by Nivre and Nilsson (2005), which means that a lifted arc is assigned the label $r \uparrow h$, where r is the original label and h is the label of the original head in the non-projective dependency graph.

Non-projective dependencies can be recovered by an inverse transformation applied to the dependency graph output by the parser, using a left-to-right, top-down, breadth-first search, guided by the extended arc labels $r \uparrow h$ assigned by the parser.

4 Experiments

In this section we describe the first experiments on parsing the SYNTAGRUS treebank using a data-driven parser. The experimental setup is described in section 4.1, while the experimental results are presented and discussed in section 4.2.

4.1 Experimental Setup

All experiments have been performed on the version of SYNTAGRUS described in section 2, con-

Model	Count	LAS	UAS
Base = POS + DEP	46506	60.2	76.0
B1 = Base + MOR	46506	73.0	84.5
B2 = Base + LEM	46506	75.5	84.6
B3 = Base + LEX	46506	74.5	84.6
BM1 = B1 + LEM	46506	82.3	89.0
BM2 = B1 + LEX	46506	81.0	88.8
All = B1 + LEM + LEX	46506	82.3	89.1

Table 2: Parsing accuracy for different feature models on the final test set (Count = Number of tokens in the test set, LAS = Labeled attachment score, UAS = Unlabeled attachment score).

verted to the CoNLL data format (Buchholz and Marsi, 2006).¹ The available data were divided into 80% for training, 10% for development, and 10% for final testing, using a pseudo-randomized split. The development set was used for tuning parameters of the parsing algorithm and pseudo-projective parsing technique, and for feature selection within the feature groups not included in the baseline model (i.e., MOR, LEM, and LEX). The test set was used for evaluating the finally selected models once.

The evaluation metrics used are *labeled attachment score* (LAS) – the percentage of tokens that are assigned the correct head and dependency type – and *unlabeled attachment score* (UAS) – the percentage of tokens that are assigned the correct head (regardless of dependency type). In addition, we present precision and recall for non-projective dependencies. Punctuation tokens are excluded in all scores, but phantom tokens are included. We use McNemar’s test for statistical significance.

4.2 Results and Discussion

Table 2 gives the parsing accuracy for different feature models on the held-out test set, measured as labeled attachment score (LAS) and unlabeled attachment score (UAS). With respect to LAS, there are statistically significant differences between all models except BM1 and All ($p < 0.01$). With respect to UAS, there are statistically significant differences between four groups, such that $\{\text{Base}\} < \{\text{B1, B2, B3}\} < \{\text{BM2}\} < \{\text{BM1, All}\}$, but there

¹Since SYNTAGRUS only distinguishes ten different parts of speech (not counting morphological features), the fields CPOSTAG and POSTAG in the CoNLL format – for coarse-grained and fine-grained parts of speech – were given the same content.

are no differences within these groups.²

Looking at the results for different models, we see that while the baseline model (Base) achieves a modest 60.2% LAS and 76.0% UAS, the addition of only one additional feature group (B1–B3) boosts unlabeled accuracy by close to ten percentage points and labeled accuracy by up to fifteen percentage points. Somewhat surprisingly, the differences between models B1–B3 are very small, and only differences with respect to LAS are statistically significant, which may be taken to suggest that morphological and lexical features capture the same type of information. However, this hypothesis is clearly refuted by the results for models BM1 and BM2, where the addition of lexical features on top of morphological features gives a further gain in LAS of eight to ten percentage points (and over four percentage points for UAS).

Comparing the use of raw word forms (LEX) and lemmas (LEM) as lexical features, we see a slight advantage for the latter, at least for labeled accuracy. However, it must be remembered that the experiments are based on gold standard input annotation, which probably leads to an overestimation of the value of LEM features. Finally, it is worth noting that including both LEX and LEM features does not result in a significant improvement over the model with only LEM features, which may be a sign of saturation, although this may again change in the presence of noisy LEM features.

The experimental results show conclusively that both morphological and lexical features are crucial for achieving high parsing accuracy. It may seem that they are most important for labeled accuracy, where the gain in absolute percentage points is the greatest with respect to the baseline, but it must be remembered that the unlabeled scores start at a higher level, thus leaving less room for improvement. In fact, the total error reduction from Base to All is over 50% for both LAS and UAS.

Table 3 gives a more detailed picture of parsing performance for the best model (All), by breaking down both LAS and UAS by the part-of-speech tag of the dependent. We note that accuracy is higher than average for nouns (S), adjectives (A), particles (PART), and reasonably good for verbs (V). For prepositions (PR), conjunctions (CONJ), and adverbs (ADV), accuracy is considerably lower, which may be attributed to attachment ambiguities.

²For the difference $BM2 < BM1$, $0.01 < p < 0.05$; for all other differences, $p < 0.01$.

Part of Speech	Count	LAS	UAS
S (noun)	7303	86.7	93.3
A (adjective)	7024	92.8	94.2
V (verb)	6946	81.9	85.8
PR (preposition)	5302	60.0	79.0
CONJ (conjunction)	2998	76.1	80.7
ADV (adverb)	2855	72.3	83.3
PART (particle)	1833	88.1	89.6
NUM (numeral)	807	88.7	93.6
NID (foreign word)	142	76.5	91.5
COM (compound)	32	93.8	96.9
P (proposition word)	7	57.1	85.7
INTJ (interjection)	5	0.0	20.0

Table 3: Accuracy by part of speech on the final test set for All features (Count = Number of tokens in the test set, LAS = Labeled attachment score, UAS = Unlabeled attachment score).

ties. It is also worth noting that both prepositions and adverbs have considerably higher UAS than LAS (almost twenty percentage points for prepositions), which shows that even when they are attached correctly they are often mislabeled. The remaining parts of speech are too infrequent to warrant any conclusions.

Looking specifically at non-projective dependencies, we find that the best model has a labeled precision of 68.8 and a labeled recall of 31.4. The corresponding unlabeled figures are 73.3 and 33.4.³ This confirms the results of previous studies showing that the pseudo-projective parsing technique used by MaltParser tends to give high precision – given that non-projective dependencies are among the most difficult to parse correctly – but rather low recall (McDonald and Nivre, 2007). It is also worth mentioning that phantom tokens, i.e., empty tokens inserted for the analysis of certain elliptical constructions (see section 2), have a labeled precision of 82.4 and a labeled recall of 82.8 (89.2 and 89.6 unlabeled), which is very close to the average accuracy, despite being very infrequent. However, it must be remembered that these tokens were given as part of the input in these experiments. In order to correctly analyse these tokens and their dependencies when parsing raw text, they would have to be recovered in a pre-processing phase along the lines of Dienes

³The precision is the percentage of non-projective dependencies predicted by the parser that were correct, while the recall is the percentage of true non-projective dependencies that were correctly predicted by the parser.

and Dubey (2003).

Summing up, the main result of the experimental evaluation is that both morphological and lexical features are crucial for attaining high accuracy when training and evaluating on the representations found in the SYNTAGRUS treebank of Russian. With regard to morphological features this is in line with a number of recent studies showing the importance of morphology for parsing languages with less rigid word order, including work on Spanish (Cowan and Collins, 2005), Hebrew (Tsarfaty, 2006; Tsarfaty and Sima'an, 2007), Turkish (Eryigit et al., 2006), and Swedish (Øvreliid and Nivre, 2007).

With regard to lexical features, the situation is more complex in that there are a number of studies questioning the usefulness of lexical features in statistical parsing and arguing that equivalent or better results can be achieved with unlexicalized models provided that linguistic categories can be split flexibly into more fine-grained categories, either using hand-crafted splits, as in the seminal work of Klein and Manning (2003), or using hidden variables and unsupervised learning, as in the more recent work by Petrov et al. (2006), among others. There are even studies showing that lexicalization can be harmful when parsing richly inflected languages like German (Dubey and Keller, 2003) and Turkish (Eryigit and Oflazer, 2006).

However, it is worth noting that most of these results have been obtained either for models of constituency-based parsing or for models of dependency parsing suffering from sparse data.⁴ In the experiments presented here, we have used a transition-based model for dependency parsing that has much fewer parameters than state-of-the-art probabilistic models for constituency parsing. Moreover, we have been able to use a relatively large training set, thereby minimizing the effect of sparseness for lexical features. We therefore conjecture that the beneficial effect of lexical features on parsing accuracy will generalize to other richly inflected languages when similar conditions hold.

As far as we know, these are the first results for a large-scale data-driven parser for Russian. There do exist several rule-based parsers for Russian, such as the ETAP-3 parser (Apresian et al., 2003) and a Link Grammar parser,⁵ as well as a prototype of a hybrid system based on the ETAP-3 parser en-

⁴The latter case applies to the probabilistic model of dependency parsing explored by Eryigit and Oflazer (2006).

⁵<http://sz.ru/parser/>

riched with statistics extracted from SYNTAGRUS (Boguslavsky et al., 2003; Chardin, 2004), but differences in both input format and output representations make it difficult to compare the performance directly.

5 Conclusion

We have presented the first results on parsing the SYNTAGRUS treebank of Russian using a data-driven dependency parser. Besides establishing a first benchmark for the SYNTAGRUS treebank, we have analyzed the influence of different kinds of features on parsing accuracy, showing conclusively that both lexical and morphological features are crucial for obtaining good parsing accuracy. We hypothesize that this result can be generalized to other richly inflected languages, provided that sufficient amounts of data are available.

Future work includes a deeper analysis of the influence of individual features, both morphological and lexical, as well as an evaluation of the parser under more realistic conditions without gold standard annotation in the input. This will require not only automatic morphological analysis and disambiguation but also a mechanism for inserting so-called phantom tokens in elliptical constructions.

Acknowledgments

We want to thank Ivan Chardin for initiating this collaboration and Jens Nilsson for converting the SYNTAGRUS data to the CoNLL format. We are grateful to the Russian Foundation of Basic Research for partial support of this research (grant no. 07-06-00339).

References

- Apresian, Ju., I. Boguslavsky, L. Iomdin, A. Lazursky, V. Sannikov, V. Sizov, and L. Tsinman. 2003. ETAP-3 linguistic processor: A full-fledged NLP implementation of the MTT. In *Proceedings of the First International Conference on Meaning-Text Theory*, 279–288.
- Black, E., F. Jelinek, J. D. Lafferty, D. M. Magerman, R. L. Mercer, and S. Roukos. 1992. Towards history-based grammars: Using richer models for probabilistic parsing. In *Proceedings of the 5th DARPA Speech and Natural Language Workshop*, 31–37.
- Boguslavsky, I., S. Grigorieva, N. Grigoriev, L. Kreidlina, and N. Frid. 2000. Dependency treebank for Russian: Concept, tools, types of information. In *Proceedings of COLING*, 987–991.

- Boguslavsky, I., I. Chardin, S. Grigorieva, N. Grigoriev, L. Iomdin, L. Kreidlin, and N. Frid. 2002. Development of a dependency treebank for Russian and its possible applications in NLP. In *Proceedings of LREC*, page 852856.
- Boguslavsky, I. M., L. L. Iomdin, V. S. Sizov, and I. Chardin. 2003. Parsing with a treebank. In *Proceedings of the Conference on Cognitive Modeling in Linguistics* [In Russian].
- Buchholz, S. and E. Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL*, 149–164.
- Chang, C.-C. and C.-J. Lin, 2001. *LIBSVM: A Library for Support Vector Machines*. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Chardin, Ivan. 2004. *Dependency Treebanks and Their Use in Parsing*. Ph.D. thesis, Russian Academy of Science [In Russian].
- Cowan, B. and M. Collins. 2005. Morphology and reranking for the statistical parsing of spanish. In *Proceedings of HLT/EMNLP*, 795–802.
- Dienes, P. and A. Dubey. 2003. Deep syntactic processing by combining shallow methods. In *Proceedings of ACL*, 431–438.
- Dubey, A. and F. Keller. 2003. Probabilistic parsing for German using sister-head dependencies. In *Proceedings of ACL*, 96–103.
- Eryiğit, G. and K. Oflazer. 2006. Statistical dependency parsing of Turkish. In *Proceedings of EACL*, 89–96.
- Eryigit, G., J. Nivre, and K. Oflazer. 2006. The incremental use of morphological information and lexicalization in data-driven dependency parsing. In *Proceedings of the 21st International Conference on the Computer Processing of Oriental Languages*, 498–507.
- Hajič, J., B. Vidova Hladka, J. Panevová, E. Hajičová, P. Sgall, and P. Pajas. 2001. Prague Dependency Treebank 1.0. LDC, 2001T10.
- Klein, D. and C. D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL*, 423–430.
- Kudo, T. and Y. Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of CoNLL*, 63–69.
- Magerman, D. M. 1995. Statistical decision-tree models for parsing. In *Proceedings of ACL*, 276–283.
- McDonald, R. and J. Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of EMNLP-CoNLL*, 122–131.
- Mel'čuk, I. 1988. *Dependency Syntax: Theory and Practice*. State University of New York Press.
- Nivre, J. and J. Nilsson. 2005. Pseudo-projective dependency parsing. In *Proceedings of ACL*, 99–106.
- Nivre, J., J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. 2007a. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007*, 915–932.
- Nivre, J., J. Hall, J. Nilsson, A. Chanev, G. Eryiğit, S. Kübler, S. Marinov, and E. Marsi. 2007b. Malt-Parser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13:95–135.
- Nivre, J. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of IWPT*, 149–160.
- Øvrelid, L. and J. Nivre. 2007. When word order and part-of-speech tags are not enough – swedish dependency parsing with rich linguistic features. In *Proceedings of RANLP*, 447–451.
- Petrov, S., L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of COLING/ACL*, 433–440.
- Ratnaparkhi, A. 1997. A linear observed time statistical parser based on maximum entropy models. In *Proceedings of EMNLP*, 1–10.
- Tesnière, L. 1959. *Éléments de syntaxe structurale*. Editions Klincksieck.
- Tsarfaty, R. and K. Sima'an. 2007. Three-dimensional parametrization for parsing morphologically rich languages. In *Proceedings of IWPT*, 156–167.
- Tsarfaty, R. 2006. Integrated morphological and syntactic disambiguation for modern hebrew. In *Proceedings of the COLING/ACL 2006 Student Research Workshop*, 49–54.
- Vapnik, V. N. 1995. *The Nature of Statistical Learning Theory*. Springer.
- Yamada, H. and Y. Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT*, 195–206.