NEALT PROCEEDINGS SERIES

VOL. 9

# Proceedings of the Ninth International Workshop on Treebanks and Linguistic Theories

3–4 December 2010

University of Tartu, Estonia

*Editors:*
Markus Dickinson
Kaili Müürisep
Marco Passarotti

NORTHERN EUROPEAN ASSOCIATION FOR LANGUAGE TECHNOLOGY

Proceedings of the Ninth International Workshop on Treebanks
and Linguistic Theories
NEALT Proceedings Series, Vol. 9

ISSN 1736-6305 (Online)
ISSN 1736-8197 (Print)

Printed in Tartu (Estonia)
*Published by*
Northern European Association for Language Technology (NEALT)
`http://omilia.uio.no/nealt/`

*Electronically published at*
Tartu University Library (Estonia)
`http://hdl.handle.net/10062/15891`

*Volume Editors:*
Markus Dickinson
Kaili Müürisep
Marco Passarotti

The publication of the proceedings is supported by the European Regional Development Fund through the Estonian Centre of Excellence in Computer Science (EXCS).

Euroopa Liit
Euroopa
Regionaalarengu Fond

Eesti tuleviku heaks

# Preface

The Ninth International Workshop on Treebanks and Linguistic Theories (TLT9) was held at the University of Tartu in Tartu, Estonia on 3-4 December 2010 (see `http://math.ut.ee/tlt9/`). This marked the first time that it was held in one of the Baltic states. Dates and locations of the previous workshops are provided in a separate section.

Since its first edition in 2002, TLT has served as an ideal venue for new and ongoing high-quality work related to syntactically-annotated corpora, i.e., treebanks. Treebanks have become crucially important for the development of data-driven approaches to natural language processing, human language technologies, grammar extraction and linguistic research in general. Additionally, there are projects that explore annotation beyond syntactic structure (including, for instance, semantic, pragmatic and rhetorical annotation), beyond a single language (for instance, parallel treebanks), and beyond simply written data, incorporating properties of speech. The papers for TLT over the years since 2002 have done much to capture this range of work, encompassing descriptive, theoretical, formal and computational aspects of treebanks.

Experiences in building syntactically-processed corpora have shown that there is a relation between linguistic theory and the practice of syntactic annotation. Since the practices of building syntactically-annotated corpora have illustrated that aiming at a more detailed description of the data becomes more and more theory-dependent, the connections between treebank development and linguistic theories need to be tightly connected in order to ensure the necessary flow of information between them.

The call for papers for TLT9 requested unpublished, completed work. 35 submissions were received, authored by 95 different authors, illustrating the extremely collaborative nature of these works. The submissions were authored by researchers from 16 different countries in America, Asia and Europe, and each submission was evaluated by three reviewers.

The Programme Committee consisted of 19 members (including the 3 co-chairs) from 10 different countries, all working as reviewers. Based on their scores and the comments they provided on the content and quality of the papers, 15 papers and 6 posters were accepted for presentation and publication. This corresponds to an acceptance rate of about 60%. The accepted submissions cover a wide range of topics related to both long-standing and new treebanks, reporting on aspects of their construction, querying, exploitation and evaluation.

Completing the programme are the invited lectures by Anke Lüdeling on "Syntactic Misuse, Overuse and Underuse: A Study of a Parsed Learner Corpus and its Target Hypothesis" and by Joakim Nivre on "Harvest Time – Explorations of the Swedish Treebank". Both of these invited talks connect years of treebanking with exciting new developments in learner corpora and the process of building larger treebanks.

Following in the tradition of TLT's recent editions, a co-located event was also organised (see `http://math.ut.ee/tlt9/aepc/`). This one-day event, preceding TLT9, was a Workshop on the Annotation and Exploitation of Parallel Corpora (AEPC). This co-located event arose from the consideration that parallel corpora are extremely useful data sets for which there is an increasing need of adding linguistic annotation. Syntax-enhanced approaches to machine translation rely on a strong connection to treebanks, and thus participants in both workshops benefitted from the unique combination of the workshops.

We wish to express our gratitude to the members of the programme committee, who worked hard to review all submissions under hard time constraints. We also want to thank University of Tartu, Institute of Computer Science and Estonian Center of Excellence in Computer Science for sponsorship and support to the workshop.

The publication of these proceedings was supported by the European Regional Development Fund through the Estonian Center of Excellence in Computer Science, EXCS.

The TLT9 Co-Chairs

Markus Dickinson; Indiana University; Bloomington, IN, USA

Kaili Müürisep; University of Tartu; Tartu, Estonia

Marco Passarotti; Catholic University of the Sacred Heart; Milan, Italy

# Programme Committee

**Programme Committee Chairs**

Markus Dickinson, University of Indiana, USA

Kaili Müürisep, University of Tartu, Estonia

Marco Passarotti, Catholic University of the Sacred Heart, Milan, Italy

**Programme Committee Members**

David Bamman, USA

Eckhard Bick, Denmark

Ann Bies, USA

Dan Flickinger, USA

Anette Frank, Germany

Eva Hajičová, Czech Republic

Dag Haug, Norway

Erhard Hinrichs, Germany

Sandra Kübler, USA

Jonas Kuhn, Germany

Anna Kupść, France

Anke Lüdeling, Germany

Simonetta Montemagni, Italy

Petya Osenova, Bulgaria

Kiril Simov, Bulgaria

Martin Volk, Switzerland

# Organizing Committee

University of Tartu, Estonia:

- Kadri Muischnek
- Mare Koit
- Krista Liin
- Kaili Müürisep
- Urve Talvik

# Proceedings of previous TLT workshops and invited speakers

E. Hinrichs & K. Simov (eds.), Proceedings of the First Workshop on Treebanks and Linguistic Theories, Sozopol (Bulgaria), September 20- 21, 2002. v + 274 pages. `http://www.bultreebank.org/Proceedings.html`.

J. Nivre & E. Hinrichs (eds.), Proceedings of the Second Workshop on Treebanks and Linguistic Theories, Växjö (Sweden), November 14-15, 2003. Växjö University Press. 232 pages.
*Invited speakers:* Thorsten Brants (Google Inc.), Stephan Oepen (U. of Oslo).

S. Kübler, J. Nivre, E. Hinrichs & H. Wunsch (eds.), Proceedings of the Third Workshop on Treebanks and Linguistic Theories, Tübingen (Germany), December 10-11, 2004. Eberhard Karls Universität Tübingen, Seminar für Sprachwissenschaft. vi + 203 pages.
Publication of selected papers in: E. Hinrichs & K. Simov (eds.), Treebanks and Linguistic Theories. Special Issue of the Journal on Research on Language and Computation. Vol. 2, Nr. 4, 2004.
*Invited speakers:* Collin Baker (ICSI, Berkeley), Fred Karlsson (U. of Helsinki).

M. Civit, S. Kübler & M.A. Martí (eds.), Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories, Barcelona (Spain), December 9-10, 2005. Universitat de Barcelona, Publicacions i Edicions. 220 pages.
*Invited speakers:* Frank van Eynde (U. Leuven), Manfred Pinkal (U. of the Saarland).

J. Hajič & J. Nivre (eds.), Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories, Prague (Czech Republic), December 1-2, 2006. Univerzita Karlova v Praze, Ústav Formální Aplikované Lingvistiky. 258 pages.
*Invited speakers:* Gosse Bouma (U. of Groningen), Martha Palmer (U. of Colorado at Boulder).

K. De Smedt, J. Hajič & S. Kübler (eds.), Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories, Bergen (Norway), December 7-8, 2007. Northern European Association for Language Technology Proceedings Series, Vol. 1. viii + 218 pages. `http://dspace.utlib.ee/dspace/handle/10062/4476`.
*Invited speakers:* Erhard Hinrichs (U. of Tübingen), Julia Hockenmaier (U. of Illinois).

F. van Eynde, A. Frank, K. De Smedt & G. van Noord (eds.), Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories,

Groningen (The Netherlands), January 23-24, 2009. Landelijke Onderzoekschool Taalwetenschap, Occasional Series. 197 pages.
*Invited speakers:* Robert Malouf (San Diego State U.), Adam Przepiórkowski (Polish Academy of Sciences).

M. Passarotti, A. Przepiórkowski, S. Raynaud & F. Van Eynde (eds.), Proceedings of the Eighth International Workshop on Treebanks and Linguistic Theories, Catholic University of the Sacred Heart, Milan, Italy, December 4-5, 2009. EDUCatt, Ente per il Diritto allo Studio Universitario dell'Università Cattolica. xii+220 pages.
*Invited speakers:* Roberto Busa SJ (Catholic University of the Sacred Heart, Milan), Eva Hajičová (Charles University, Prague).

# Contents

# Syntactic Misuse, Overuse and Underuse: A Study of a Parsed Learner Corpus and its Target Hypothesis

Anke Lüdeling, Amir Zeldes, Marc Reznicek,
Ines Rehbein, Hagen Hirschmann

Humboldt-Universität zu Berlin

This talk is concerned with using syntactic annotation of learner language and the corresponding target hypothesis to find structural acquisition difficulties in German as a foreign language. Using learner data for the study of acquisition patterns is based on the idea that learners do not produce random output but rather possess a consistent internal grammar (interlanguage; cf. [1] and many others). Analysing learner data is thus an indirect way of assessing the interlanguage of language learners. There are two main ways of looking at learner data, error analysis and contrastive interlanguage analysis [2, 3]. A careful analysis of errors makes it possible to understand learners' hypotheses about a given grammatical phenomenon. Contrastive interlanguage analysis is not concentrated on errors but compares categories (of any kind) of learner language with the same categories in native speaker language. Learners' underuse of a category (i.e. a significantly lower frequency in learner language than in native speaker language) can be seen as evidence for the perceived difficulty of that category (either because learners fail to acquire it, or because they deliberately avoid it).

While some learner corpora are annotated (manually or automatically) with part-of-speech or lemma information [4], or even error types, there are as yet only very few attempts to annotate them syntactically (some exceptions are [5] or [6]. Parsing learner data is very difficult because of the learner errors but would be very helpful for the analysis of errors and overuse/underuse of syntactic structures and categories. In our paper we therefore discuss how the comparison of parsed learner data and the corresponding target hypotheses helps in understanding syntactic properties of learner language.

We use the Falko corpus which contains essays of advanced learners of German as a foreign language and control essays by German native speakers [7]; the corpus is freely available[1]. Since it is very difficult to decide what an error is and often there can be different hypotheses about the 'correct' structure the learner utterance

[1] http://www.linguistik.hu-berlin.de/institut/professuren/korpuslinguistik/forschung-en/falko/standardseite-en

is evaluated against [8] both subcorpora are annotated manually with several layers of target hypotheses, as well as automatically with part-of-speech, lemma, and edit error tags [9].

The original learner data and the target hypotheses were parsed with a state-of-the-art statistical parser trained on the TiGer treebank [10]. Since the target hypotheses are aligned with the original data we can identify those sections in the data where parsing of the original fails but parsing of the target hypothesis is possible. We can then see which syntactic structures are assigned to the target hypothesis and use this as a diagnostic for syntactic learner errors. We can also analyse the syntactic categories in the learner data quantitatively against the native speaker data.

## References

[1] Larry Selinker. Interlanguage. *IRAL*, 10/3:31–54, 1972.

[2] Sylviane Granger. From CA to CIA and back. An integrated approach to computerized bilingual and learner corpora. In Karin Aijmer, editor, *Papers from a Symposium on Text-based Cross-linguistic Studies Lund 4 - 5 March 1994*, page 37–51. Lund University Press,, 1996.

[3] Sylviane Granger. Learner corpora. In Anke Lüdeling and Merja Kytö, editors, *Corpus Linguistics. An International Handbook*, pages 259–275. Mouton de Gruyter, Berlin, 2008.

[4] Ana Díaz-Negrillo, Detmar Meurers, Salvador Valera, and Holger Wunsch. Towards interlanguage POS annotation for effective learner corpora in SLA and FLT. *Language Forum*, 36(1–2), 2010.

[5] Markus Dickinson and Marwa Ragheb. Dependency Annotation for Learner Corpora. In *Proceedings of the Eighth Workshop on Treebanks and Linguistic Theories (TLT-8)*, 2009.

[6] Niels Ott and Ramon Ziai. Evaluating Dependency Parsing Performance on German Learner Language. In *Proceedings of the Ninth Workshop on Treebanks and Linguistic Theories (TLT-9)*, Tartu, 2010.

[7] Anke Lüdeling, Seanna Doolittle, Hagen Hirschmann, Karin Schmidt, and Maik Walter. Das Lernerkorpus Falko. *Deutsch als Fremdsprache*, 2:67–73, 2008.

[8] Anke Lüdeling. Mehrdeutigkeiten und Kategorisierung: Probleme bei der Annotation von Lernerkorpora. In Maik Walter and Patrick Grommes, editors, *Fortgeschrittene Lernervarietäten*, pages 119–140. Niemeyer, Tübingen, 2008.

[9] Marc Reznicek, Maik Walter, Karin Schmid, Anke Lüdeling, Hagen Hirschmann, and Cedric Krummes. *Das Falko-Handbuch. Korpusaufbau und Annotationen Version 1.0*, 2010.

[10] Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. TIGER: Linguistic Interpretation of a German Corpus. *Research on Language & Computation*, 2:597–620, 2004.

# Harvest Time – Explorations of the Swedish Treebank

Joakim Nivre

Uppsala University

Work on building a large treebank for Swedish started at about the same time as the TLT workshop series and reached a significant milestone this year with the second release of the Swedish Treebank, a corpus developed by merging and harmonizing the existing corpora Talbanken and the Stockholm-Umeå Corpus. In this talk, I will first present the treebank itself, explaining how it was developed using cross-corpus harmonization and annotation projection and describing the final result, which is a multi-representation treebank including annotation of phrase structure, grammatical functions and dependency structure. I will go on to describe ongoing work at exploiting the treebank for parser development, using data-driven methods for dependency parsing, and I will end by discussing our plans to use the treebank for cross-framework parser evaluation, in particular for comparing constituency-based and dependency-based parsing methods.

# Conversion of a Russian dependency treebank into HPSG derivations

Tania Avgustinova and Yi Zhang

Language Technology Lab
DFKI GmbH

{avgustinova; yzhang} @ dfki.de

### Abstract

The Russian syntactic treebank SynTagRus is annotated with dependency structures in line with the Meaning-Text Theory (MTT). In order to benefit from the detailed syntactic annotation in SynTagRus and facilitate the development of a Russian Resource Grammar (RRG) in the framework of Head-driven Phrase Structure Grammar (HPSG), we need to convert the dependency structures into HPSG derivation trees. Our pilot study has shown that many of the constructions can be converted systematically with simple rules. In order to extend the depth and coverage of this conversion, we need to implement conversion heuristics that produce linguistically sound HPSG derivations. As a result we obtain a structured set of correspondences between MTT surface syntactic relations and HPSG phrasal types, which enable the cross-theoretical transfer of insightful syntactic analyses and formalized deep linguistic knowledge. The converted treebank SynTagRus++ is annotated with HPSG structures and of crucial importance to the RRG under development, as our goal is to ensure an optimal and efficient grammar engineering cycle through dynamic coupling of the treebank and the grammar.

## 1 Introduction

Key issues brought up recently in the research and development community concern the application of treebanks in acquiring linguistic knowledge for natural language processing, the role of linguistic theories in treebank development, and the suitability of treebanks as a basis for linguistic research.[*] In this context, we discuss the conversion of a Russian dependency treebank into Head-driven Phrase Structure Grammar (HPSG) derivations needed in the context of the Russian Resource Grammar (RRG) under development in our group ([3], [4]). We shall, therefore, focus on the problems of annotation transfer revealing possibilities for conceptual alignment of the underlying linguistic theories. Other aspects that will be

[*] We are grateful to Leonid L. Iomdin for providing us with access to the SynTagRus dependency treebank and for helpful answers to annotation-related questions.

touched upon are related to the use of a bootstrapping approach towards an incremental treebank conversion process.

The Russian treebank SynTagRus – cf. [6], [7], [2] – contains a genuine dependency annotation theoretically grounded in the long tradition of dependency grammar represented by the work of Tesnière [15] and Mel'čuk [10] among others. In particular, a complete dependency tree is provided for every sentence in the corpus. Supplied with comprehensive linguistic annotation, this treebank has already served as a basis for experimental investigations using data-driven methods [13]. By way of background, we start by introducing the Meaning-Text Theory (MTT) tradition of dependency grammar as reflected in the syntactic annotation of the Russian treebank SynTagRus and obtained with the ETAP-3 linguistic processor [1]. The main part of the paper is then devoted to the step-by-step "on-demand" conversion of the original dependency representation into an HPSG-conform phrase structure format. Finally, we discuss some non-trivial cross-theoretical issues and consider possibilities for phenomena-oriented re-structuring of the inventory of surface syntactic relations to enable a linguistically informed treebank transformation.

# 2 Background

The MTT-based annotation of the SynTagRus treebank provides various types of linguistic information. In particular, the morphological features associated with individual lexical items include the respective part of speech, and depending on it, further features like animacy, gender, number, case, degree of comparison, short form (for adjectives and participles), representation (of verbs), aspect, tense, person, and voice. In SynTagRus, sentences are represented as trees in which words are nodes and edges between them are marked with the appropriate syntactic relation. The number of nodes in the tree structure typically corresponds to the number of word tokens, and the dependencies between them are binary and oriented, i.e. linking single words rather than syntactic groups. For every syntactic group, one word (*head*) is chosen to represent it as a dependent in larger syntactic units; all other members of the group become dependents of the head word. Punctuation marks do not carry any labeling and are not included in syntactic trees.

The rich inventory of MTT surface syntactic relations – about sixty, as currently annotated in the treebank – captures fine-grained language-specific grammatical functions of the lexemes in a sentence and is traditionally divided into six major groups – i.e. actantial, attributive, quantitative, adverbial, coordinative, or auxiliary – which, in fact already provides a generic picture of abstract dependency relations and guidelines for our cross-theoretical investigation.

I. Actantial relations link a predicate word to its arguments. Prototypical instances thereof are: *predicative*, *completive, prepositional*

II. Attributive relations often link a noun to a modifier expressed by an adjective, another noun or a participle clause. Prototypical instances thereof are: *attributive, modificational, relative*

III. Quantitative relations link a noun to a quantifier or numeral, or two such words together. A prototypical instances thereof is: *quantitative*

IV. Adverbial relations link a predicate word to various adverbial modifiers. Prototypical instances thereof are: *circumstantial, parenthetic*

V. Coordinative relations serve phrases and clauses coordinated by conjunctions. Prototypical instances thereof are: *coordinative*, *coordinative-conjunctive*

VI. Auxiliary relations typically link two elements that form a single syntactic unit (e.g. an analytical verb form). Prototypical instances thereof are: *auxiliary*, *analytical*

As SynTagRus authors point out, the language-specific inventory of surface syntactic relations is not closed, as the process of data acquisition brings up rare syntactic constructions not covered by traditional grammars, which requires new syntactic link types to be introduced for make the respective syntactic structure unambiguous. Let us consider an example of the original SynTagRus annotation.

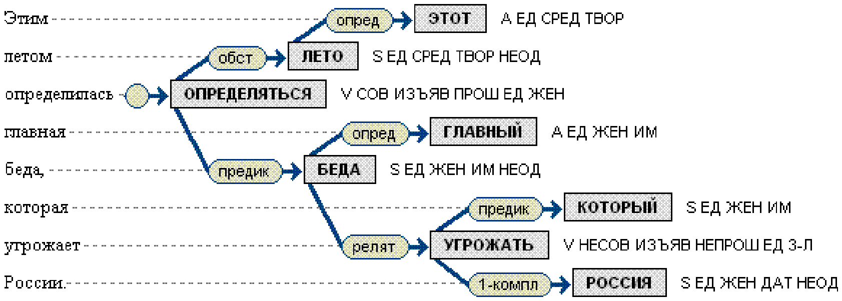


Figure 1: Original SynTagRus annotation

The sentence in Figure 1 may be indicatively translated as: “This summer took shape the main adversity that threatens Russia.” The matrix verb определилась (took shape) is in a *predicative* (предик) dependency with its subject беда (distress) and in a *circumstantial* (обст) dependency with the temporal adverbial летом (summer). The former is in a *modificational* (опред) dependency with the attributive adjective главная (main) and in a *relative* (релят) dependency with the verb of the relative clause угрожает (threatens). The latter, on the other hand, is in a modificational (опред) dependency with the demonstrative pronominal adjective этим (this). The embedded verb, in turn, is in a predicative (предик) dependency with the relative pronoun которая (which) and in a 1-completive (1-комл) dependency with its object России (Russia).

# 3 Treebank conversion

The conversion of the SynTagRus dependency treebank to the HPSG derivations is achieved in the following three steps. First, the dependency trees are converted into pseudo phrase structure trees by creating constituents for head words and their dependents. As the majority of the dependencies are projective, the conversion results in mostly continuous constituents. The non-continuous constituents produced from the non-projective dependencies are also preserved at this point, and will be handled in the later conversion stages. We use the Negra/Tiger XML format [11] to record the syntactic structures throughout the conversion. The format conveniently supports non-continuous constituents. The dependency relation types are also preserved in the resulting constituent tree as edge labels: the head word is governed by its upper constituent with the "HD" edge, while all its immediate dependents are governed by the upper constituent with an edge named after the corresponding dependency relation. Figure 2 shows the pseudo phrase structure tree of the example sentence from the previous section (cf. Figure 1). The constituents SP, and VP are created automatically, and named according the part of speech of the head word (i.e. "substantive" and "verb", respectively). Different bar levels of the constituents are not yet determined, and the tree structure can be rather "flat".

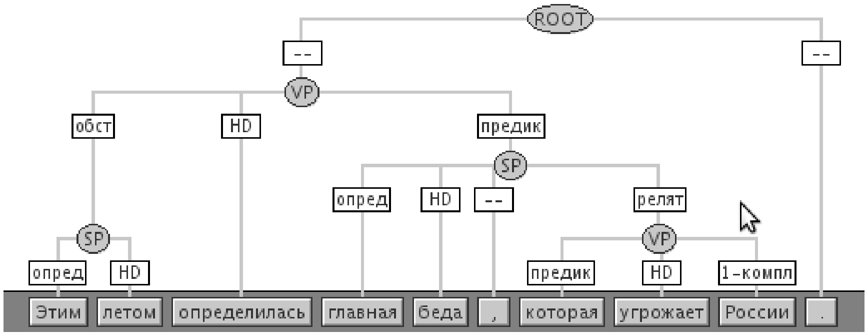


Figure 2: Converted SynTagRus format

The next step of the conversion aims to annotate the branches in the pseudo phrase structure tree with HPSG-oriented schemata. In the initial phase of the treebank conversion we work with a small set of HPSG-oriented schemata for headed phrases (cf. Table 1) which have straightforward structure-preserving correspondences in terms of MTT *surface syntactic relations*.

It is worth noting that during this conversion a language specific theory evolves. Starting from the standard HPSG inventory of schemata we eventually arrive at more fine-grained inventory modeling language specific phenomena. The resulting theory would be still HPSG inspired but also draw insight form the MTT approach.

Table 1: Initial basic inventory of HPSG phrasal schemata

| | | |
|---|---|---|
| **<01>** | HD+SBJ | *predicative* |
| **<02>** | HD+CMP | *1/2/3-completive* (with non-nominal head)*; agentive; prepositional* |
| **<03>** | HD+CMP/PRD | *copulative* |
| **<04>** | HD+CMP/ADJ | *quasi-agentive; 1-completive* (with nominal head); *elective; comparative* |
| **<05>** | HD+ADJ | *attributive, circumstantial, delimitative, relative modificational* |
| **<06>** | HD+ADJ/CPD | *compound* |
| **<07>** | HD+SPR | *quantitavie* |
| **<08>** | HD+AUX | *auxiliary* |
| **<09>** | HD+PARENTH | *parenthetical* |

Schema <01> covers the *predicative* (предик) dependency holding between the verb and its subject. Schema <02> covers all *completive* (компл) dependencies of non-nominal heads as well as the *agentive* (агент) dependency introducing the “demoted” instrumental agent in passivization or nominalization constructions (i.e. equivalent to “by-phrase”), and the *prepositional* dependency between a preposition and the noun. Schema <03> covers the *copulative* (присвяз) dependency holding between a copula verb and the predicative. Schema <04> is underspecified with regard to complement or adjunct status and – with nominal heads only – covers the *completive* (компл) dependencies and the *quasi-agentive* (квазиагент) dependency to a genitive noun (i.e. equivalent to “of-phrase”), as well as the *comparative* (сравнит) dependency between a head and an indicated object of comparison and the *elective* (электив) dependency between a head and a respectively indicated set. Schema <05> covers various kinds of adjuncts corresponding to the *modificational* (опред) dependency between a noun and its agreeing (i.e. adjectival) attribute, the *attributive* (атриб) dependency between a noun and its non-agreeing (i.e. non-adjectival) attribute, the *circumstantial* (обст) dependency of a head to its adverbial modification, the *delimitative* (огранич) dependency of a particle or a quantifying adverb to the head it restricts, the *relative* (релят) dependency between the head noun and the relative clause modifying it. Schema <06> corresponds to the *compound* (композ) dependency between a head and a modifier part of a compound. Schema <07> corresponds to the *quantitative* (количест) dependency to a numeral expression. Schema <08> covers the *auxiliary* (вспом) dependency between a head and various auxiliary elements. Finally, schema <09> covers the *parenthetical* (вводн) dependency between a head and an inserted parenthetical expression which is usually divided by punctuation marks.

These schemata cover an essential part of the phenomena in the HPSG view. While some of the schemata correspond clearly with some dependency relations in a one-to-one fashion, others are not as straightforward. This reflects the asymmetry of different linguistic frameworks, and presents a major difficulty in developing conversion programs. Previous attempts in this direction usually involve the design of complex rule-based conversion heuristics (cf. [12], [9], [8]). In practice, these heuristics are also highly dependent on the annotation schema, and do not always carry linguistically interesting analyses.

In this work, we propose to use a different bootstrapping approach towards an incremental treebank conversion process. The process starts with linguists annotating instances of particular target structures, e.g. specific HPSG schemata like head-subject, head-complement, and head-adjunct. These annotations are attached to the original treebank annotation as already converted into pseudo phrase structure trees. A machine learning classifier will learn from these instances, and try to predict for the remainder of the treebank the conversion outcome. The conversion quality will be manually checked. Then the conversion results will be used as the starting point for the next (and potentially more difficult) conversion sub-step. Since for each round, we are only adding limited additional conversion decisions, annotation from a few dozen up to a few hundred instances will be enough for training the statistical classifiers.

Figure 3 shows the manual annotation of HPSG schemata on the pseudo phrase structure trees. Although the complete annotation is shown in this example, the annotators can choose to only visualize analyses they are interested in and annotate the instances they are sure about.

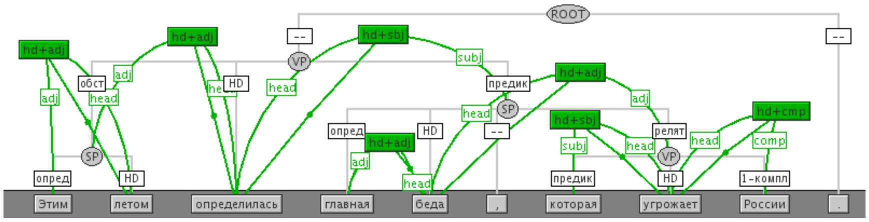


Figure 3: Manual HPSG-oriented meta-annotation

These annotations are then sent to train the statistical classifier, which is applied to disambiguate the mappings from dependency relations to the HPSG schemata. We use a maximum entropy-based classifier (TADM, http://tadm.sourceforge.net). The effective features for schemata classification include the part-of-speech of the head and daughter in the pseudo phrase structure tree, the dependency label, together with the sibling non-head daughters. The results are illustrated in Figure 4. While the edge labels now bear more resemblance to HPSG, the phrases structures are still flat.

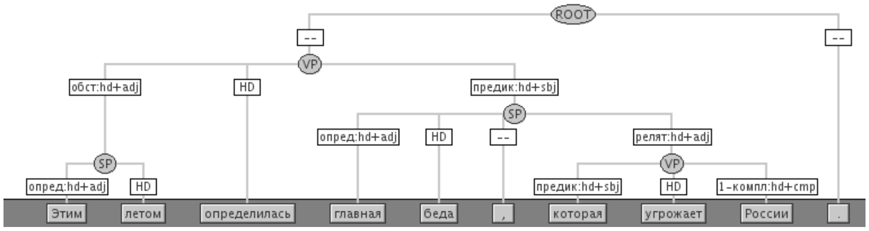


Figure 4: Automatic annotation with statistical classifier

Our experiments resulted in adequate automatic meta-annotation of the development corpus summarized in Table 2. In general, the assignment of core HPSG schemata, i.e. head-subject (with 172 occurrences), head-complement (with 354 occurrences), head adjunct (with 521 occurrences), and head-specifier (with 15 occurrences), is convincingly stable and highly conform to the initial setup of basic phrasal types outlined in Table 1. The same is true of the head-complement/adjunct schema (with 149 occurrences), which we introduced to account for the systematic functional status under-specification of a nominal head's dependents in *quasi-agentive* and *completive* surface syntactic relations.

Table 2: Experimental automatic meta-annotation results

| **Development corpus statistics** | | **Dependency** | **Schema** |
|---|---|---|---|
| **242** | опред:hd+adj | *modificational* | HD+ADJ |
| **174** | предл:hd+cmp | *prepositional* | HD+CMP |
| **172** | предик:hd+sbj | *predicative* | HD+SBJ |
| **145** | 1-компл:hd+cmp | *1-completive* | HD+CMP |
| **112** | обст:hd+adj | *circumstantial* | HD+ADJ |
| **105** | огранич:hd+adj | *delimitative* | HD+ADJ |
| **94** | квазиагент:hd+cmp/adj | *quasi-agentive* | HD+CMP/ADJ |
| **47** | 1-компл:hd+cmp/adj | *1-completive* | HD+CMP/ADJ |
| **38** | атриб:hd+adj | *attributive* | HD+ADJ |
| **28** | 2-компл:hd+cmp | *2-completive* | HD+CMP |
| **15** | количест:hd+spr | *quantitative* | HD+SPR |
| **15** | релят:hd+adj | *relative* | HD+ADJ |
| **13** | вводн:hd+parenth | *parenthetic* | HD+PARENTH |
| **8** | 2-компл:hd+cmp/adj | *2-completive* | HD+CMP/ADJ |
| **8** | сравнит:hd+adj | *comparative* | HD+ADJ |
| **6** | 3-компл:hd+cmp | *3-completive* | HD+CMP |
| **6** | присвяз:hd+cmp/prd | *copulative* | HD+CMP/PRD |
| **3** | вспом:hd+aux | *auxiliary* | HD+AUX |
| **2** | композ:hd+adj/cpd | *compound* | HD+ADJ/CPD |
| **1** | агент:hd+cmp | *agentive* | HD+CMP |
| **1** | электив:hd+cmp/adj | *elective* | HD+ADJ |

The assignment of other schemata, i.e. head-parenthetical (with 13 occurrences), head-predicative-complement (with 6 occurrences), head-auxiliary (with 3 occurrences), and head-adjunct-in-compound (with 2 occurrences), appears to give quite satisfactory results too. The *delimitative* (огранич) dependency, which involves heterogeneous non-head categories, has received in the experimental results an interpretation mainly as a head-adjunct structure (105 occurrences). Nevertheless, the theoretical question arises of whether to re-interpret this surface syntactic relation – at least in cases involving quantifying particles (negative, interrogative, topicalising, etc.) – as a head-marker structure. Also, a linguistically motivated interpretation of both *comparative* (сравнит) and *elective* (электив) surface syntactic relations would favor under-specification of the non-head component with regard to its complement or adjunct status, which corresponds to the head-complement/adjunct schema.

There are, in fact, a whole bunch of surface syntactic relations that have been intentionally excluded from the current experiment and, hence, got no meta-annotation in terms of HPSG schemata – cf. Table 3. For examples of individual dependency types refer to [7]. These are all, to a various degree, non-trivial cases, with the most representative group being the treatment of coordination phenomena.

Table 3: Dependencies currently excluded from meta-annotation

| **Development corpus statistics** | **Dependency** |
|---|---|
| **98** сочин | *coordinative* |
| **90** соч-союзн | *conjunctive-coordinative* |
| **30** подч-союзн | *conjunctive-subordinative* |
| **26** сент-соч | *sentential-coordinative* |
| **15** разъяснит | *expository* |
| **7** аппоз | *appositive* |
| **7** эксплет | *expletive* |
| **5** сравн-союзн | *conjunctive-comparative* |
| **4** примыкат | *adjunctive* |
| **3** 1-несобст-компл | *1-nonintrinsic-competive* |
| **3** 4-компл | *4-completive* |
| **3** аналит | *analytical* |
| **3** инф-союзн | *conjunctive-invinitival* |
| **3** кратн | *multiple* |
| **3** распред | *distributive* |
| **2** длительн | *durative* |
| **2** оп-опред | *descriptive-modificational* |
| **2** пролепт | *proleptic* |
| **2** соотнос | *correlational* |
| **1** 2-несобст-компл | *2-nonintrinsic-completive* |
| **1** компл-аппоз | *completive-appositive* |
| **1** ном-аппоз | *nominative-appositive* |
| **1** об-аппоз | *detached-appositive* |

Inasmuch as coordination relations are not dependencies in the strict sense of the word, their handling is always one way or another conventionalized in dependency grammar approaches. In SynTagRus, according to the Meaning Text Theory, the first coordination member is the head and is attached to the common parent, i.e. to the word governing the entire coordination. Each other coordination member (including conjunctions) is attached to the previous one, with the edges between coordination members being labeled with the *coordinative* (сочин) or the *conjunctive-coordinative* (соч-союзн) dependencies. With respect to common dependents, i.e. words depending on all coordination members, one particular solution has been favored in SynTagRus, namely, that these are attached to the nearest coordination member, often to the first one, with the other coordination members, including conjunctions, being attached to the respectively preceding one. The systematic source of ambiguity – whether a dependent of a coordination member actually refers to the whole coordination or only to that one member – is thus deliberately avoided in SynTagRus.

All the HPSG schemata we have are binary structures. This is because they are always more informative than flat structures involving more than two daughters. Also binary structure bears more resemblance to the dependency relations between pairs of words. For this reason, we need to further binarize the pseudo phrase structure trees. This turns out to be a non-trivial step for languages with relatively free word order. As there is less constraints over the linear precedence between constituents, it is hard to hard-wire schema priorities directly. Similar to the previous step, we start by annotating some of the binarization preferences by hand, and hope that the regularities will be then transferred to the remainder of the corpus. For example, in Figure 5, the left-most binarization annotation indicates that the verbal head will pick up the right-adjacent subject before combing with the modifying noun phrase to its left.

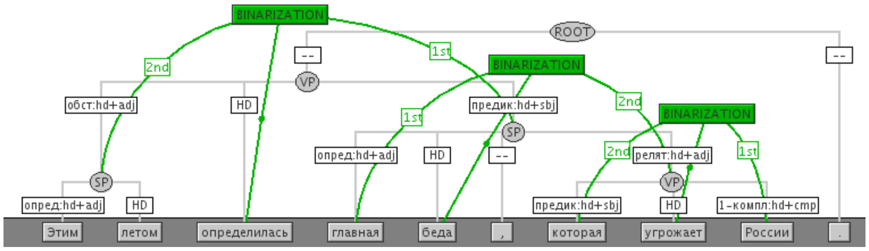


Figure 5: Manual binarization

The learning of such regularities turns out to be more difficult too. For a constituent with a head $H$ together with additional $m$ pre-head daughters and $n$ post-head daughters, there are in total $(m+n)!/(m!\ n!)$ possible binarizations of the tree. While a simple classifier is employed to guess the structure, better formulation of this as a machine learning task will be investigated in the future. Figure 6 shows an example of the binarization result.

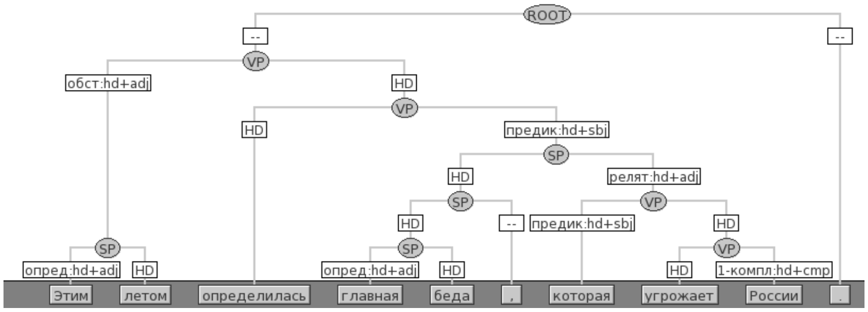


Figure 6: Final structure

It is worth pointing out that the resulting derivation trees only reflect partial view on a complete HPSG analysis. In our case, both corpus and grammar are under parallel development, and draw insights from each other's progress. In future development, we will apply the constraints of the HPSG schemata in the hand-written grammar to the derivation trees. The HPSG signs will be instantiated through this process, allowing us to acquire detailed lexicon for our grammar. For the core grammar development, we are using the DELPH-IN grammar engineering platform (http://www.delph-in.net/), which supports the dynamic evolution of both grammar and treebank as in the LinGO Redwoods approach [14].

# 4 Conclusion

In our view, phenomena-oriented re-structuring of the inventory of surface syntactic relations has the potential of enabling linguistically informed treebank transformation. In this contribution we've presented the first results of creating a constituency treebank of Russian by converting the detailed dependency annotation of SynTagRus to schematic HPSG derivations, taking into account the genuine hierarchy of surface syntactic relations.

The general setup is sketched in Figure 7. We have no access to the grammar and the lexicon of the ETAP-3 linguistic processor [1]. Nevertheless we can utilize the structured linguistic knowledge contained in it, working directly with the output of the system as provided in the syntactic annotation of the SynTagRus treebank. The resulting converted treebank, which we tentatively call SynTagRus++, is of crucial importance for the implementation of a broad-coverage precision Russian resource grammar in the context of creation of open-source Slavic grammatical resources [5]. The latter initiative aims at ensuring an optimal and efficient grammar engineering cycle through dynamic coupling of treebanks, computer grammars and other relevant resources for the Slavic language family.

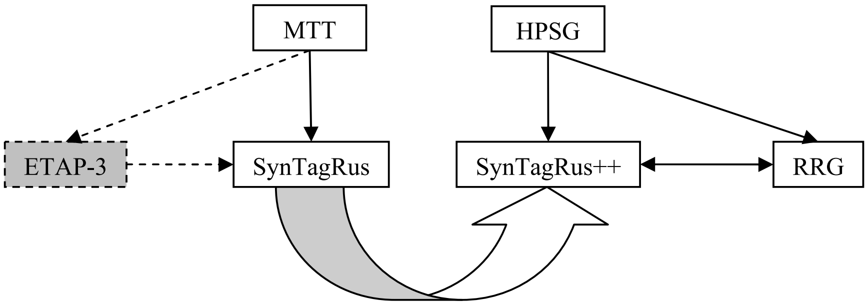


Figure 7: General setup

On the theoretical level our work contributes towards a conceptual alignment between two established linguistic theories: MTT and HPSG. This is a novel and extremely challenging topic, which calls for treebank-supported in-depth cross-theoretical investigations.

# References

[1] Apresian, Juri, Igor Boguslavsky, Leonid Iomdin, Aleksandr. Lazursky, Viktor Sannikov, Viktor Sizov, and Leonid Tsinman. (2003) ETAP-3 linguistic processor: A full-fledged NLP implementation of the MTT. In *First International Conference on Meaning-Text Theory*. p. 279–288.

[2] Apresjan, Juri, Igor Boguslavsky, Boris Iomdin, Leonid Iomdin, Andrei Sannikov, and Victor Sizov (2006) A Syntactically and Semantically Tagged Corpus of Russian: State of the Art and Prospects. In *The fifth international conference on Language Resources and Evaluation, LREC 2006*. Genoa, Italy.

[3] Avgustinova, Tania and Yi Zhang (2009) Developing a Russian HPSG based on the Russian National Corpus. In *DELPH-IN Summit*. Barcelona.

[4] Avgustinova, Tania and Yi Zhang (2009) Exploiting the Russian National Corpus in the Development of a Russian Resource Grammar. In *Proceedings of the RANLP-2009 Workshop on Adaptation of Language Resources and Technology to New Domains*. Borovets, Bulgaria.

[5] Avgustinova, Tania and Yi Zhang (2009) Parallel Grammar Engineering for Slavic Languages. In *Workshop on Grammar Engineering Across Frameworks at the ACL/IJCNLP*. Singapore.

[6] Boguslavsky, Igor, Svetlana Grigorjeva, Nikolai Grigorjev, Leonid Kreidlin, and Nadezhda Frid (2000) Dependency treebank for Russian: Concept, tools, types of information. In *COLING*. p. 987-991.

[7] Boguslavsky, Igor, Ivan Chardin, Svetlana Grigorjeva, Nikolai Grigoriev, Leonid Iomdin, Leonid Kreidlin, and Nadezhda Frid. (2002) Development of a dependency treebank for Russian and its possible applications in NLP. In *Third International Conference on Language Resources and Evaluation (LREC-2002)*. Las Palmas. p. 852–856.

[8] Cahill, Aoife, Mairéad McCarthy, Josef van Genabith, and Andy Way (2002) Automatic Annotation of the Penn-Treebank with LFG F-Structure Information. In *LREC 2002 workshop on Linguistic Knowledge Acquisition and Representation - Bootstrapping Annotated Language Data*. Third International Conference on Language Resources and Evaluation, post-conference workshop: ELRA – European Language Resources Association. p. 8-15.

[9] Hockenmaier, Julia and Mark Steedman (2007) CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. Computational Linguistics 33(3): p. 355-396

[10] Mel'cuk, Igor' A. (1988) Dependency Syntax: Theory and Practice. State University of New York Press.

[11] Mengel, Andreas and Wolfgang Lezius (2000) An XML-based encoding format for syntactically annotated corpora. In *Proceedings of the Second International Conference on Language Resources and Engineering (LREC 2000)*. Athens. p. 121-126.

[12] Miyao, Yusuke, Takashi Ninomiya, and Junichi Tsujii (2005) Corpus-oriented Grammar Development for Acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank, In *Natural Language Processing - IJCNLP 2004, LNAI3248, Hainan Island, China*, Keh-Yih Su, Jun'ichi Tsujii, Jong-Hyeok Lee, and Oi Yee Kwong, Editors. Springer-Verlag. p. 684-693.

[13] Nivre, Joakim, Igor Boguslavsky, and Leonid Iomdin (2008) Parsing the SynTagRus Treebank. In *COLING*. p. 641–648.

[14] Oepen, Stephan, Dan Flickinger, Kristina Toutanova, and Christopher D. Manning (2004) LinGO Redwoods. A rich and dynamic treebank for HPSG. Journal of Research on Language and Computation. 2(4 ): p. 575-596

[15] Tesnière, Lucien (1959) Éléments de syntaxe structurale. Paris Klincksieck.

# Annotation Schema Oriented Validation for Dependency Parsing Evaluation

Cristina Bosco

Università di Torino
Dipartimento di Informatica
E-mail: bosco@di.unito.it

Alberto Lavelli

Fondazione Bruno Kessler, Trento
HLT Research Unit
E-mail: lavelli@fbk.eu

**Abstract**

Recent studies demonstrate the effects of various factors on the scores of parsing evaluation metrics and show the limits of evaluation centered on single test sets or treebank annotation. The main aim of this work is at contributing to the debate about the evaluation of treebanks and parsers, and, in particular, about the influence on scores of the design of the annotation schema applied in the data. Therefore the paper focusses on a dependency-based treebank whose annotation schema includes relations that can be set at different degrees of specificity, and quantitatively describes how the parser performance is affected when processing a selection of hard to parse constructions taken from a recent evaluation campaign for Italian parsing.

## 1 Introduction

In most cases parsers are evaluated against gold standard test data and mainly referring to particular resources, see e.g. the recent shared tasks for multilingual parsers [29, 9] and single language parsers (e.g. [17] for German, [4, 5, 6] for Italian, [30] and http://atoll.inria.fr/passage/eval2.en.html for French). Nevertheless, this kind of evaluation has been criticized under various respects, which are strictly related to the nature of treebanks, showing that scores obtained on a single set of data can be significantly limited by a variety of factors among which the following:

- The domains and genres of texts [14].
- The paradigm and metrics used for the evaluation. Starting from [23, 10], PARSEVAL metrics have been criticized for not representing the real quality of parsing, since they neither weight results nor differentiate between linguistically more or less severe errors [31]. By contrast, dependency–based evaluations and metrics are appreciated since they mainly refer to the encoding of predicate argument structures, a crucial factor for several NLP tasks.

- The language, whose characteristics can influence parsing performance; e.g. a long-standing unresolved issue in parsing literature is whether parsing less-configurational languages is harder than parsing English [16], standing the irreproducibility of the results obtained on the Penn Treebank on other languages.

- The frequency in the test data of constructions which are hard to parse, such as coordination or PP-attachment, where the performance of parsers is much lower than the overall score [32].

- The annotation schema on which the evaluation is based, since treebank annotation schemes may have a strong impact on parsing results [31, 16, 24] and cross–framework evaluation is a complex and unresolved issue. Conversions[1], applied for enabling cross-framework comparisons, are difficult [26, 2, 12] and often decrease the reliability of data introducing errors.

The scenario of parsing evaluation is further complicated by the interrelation of these factors. For instance, [8] demonstrated the influence of annotation schemes on some evaluation metrics, and various scholars often considered differences in schemes applied to different languages among the major causes of the different parsing performance for such languages.

New methods have been proposed to increase the reliability of parsing evaluation, e.g. [18, 32, 33]. They are language-oriented and, at least in principle, framework-independent, and have the advantage of annealing the effects of most of the factors that limit the reliability of evaluations based on test sets. Since these methods focus on specific constructions and explicitly take into account the features of the analyzed language, they can provide additional means to assess parser performance on a linguistic level and enable us to develop more informed comparisons of results across different annotation schemes and languages.

In this paper, we present the application of a similar approach to the dependency parsing of Italian. The main aim of this work is at contributing to the debate about the evaluation of parsing results centered on treebanks, to go beyond the simple assessment of results by presenting evidences about the influence on scores of some of the above mentioned factors, i.e. the language, the frequency of hard to parse constructions, and mainly the design of the annotation schema.
Italian has been selected as a case study because the results of the Evalita'09 Parsing Task (henceforth EPT) [6] have shown that performance is now very close to the scores known for English[2] (top systems LAS are 88.73 and 88.67). They were obtained in EPT by systems based on different assumptions, e.g. rule-based, like TULE [22], and statistical parsers, such as DeSR [1] and MaltParser [28, 20][3],

[1] If the evaluation of a parser P is based on a format F, which is different from that of the output of P, a conversion to F is applied to the output of P and/or to the data used for the training of P.

[2] LAS 89,61 [29] is the best result for English dependency parsing, whilst LAS 86.94 [21] is that previously published for Italian in Evalita'07 Parsing Task [4].

[3] See [29] for the results of DeSR and MaltParser in the CoNLL'07 multi-lingual shared task.

evaluated against two different annotation formats, i.e. those of TUT (Turin University Treebank) and ISST-TANL (Italian Syntactic Semantic Treebank [25]).
Our analysis is based on TUT, which allowed for the best results in EPT, and the MaltParser, a statistical parser tested on different languages and treebanks that participated to EPT with results among the best ones. In particular, we will show experiments focussed on a set of Italian hard to parse constructions and three settings of the annotation schema of TUT, which vary with respect to the amount of underlying linguistic information.
The paper is structured as follows: Section 2 gives an overview of the main features of the TUT treebank and its settings. Section 3 describes the methodology and the experiments. Finally, in Section 4 we discuss the results.

## 2 TUT: data and annotations

TUT[4] is the Italian treebank developed by the Natural Language Processing group of the Department of Computer Science of the University of Turin. The treebank currently includes 2,400 sentences (72,149 annotated tokens in TUT native format) organized in three subcorpora that represent different text genres: newspapers (1,100 sentences), Italian Civil Law Code (1,100 sentences), and 200 sentences from the Italian section of the JRC-Acquis Multilingual Parallel Corpus, a collection of declarations of the European Community shared with the evaluation campaign for parsing French Passage[5].
Even if smaller than other Italian treebanks (i.e. ISST-TANL and the Venice Italian Treebank, VIT, [13]), TUT not only has allowed for best results in EPT, but also makes possible theoretical and applicative comparisons among different formalisms, since TUT is available with annotation formats based on different approaches, e.g. CCG-TUT, a treebank of Combinatory Categorial Grammar derivations [3], and TUT-Penn, a constituency-based treebank [5].

The native annotation scheme of TUT features a pure dependency format centered upon the notion of argument structure, which applies the major principles of Hudson's *word grammar* [15]. This is mirrored, for instance, in the annotation of determiners and prepositions as complementizers of nouns or verbs (see figures below). In fact, since the classes of determiners and prepositions include elements[6] which often are used without complements and can occur alone (like possessive and deictic adjectives or numerals used as pronouns, or prepositions like 'before' and 'after'), all the members of these classes play the same head role when occur with or without nouns or verbs. Moreover, the annotation schema includes null elements to deal with non-projective structures, long distance dependencies, equi

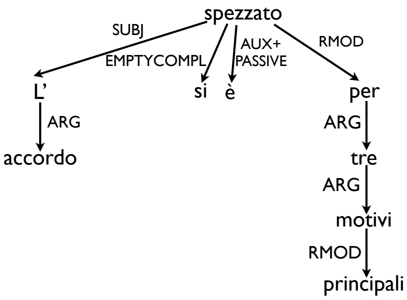
[4] http://www.di.unito.it/∼tutreeb

[5] See http://langtech.jrc.it/JRC-Acquis.html and http://atoll.inria.fr/passage/index.en.html respectively for the JRC-Acquis corpus and Passage.

[6] According to the word grammar, many words qualify as prepositions or determiners which traditional grammar would have classified as adverbs or subordinating conjunctions.

phenomena, pro drop and elliptical structures.

But the most typical feature of the treebank is that it exploits a rich set of grammatical relations designed to represent a variety of linguistic information according to three different perspectives, i.e. morphology, functional syntax and semantics. The main idea is that a single layer, the one describing the relations between words, can represent linguistic knowledge that is proximate to semantics and underlies syntax and morphology, which seems to be unavoidable for efficient processing of human language, i.e. the predicate argument structure of events and states. Therefore, each relation label can in principle include three components, i.e. morpho-syntactic, functional-syntactic and syntactic-semantic, but can be made more or less specialized, including from only one (i.e. the functional-syntactic) to three of them. For instance, the relation used for the annotation of locative prepositional modifiers, i.e. PREP-RMOD-LOC (which includes all the three components), can be reduced to PREP-RMOD (which includes only the first two components) or to RMOD (which includes only the functional-syntactic component).
This works as a means for the annotators to represent different layers of confidence in the annotation, but can also be applied to increase the comparability of TUT with other existing resources, by exploiting the amount of linguistic information more adequate for the comparison, e.g. in terms of number of relations, as happened in EPT. Since in different settings several relations can be merged in a single one (e.g. PREP-RMOD-TIME and PREP-RMOD-LOC are merged in RMOD), each setting includes a different number of relations: the setting based on the single functional-syntactic component (henceforth *1-Comp*) includes 72 relations, the one based on morpho-syntactic and functional-syntactic components (*2-Comp*) 140, and the one based on all the three components (*3-Comp*) 323. In figure 1 the tree (a) for the sentence ALB-356 from TUT corpus, i.e. *"L'accordo si è spezzato per tre motivi principali"* (The agreement has been broken for three main motivations)[7], shows

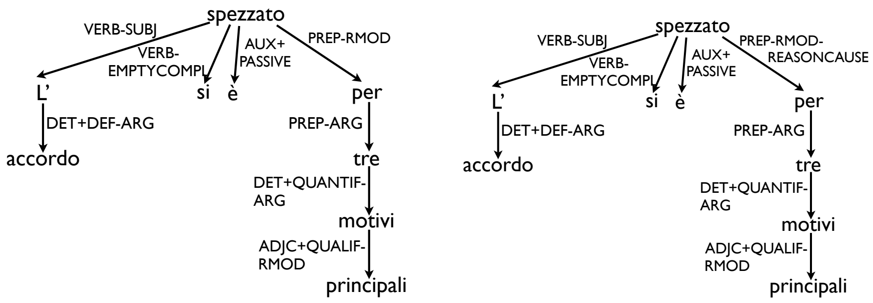


(a)

Figure 1: Sentence ALB–356 in 1–Comp setting, like in EPT.

[7] English translations of the Italian examples are literal and so may appear awkward in English.

the features of the annotation schema. In particular, we see the role of complementizer played by determiners (i.e. the article *"L'"* (The) and the numeral *"tre"* (three)) and prepositions (i.e. *"per* (for)), and the selection of the main verb as head of the structure instead of the auxiliary. If we compare the tree (a) (in figure 1), with the trees (b) and (c) (in figure 2.b and .c), we see also the variation of relations in the three settings for the same sentence. For instance, the relation between *spezzato* (broken) and the prepositional modifier *per tre motivi principali* (for three main motivations), or the argument articles that are ARG in 1-Comp and DET+DEF-ARG (i.e. ARGument of a DEFinite DETerminer) in the other settings. The latter case is an example of relation that does not include semantic information and therefore remains the same in 2- and 3-Comp settings.



(b) (c)

Figure 2: Sentence ALB-356 in: (b) 2-Comp setting; (c) 3-Comp setting.

## 3 Development of the methodology

The approach we propose is language oriented and construction-based, but it differs e.g. both from those in [18] and in [32]. First, by contrast with [18], we follow a pure dependency approach, i.e. the treebank implements a pure dependency annotation, and our analysis is mainly focused on grammatical relations. Second, the selection of the hard to parse phenomena for our experiments is motivated not only by linguistic and applicative considerations, as in related works, but also driven by the performance of different parsers. Third, the analysis is based on three different annotation schemes which are however extracted from the same treebank rather than derived from different sources. Last but not least, our reference language is Italian, which is considered as relatively free word order like German, but less studied until now than Czech or German.

Assuming that most of the parsing errors are related to some specific relation and construction, like in [18, 32], we begin our analysis by identifying cases that can be considered as hard to parse for Italian. For the results of each of the six participant parsers on the EPT test set[8] we compute precision and recall[9] for each type of grammatical relations. To further assess the results, we perform the same kind of evaluation on the three relation settings running a 10-fold cross validation on the entire treebank with MaltParser. After identifying the hard to parse relations, we develop a comparative analysis of the behavior of MaltParser in such cases.

### 3.1 Selecting phenomena and features

Observing the average score of the six parsers which participated in EPT we can identify the following hard to parse constructions:

- the predicative complement of the object, i.e. PREDCOMPL+OBJ (which occurs 141 times in the full treebank, i.e. 0.19%). For instance, in *"Il parlamentare si è detto* ***favorevole*** *ad una maggiore apertura delle frontiere ai rifugiati politici."*(The parliamentarian itself has said **in favour** of a major opening of frontiers to the political refugees.)
- the indirect object, i.e. INDOBJ (which occurs 325 times, i.e. 0.45%). For instance, in *"Noi non permetteremo* ***a nessuno*** *di imbrogliarci."* (We will not allow **to anybody** to cheat us.)
- various relations involved in coordinative structures that represent comparisons (e.g. COORDANTEC+COMPAR and COORD+COMPAR (which occurs 64 times, i.e. 0,08%), like in *"Usa un test* ***meno*** *raffinato* ***di quello tradizionale****."* ([He] exploits a test **less** refined **than the traditional one**.)).
- various relations for the annotation of punctuation, in particular SEPARATOR, OPEN+PARENTHETICAL (which occurs 1,116 times, i.e. 1.5%) and CLOSE +PARENTHETICAL (which occurs 1097 times, i.e. 1.5%)). For instance, SEPARATOR (which occurs 1,952 times, i.e. 2.7%) is used in cases where commas play the role of disambiguating marks and an ambiguity could result if the marks were not there [19], e.g. in *"Quando il meccanismo si inceppa, è il disastro."* (When the mechanism hinds itself, is a disaster). OPEN+/CLOSE+PARENTHETICAL are instead used for the annotation of paired punctuation that marks the parenthetical in *"Pochi quotidiani* ***,*** *solo quelli inglesi****,*** *saranno oggi in vendita."* (Few newspapers**,** only those English**,** will be today on sale.).

Since not all the grammatical relations of 1-Comp occur in the test set, the above list cannot be in principle considered as representative of how hard to parse is the treebank (and the Italian language). A 10-fold cross validation performed on the whole TUT with the 1-Comp setting shows that other low-scored relations exist, but since they appear with a very low frequency we did not include them in our

[8] The EPT test set included 240 sentences (5,287 tokens) balanced alike to those of the treebank used for training: 100 sentences (1,782 tokens) from newspapers, 100 (2,293 tokens) from Civil Law Code and 40 (1,212 tokens) from the Passage/JRC-Acquis corpus.

[9] The evaluation has been performed by using the MaltEval tools [27].

experiments. This shows however that the test set, even if it shows the same balancement of TUT, does not represent at best the treebank in terms of relations and constructions. Moreover, a comparison with ISST-TANL, based on the EPT results and developed in [6] and [7], shows that similar relations, in particular coordination and punctuation, are low-scored also in this other resource, notwithstanding the different underlying annotation schema where, e.g. it is the determiner which depends on the noun. Nevertheless this comparison is of limited interested, since in ISST-TANL the annotation of punctuation is far less fine-grained than in TUT.

### 3.2 Comparing the test set and the whole treebank

The comparisons of this section exploit the relation settings of TUT, and are oriented to the assessment of the influence of the annotation schema design on parsing results. They show that the evaluation has to be weighted observing at least the distribution and kind of hard to parse constructions and the degree of difficulty of hard to parse constructions, which can vary in the test set and in the whole treebank.

First of all, we test the hypothesis that the test set is an aggregate over a highly skewed distribution of relations and constructions, where the frequency of hard to parse phenomena can be different from that of the whole treebank. The application of MaltParser on all the treebank with the 1-Comp setting, like in the EPT test set, exploiting a 10-fold cross validation strategy shows that this hypothesis is correct, since the performance significantly varies when the parser is applied to the EPT test set rather than to all the treebank, i.e. from LAS 86.5 and UAS 90.96, in the test set [20], to LAS 83.24 e UAS 87.69 in all TUT[10]. This suggests that the distribution of hard to parse phenomena is not the same in both cases.

In order to test the hypothesis that the degree of difficulty of the same hard to parse constructions can vary in the test set with respect to the treebank, we first analyze the performance of MaltParser on all TUT with the 3 settings, and, second, we analyze the variation of precision and recall for each hard to parse case according to the three settings. As table 1 shows, the performance in terms of UAS is not significantly influenced by the different settings, since the difference concerns the relation labels rather than the tree structures. Instead, LAS decreases when the number of relations is enlarged in settings that should be more informative, going from 72 (1-Comp), to 140 (2-Comp), to 323 relations (3-Comp). The larger amount of relations occurring a small number of times in 2- and 3-Comp (with

| | **1-Comp** | **2-Comp** | **3-Comp** |
|---|---|---|---|
| **LAS** | 83.24 | 82.56 | 78.77 |
| **UAS** | 87.69 | 87.60 | 87.20 |

Table 1: MaltParser scores in 10-fold cross validation over the whole treebank.

[10] This is only partially explained by the sentence length, which is lower than 40 words only in the test set, and by the smaller size of the training set for the 10-fold cross validation.

respect to 1-Comp) increases the sparseness of relations and negatively influences the performance. Also the stability across all settings of the performance only on more frequent relations, further supports this conclusion.

Now we focus on single hard to parse relations in order to show the variation of parser performance in the three settings. Tables 2, 3 and 4 show that the parser behavior varies in different way for different relations and sometimes following a different trend with respect to the results on all the treebank. For instance, for COORD+COMPAR (table 2) the best performance is in 1-Comp and the worst in the EPT test set. For PREDCOMPL+OBJ (table 3), instead, the best performance is in 3-Comp and the worst in the EPT test set. Therefore, in this case there is a contrast with the general trend shown in table 1, since the results are significantly better when the relation labels include the morphological component.

| | **EPT** | **1-Comp** | **2-Comp** | **3-Comp** |
|---|---|---|---|---|
| prec | 50.00 | 89.66 | 83.33 | 86.21 |
| rec | 25.00 | 54.17 | 52.08 | 52.08 |

Table 2: MaltParser scores for COORD+COMPAR with different settings.

| | **EPT** | **1-Comp** | **2-Comp** | **3-Comp** |
|---|---|---|---|---|
| prec | 50 | 57.81 | 60.00 | 61.16 |
| rec | 40 | 52.48 | 53.19 | 52.48 |

Table 3: MaltParser scores for (VERB-)PREDCOMPL+OBJ with different settings.

| | **EPT** | **1-Comp** | **2-Comp** | **3-Comp** |
|---|---|---|---|---|
| prec | 68.97 | 57.00 | 55.96 | 48.26 |
| rec | 58.82 | 52.35 | 50.49 | 63.19 |

Table 4: MaltParser scores for (VERB-)INDOBJ with different settings.

For what concerns instead punctuation, we observe that it is not always considered when performing evaluation. As we have seen before, in our evaluation punctuation is instead taken into account, but the related relations are among the low-scored ones. For instance, SEPARATOR (see section 3.1) is in the set of the 9 most frequent relations[11] (in 1-Comp setting in both all the treebank and the test set) and occurs around 2,000 times in the full treebank, but it is the one scoring the lower

[11] The ten most frequent relations in all the 1-Comp treebank (with respect to 72,149 annotated tokens) are ARG (30.3%), RMOD (19.2%), OBJ (4.5%), SUBJ (3.9%), END (3.3%), TOP (3.2%), COORD2ND+BASE (3.1%), COORD+BASE (3.1%), SEPARATOR (2.7%), INDCOMPL (1.9%).

in precision and recall of this set for all the parsers participating to EPT. Therefore, in the perspective of a comparison with other evaluations and resources, it would be useful to see how our results vary when punctuation is excluded, as in table 5. The UAS and LAS scores of MaltParser are in all TUT settings 3.5 points higher when the punctuation is not taken into account. As for ISST-TANL, the experiments show that the difference in performance when considering or not considering punctuation is between 1.76 and 2.50 according to different parser parameters. This lower difference can be explained by the different annotation of punctuation, less fine-grained in ISST-TANL where a single relation PUNC is used. This means that some improvement in parsing can be obtained by more adequate processing of punctuation, as said e.g. in [11], and/or by more adequate annotation of it. In fact punctuation is often relevant from a linguistic point of view as a marker of clause or phrase boundaries, thus if a parser does not predict it correctly, it can lead to incorrect parses and lower scores when evaluated against a resource that annotates punctuation.

| | **1-Comp** | **2-Comp** | **3-Comp** |
|---|---|---|---|
| **LAS Punct** | 83.24 | 82.56 | 78.77 |
| **LAS noPunct** | 86.78 | 86.02 | 81.88 |
| **UAS Punct** | 87.69 | 87.60 | 87.20 |
| **UAS noPunct** | 91.10 | 91.01 | 90.70 |

Table 5: MaltParser scores on 1-, 2- and 3-Comp TUT with and without punctuation, in 10-fold cross validation.

As for the comparison with other languages, we have seen that part of the hard to parse phenomena for Italian are included also in the test suites proposed for German, e.g. forms of coordination. But, since the lists presented in [18] and in [32] are mainly linguistically motivated and not quantitatively determined, we cannot go beyond this observation and further extend the comparison here.

For what concerns single phenomena, following the idea that parsing can be made more or less hard by the availability of different amount of linguistic information, we have seen that different effects can be caused by the use of more or less informative grammatical relations. The results demonstrate, in particular, that the evaluation based on the test set is limited with respect to the distribution and kind of hard to parse constructions, which in the test set and in the treebank can be different, and the degree of difficulty of hard to parse constructions, which in the test set and in the treebank can be not the same.

## 4 Conclusions and future work

Most parser evaluations are based on single resources, but the design and features of the treebank used for testing can strongly influence the results.

This paper presents issues for the development and application to Italian parsing of a methodology for the validation of the evaluation of parsing results. Starting from the results of an evaluation campaign for Italian parsing, i.e. EPT, it provides evidence about the skewedness of the test set of this contest. The experiments presented confirm the hypothesis that evaluations based on test sets and single resources present several shortcomings. They demonstrate, in particular, that the validity of an evaluation based on a test set and a single resource is limited with respect to the distribution and kind of hard to parse constructions, which in the test set and in the treebank can be different, and with respect to the degree of difficulty of hard to parse constructions, which in the test set and in the treebank can vary.

A variety of directions for future research is raised by the present work that go beyond the simple assessment of the results to give suggestions for both treebank design and the development of more informed evaluation methodologies. Among them, in particular, a deeper analysis of the presented data and results by trying new experiments based also on parsers that apply different approaches, e.g. TULE; the comparison with other existing resources and annotation schemes, and last but not least the comparison with other languages.

## References

[1] Attardi, G., Dell'Orletta, F., Simi, M. and Turian, J. (2009) Accurate dependency parsing with a stacked multilayer perceptron. In *Proceedings of Evalita'09*, Reggio Emilia.

[2] Bick, E. (2006) Turning a dependency treebank into a PSG-style constituent treebank. In *Proceedings of LREC'06*, pp. 1961–1964, Genova.

[3] Bos, J., Bosco, C. and Mazzei, A. (2009) Converting a Dependency Treebank to a Categorial Grammar Treebank for Italian. In *Proceedings of TLT-8*, pp. 27–38, Milano.

[4] Bosco, C., Mazzei, A. and Lombardo, V. (2007) Evalita Parsing Task: an analysis of the first parsing system contest for Italian. In *Intelligenza Artificiale*, Vol. 2, pp. 30–33.

[5] Bosco, C., Mazzei, A. and Lombardo, V. (2009) Evalita Parsing Task 2009: constituency parsing and a Penn format for Italian. In *Proceedings of Evalita'09*, Reggio Emilia.

[6] Bosco, C., Montemagni, S., Mazzei, A., Lombardo, V., Dell'Orletta, F. and Lenci, A. (2009) Evalita'09 Parsing Task: comparing dependency parsers and treebanks. In *Proceedings of Evalita'09*, Reggio Emilia.

[7] Bosco, C., Montemagni, S., Mazzei, A., Lombardo, V., Dell'Orletta, F., Lenci, A., Lesmo, L., Attardi, G., Simi, M., Lavelli, A., Hall, J., Nilsson,

J. and Nivre, J. (2010) Comparing the Influence of Different Treebank Annotations on Dependency Parsing. In *Proceedings of LREC'10*, pp. 1794–1801, Malta.

[8] Boyd, A. and Meurers, D. (2008) Revisiting the impact of different annotation schemes on PCFG parsing: a grammatical dependency evaluation. In *Proceedings of ACL'08: HLT Workshop on parsing German*, pp. 24-32, Columbus Ohio.

[9] Buchholz, S. and Marsi, E. (2007) CoNLL-X Shared Task on Multilingual Dependency Parsing. In *Proceedings of CoNLL-X*, pp. 149-164, New York.

[10] Carroll, J., Briscoe, T. and Sanfilippo, A. (1998) Parser evaluation: a survey and a new proposal. In *Proceedings of LREC'98*, pp. 447-454, Granada.

[11] Cheung, J. C.K. and Penn, G. (2009) Topological field parsing of German. In *Proceedings of ACL-IJCNLP'09*, pp. 64-72, Singapore.

[12] Clark, S. and Curran, J. R. (2007) Formalism-independent parser evaluation with CCG and DepBank. In *Proceedings of ACL'07*, pp. 248-255, Prague.

[13] Delmonte, R. (2008) *Strutture sintattiche dall'analisi computazionale di corpora di italiano.* Milano: Franco Angeli.

[14] Gildea, D. (2001) Corpus variation and parser performance. In *Proceedings of EMNLP'01*, pp. 167-202, Pittsburg.

[15] Hudson, R. (1984) *Word grammar*, Oxford and New York: Basil Blackwell.

[16] Kübler, S., Hinrichs, H. and Maier, W. (2006) Is it really that difficult to parse German?. In *Proceedings of EMNLP'06*, pp. 111-119, Sydney.

[17] Kübler, S. (2008) The PaGe 2008 shared task on parsing German. In *Proceedings of ACL Workshop on parsing German*, pp. 55-63, Columbus Ohio.

[18] Kübler, S., Rehbein, I. and van Genabith, J. (2009) TePaCoC a corpus for testing parser performance on complex German grammatical constructions. In *Proceedings of TLT-7*, pp. 15–28, Groningen: The Netherlands.

[19] Jones, B. E. M. (1994) Exploring the role of punctuation in parsing natural text. In *Proceedings of COLING'94*, pp. 421-425, Kyoto.

[20] Lavelli, A., Hall, J., Nilsson, J. and Nivre, J. (2009) MaltParser at the Evalita 2009 Dependency parsing task. In *Proceedings of Evalita'09*, Reggio Emilia.

[21] Lesmo, L. (2007) The rule-based parser of the NLP group of the University of Torino. In *Intelligenza Artificiale*, Vol. 2, pp. 46–47.

[22] Lesmo, L. (2009) The Turin University Parser at Evalita 2009. In *Proceedings of Evalita'09*, Reggio Emilia.

[23] Lin, D. (1995) A dependency-based method for evaluating broad-coverage parsers. In *Proceedings of IJCAI'95*, pp. 1420-1427, Montreal.

[24] Maier, W. (2006) Annotation schemes and their influence on parsing results. In *Proceedings of COLING-ACL'06 Student Research Workshop*, pp. 19-24, Sydney.

[25] Montemagni, S., Barsotti, F., Battista, M., Calzolari, N., Corazzari, O., Lenci, A., Pirrelli, V., Zampolli, A., Fanciulli, F., Massetani, M., Raffaelli, R., Basili, R., Pazienza, M. T., Saracino, D., Zanzotto, F., Mana, N., Pianesi, F. and Delmonte, R. (2003) Building the Italian Syntactic-Semantic treebank. In *Building and using Parsed Corpora* A. Abeillè (ed.), pp. 189–210, Dordrecht: Kluwer.

[26] Musillo, G., Sima'an, K. (2002) Towards comparing parsers from different linguistic frameworks. An information theoretic approach. In *Proceedings of Workshop Beyond PARSEVAL - Towards improved evaluation measures for parsing systems at the LREC'02*, pp. 44–51, Las Palmas Canary Islands.

[27] Nilsson, J., Nivre, J. (2008) MaltEval: An Evaluation and Visualization Tool for Dependency Parsing. In *Proceedings of LREC'08*, pp. 161–166, Marrakech.

[28] Nivre, J., Hall, J. and Nilsson, J. (2006) MaltParser: A Data-Driven Parser-Generator for Dependency Parsing. In *Proceedings of LREC'06*, pp. 2216–2219, Genova.

[29] Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S. and Yuret, D. (2007) The CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of EMNLP-CoNLL'07*, pp. 915–932, Prague.

[30] Paroubek, P., Vilnat, A., Loiseau, S., Hamon, O., Francopoulo, G., and Villemonte de la Clergerie, E. (2008) Large scale production of syntactic annotations to move forward. In *Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pp. 36–43, Manchester UK.

[31] Rehbein, I. and van Genabith, J. (2007) Treebank annotation schemes and parser evaluation for German. In *Proceedings of EMNLP-CoNLL'07*, pp. 630–639, Prague.

[32] Rimell, L. and Clark, S. and Steedman, M. (2009) Unbounded dependency recovery for parser evaluation. In *Proceedings of EMNLP'09*, pp. 813–821, Singapore.

[33] Tam, W. L., Sato, Y., Miyao, Y. and Tsujii, J. (2008) Parser evaluation across frameworks without format conversion. In *Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pp. 29–35, Manchester UK.

# Building and exploiting a dependency treebank for French radio broadcasts

Christophe Cerisara, Claire Gardent and Corinna Anderson

CNRS/LORIA, Nancy
Firstname.Lastname@loria.fr

**Abstract**

We describe the construction of a dependency treebank for French radio broadcasts and present some results on how genre-specific phenomena affect parsing. Preliminary experimental results realized on one hour of speech suggest in particular that not only disfluencies but also radio headers and guest speech have a negative impact on parsing accuracy.

## 1 Introduction

Much work in recent years has focused on developing treebanks for transcribed speech. For English, the most well known is the Switchboard corpus [7]. For French however, the only efforts made in this direction we are aware of concerns the syntactic annotations of the European parliament debates [2]. Although this treebank is very large, it has only been automatically analyzed so far. We thus initiate in this work some efforts to manually analyze a more common type of speech, broadcast news transcripts.

The ESTER Corpus contains transcripts of broadcast news while the Media corpus contains about 70 hours of dialogs which were manually transcribed and semantically annotated with a set of 80 basic concepts. However, neither of these two corpora is syntactically annotated. Similarly, the Paris 7 treebank (P7TB, 12 500 sentences, 325 000 words, [1]) consists of articles from *Le Monde* newspaper semi-automatically enriched with phrase structure annotations and manually verified. The P7 dependency treebank (P7Dep, [6]) was automatically derived from it by conversion. Neither of these treebanks however contain a sizable portion of speech.

In this paper, we report on the construction and exploitation of a dependency treebank for spoken French. We start by presenting the annotation schema used and relate it to the schema used for written French in the P7Dep treebank (Section 2). We then describe the tools and methodology used to construct the treebank (Section 3). Finally, we exploit the speech treebank for training a parser and present some first results concerning the impact of various speech-specific constructs on speech parsing. In particular, we show that for broadcast news, disfluency might not be the main factor for performance degradation, and that genre-specific constructs such as headlines and guest speech, which tends to be characteristic of natural unplanned discourse (as opposed to the prepared speech of professional radio journalists), play an important role in performance loss (Section 4).

## 2 Corpus and Annotation schema

To develop a treebank of spoken French, we build on existing work and take as a starting point the ESTER corpus of French radio broadcasts and the P7Dep and Syntex [3] annotation schema for dependency structures.

### 2.1 Corpus

The corpus used to develop a treebank of spoken French (the ESTER treebank henceforth, ETB) is the ESTER corpus of manual transcripts for French radio news (1998 - 1999 and 2003).

Because these transcripts were developed with the aim to evaluate speech recognisers, only complete words were transcribed. Hesitations “euh” were considered as words and were transcribed but noise, starts, laughs, jingles indications, etc. were not. For parsing, all punctuation information was removed so as to simulate the output of a speech recogniser, which typically does not produce such information.

Further, words are grouped by the transcribers into prosodic segments which do not necessarily coincide with sentences. During the annotation however, the annotators can join segments together whenever the prosodic segments form incomplete constituents.

### 2.2 The ESTER Annotation schema and its relation to the P7Dep annotation schema

The annotation schema (The ESTER Annotation schema) we define to annotate the ESTER corpus is derived from our previous works with the Syntex parser [3] and with the P7Dep corpus. It comprises 15 dependency relations: SUJ (subject), OBJ (object), POBJ (prepositional object), ATTS (subject attribute), ATTO (object attribute), MOD (modifier), COMP (complementizer), AUX (auxiliary), DET (determiner), CC (coordination), REF (reflexive pronoun), JUXT (juxtaposition), APPOS (apposition), DUMMY (syntactically governed but semantically empty dependent e.g. expletive subject “il / it” in “il pleut / it rains”), DISFL (disfluency).

As shown in Table 1, this schema has a direct partial mapping to the annotation schema used to annotate the P7 treebank of newspaper text. The differences between the two annotation schemes relate to prepositional objects, auxiliaries, modifiers and speech or written text specific constructs such as disfluencies (speech) or punctuation (text).

We thus defined and implemented a rule-based converter from the ETB to the P7Dep formats as follows. Prepositional objects are differentiated in the P7Dep annotation schema, between a_obj, de_obj and p_obj , while in the ETB, all prepositional objects are marked as POBJ. To convert from ETB to P7Dep, we systematically convert prepositional objects headed with the "à" and "de" preposition into a_obj and de_obj respectively. For clitics (which do not contain a preposition and can be ambiguous), we use default mappings and exception lists for non-default cases. For instance, the clitic "*en*" usually indicates a de_obj (e.g., *en rêver / rêver de Paris*) but can also pronominalise the object NP (e.g., *Jean donne des pommes à Marie / Jean en donne à Marie*). We use the information that *donne* is a ditransitive and *rêver* a de_obj verb to appropriately map the clitics *en* to obj and de_obj respectively.

To differentiate between the various types of auxiliaries (passive, causative or temporal), we use hand written rules and lists of e.g., passivisable verbs, causative verbs and temporal auxiliaries that have been compiled over the years in our team. For instance, we use the list of passivisable verbs and pattern matching rules to decide whether the auxiliary "be" is used as a passive (*Il est aimé / He is loved*) or as present perfect (*Il est venu / He has arrived*) auxiliary.

Similarly, modifiers are differentiated into mod_rel (a relative clause modifying a noun), dep (a prepositional phrase modifying something else than a verb) and mod (all other types of modifiers) using hand-written pattern matching rules describing these three configurations. Further rules are used to convert coordination constructs (coord and dep_coord P7Dep relations) and reflexive clitics (REF ETB relation). Relations that have an onto mapping in the P7Dep format (SUJ, ATTS, ATTO, OBJ, COMP, DET, DUMMY) are mapped to the corresponding ETB relation. Relations that exists only in the ETB (DISFL,JUXT,APPOS,MULTIMOTS) are all mapped to the mod relation.

We assessed the accuracy of these conversion rules on the ESTER test corpus manually annotated in the P7Dep format: the conversion labelled attachment score (percentage of tokens with correct predictor governor and dependency type, LAS) is 92.6% and unlabelled attachment score (the ratio of words with a correct head, UAS) is 98.5%.

### 2.2.1 Annotation of speech-specific constructs.

Some constructs which are very frequent in speech are either absent in the P7Dep schema (disfluencies) or not differentiated from one another (juxtaposition and apposition treated as modification). To allow for a detailed study of these constructs, the ESTER schema labels them separately and annotates them as described below. Additionnally, sentence-level annotations were introduced in order to support a finer-grained analysis of the impact of speech constructs on parsing.

| ETB | Label description | P7Dep |
|---|---|---|
| MOD | modifier | mod<br>mod_rel<br>dep |
| COMP | complementizer | obj |
| DET | determiner | det |
| SUJ | subject | suj |
| OBJ | object | obj<br>aux_caus |
| DISFL | disfluency | mod |
| CC | coordination | coord<br>dep_coord |
| POBJ | prepositional object | a_obj<br>de_obj<br>p_obj |
| ATTS | subject attribute | ats |
| JUXT | juxtaposition | mod |
| MultiMots | multi-word expression | mod |
| AUX | auxiliary | aux_tps<br>aux_pass<br>aux_caus |
| DUMMY | empty dependent | aff |
| REF | reflexive pronoun | obj<br>a_obj<br>de_obj |
| APPOS | apposition | mod |
| ATTO | object attribute | ato |

Table 1: The mapping between the ESTER and the P7 dependency relations

**Juxtaposition, apposition and disfluencies** While apposition and juxtaposition occur in written text as well as in speech, both relations are labeled as *mod* in the P7 treebank. Because they occur very frequently in transcribed speech, we created specific labels (*appos* and *juxt* ) for them in the ESTER corpus.

The apposition relation is used to relate two adjacent constituents denoting the same referent (1a)[1].

(1) a. Jean Tiberi le maire de la capitale *maire = appos ( Jean )*
*Jean Tiberi the mayor of the capital city*

b. la seconde modification de la constitution celle qui concerne la réforme législative *celle = appos ( modification )*
*The second modification of the constitution that which concerns the legal reform*

[1] The notation *D = rel ( G )* indicates that the dependent D is related to its governor G by the dependency relation *rel*.

In contrast, the juxtaposition relation is used to relate constituents which occur in the same prosodic group but are not syntactically related: juxtaposed constituents (2a), enumeration (2b) and incidents (2c).

(2) a. le téléphone sonne bonsoir *bonsoir = juxt ( téléphone )*
*Le téléphone sonne good evening*[2]

b. des gens qui roulent à 70 80 *80 = juxt ( 70 )*
*People who drive at 70 80 mph*

c. Un voyage j'espère agréable *espère = juxt ( voyage )*
*A trip I hope pleasant*

Disfluencies include hesitations, repairs and missing words. Hesitations (euh, 3a) are attached as a dependent to the head of the right adjacent constituent. When a word is missing (3b), the incomplete constituent is linked to the intended governor. Finally, in case of repairs, if the disfluency forms a constituent (3c), its head is linked to the head of the repair. Otherwise, the tokens forming the disfluency are each attached to the right adjacent token (3d).

(3) a. des conditions euh identiques *euh = disfl ( identiques )*
*Conditions hum identical*

b. on n' attend pas euh euh 6 mois lui dire ce que tu as fait n' est pas bien
*dire = disfl ( attend )*
*You dont wait hum hum 6 months tell him that what you did was not good*

c. Nous en relevons nous le relevons
*nous = suj ( relevons ) en = mod ( relevons ) relevons = disfl ( nous )*
*We notice them we notice it*

d. et bien je ... *et = disfl ( je )* *ben = disfl ( je )*
*and so I ...*

**Sentence level annotations** As mentioned above, we introduced sentence-level annotations in addition to the dependency annotations in order to facilitate data analysis. These sentence-level annotations are the following:

- GUEST / SPEAKER: differentiates guests utterances from radio speaker ones
- ELLIPSIS: indicates that the sentence contains an ellipsis
- HEADER: indicates a radio header e.g.,(4)

(4) Michel à Aix-en-Provence en ligne bonsoir Michel
*Michel from Aix-en-Provence on line good evening Michel*

[2] *Le téléphone sonne* is the name of the show.

# 3 Treebank construction

Manual annotation of a full raw textual corpus with dependencies is time consuming, error prone and cognitively demanding for the human annotators. We have therefore opted for an iterative annotation procedure alternating training, parsing and manual correction. The cognitive effort required by the human annotator per sentence is thus greatly reduced, as she only has to check the proposed dependencies and possibly to modify some of them. The corpus produced in this way is further validated by an expert linguist. This validation phase involves several meetings between the annotators and the expert linguist where ambiguous and difficult examples are discussed and solved.

## 3.1 Methodology

The annotation procedure is as follows:

1. The manual transcription of a contiguous session of one hour-length is extracted from the development set of the broadcast news ESTER corpus [4]. This session constitutes the raw corpus that is annotated next.

2. This full raw unlabeled text corpus is split into 17 sub-corpora $(C_1,\cdots,C_{17})$ of about 630 words each. These sequences of words are manually segmented into utterances during the course of the following annotation procedure.

3. At iteration $t$, the unlabeled sub-corpus $C_t$ is first automatically tagged with POS tags with the TreeTagger configured for French.

4. $C_t$ is then automatically parsed with the Malt Parser [8] (cf. section 3.2) and the previous models $\lambda_{t-1}$. Note that the initial models $\lambda_0$ have been trained on another corpus of 20000 words that has been previously annotated with dependencies, but which is not used in the work described here [3].

5. The annotator (a linguistics student) then loads $C_t$ into the edition software J-Safran, segments it into utterances, checks the proposed dependency labels and may modify them according to the annotation guide.

6. The malt parser models are retrained on $C_t$, leading to the new parameters $\lambda_t$.

7. This process is iterated from step 4 until the whole corpus is annotated.

This iterative process is interleaved with meetings between the annotators and the expert linguist whose aim is to discuss outstanding issues and to validate the annotations produced. The treebank thus obtained contains 1 hour of speech, i.e., 10654 words and 594 segments. The distribution of dependencies is shown in table 2.

[3] Because of a minor mismatches in the annotation schemas and because of the absence of sentence-level annotations

| Dependency | Description | Number of occurrences | Proportion |
|---|---|---|---|
| MOD | modifier | 2707 | 27% |
| COMP | complementizer | 1723 | 17% |
| DET | determiner | 1346 | 13% |
| SUJ | subject | 1073 | 11% |
| OBJ | object | 843 | 8% |
| DISFL | disfluency | 580 | 6% |
| CC | coordination | 495 | 5% |
| POBJ | prepositional object | 312 | 3% |
| ATTS | subject attribute | 198 | 2% |
| JUXT | juxtaposition | 180 | 2% |
| MultiMots | multi-word expression | 179 | 2% |
| AUX | auxiliary | 161 | 2% |
| DUMMY | empty dependent | 91 | <1% |
| REF | reflexive pronoun | 75 | <1% |
| APPOS | apposition | 62 | <1% |
| ATTO | object attribute | 18 | <1% |
|  | **total:** | 10043 |  |

Table 2: Distribution of the different types of dependencies in the ETB corpus

## 3.2 Software environment

The J-Safran platform was developed with the aim to facilitate the iterative annotation procedure described in the previous section. One important motivation for developing yet another annotation platform was the need for easy use, installation and portability. Because the annotators were linguistics students working from home, it was necessary to have a platform that could be easily installed and used under different operating systems. Another important motivation was the need for easy modification and extension. For instance, we recently extended J-Safran to support joint syntactic-semantic annotation in view of adding semantic role labels and training a semantic role labeller for French.

Implemented in Java and available in open source on the web[4], J-Safran (Java Syntaxico-semantic French Analyser) integrates the following modules:

- The Malt Parser: a deterministic shift-reduce parser with a machine learning approach for computing local decisions and actions. The version used in this work exploits a Support Vector Machine (SVM) for this purpose, and integrates an interface module that facilitates the control of the parser from the Graphical User Interface (GUI).

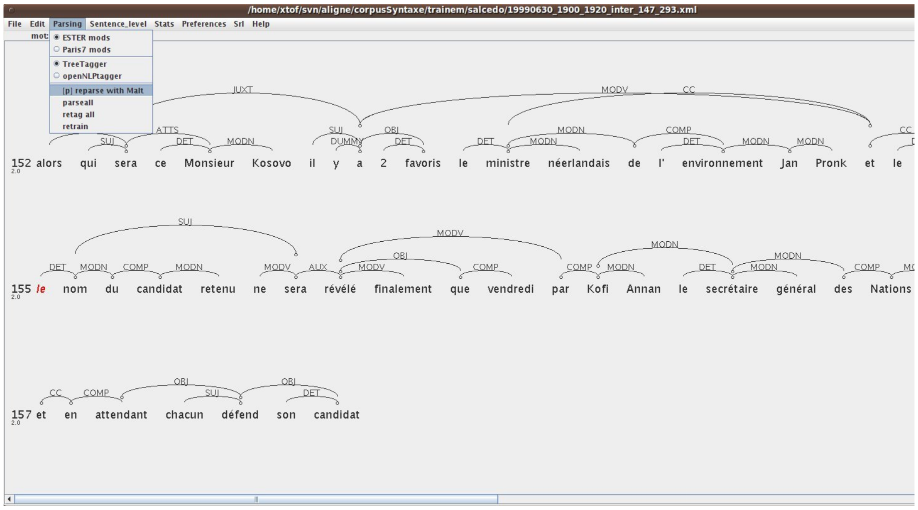
- A part-of-speech (POS) tagger: the TreeTagger [9] with its associated Java Wrapper TT4J [5];

[4] http://www.loria.fr/~cerisara/jsafran/index.html

[5] http://www.annolab.org/tt4j

- The evaluation scripts derived from the standard CoNLL evaluation campaign [10].
- A Java GUI that provides most common vizualisation, editing and search functionalities for dependency annotations.

A screenshot of the J-Safran GUI is shown in Figure 1.



/home/xtof/svn/aligne/corpusSyntaxe/trainem/salcedo/19990630_1900_1920_inter_147_293.xml

152 alors qui sera ce Monsieur Kosovo il y a 2 favoris le ministre néerlandais de l' environnement Jan Pronk et le

155 *le* nom du candidat retenu ne sera révélé finalement que vendredi par Kofi Annan le secrétaire général des Nations

157 et en attendant chacun défend son candidat

Figure 1: Screenshot of the J-Safran GUI for dependency tree edition

# 4 The impact of spoken data on parsing accuracy

We used the ESTER treebank, annotated in dependencies, to train and test the Malt parser. Using 8544 words for training and 1747 words for testing, we obtained a LAS (labelled attachment score i.e., percentage of tokens with correct predictor governor and dependency type) of 63.6% . Unsurprisingly, training a parser on such a small quantity of annotated speech transcripts yields results well below the state of the art both for written and for spoken data. Training on a larger speech corpus would obviously help improve performance. However, syntactic annotation is costly, and it would be both useful and interesting to have a better understanding of which phenomena in speech most affect parsing performance. Such an understanding could be used for instance, either to manually enrich the training corpus with annotated data for these phenomena or, within an active learning approach, to guide the automated selection of the data to be annotated.

## 4.1 Impact of disfluencies

We start by investigating the impact of disfluencies. A characteristic feature of spoken language is that, because there is no possibility of deleting what has been said, all editing is performed online so that utterances often contain disfluencies i.e., false start, filled pauses, word fragments, repetitions, corrections and interruptions. To determine the impact of disfluencies on parsing, we manually remove disfluencies in the test corpus so as to isolate the impact of disfluencies from other factors, such as sentence length. Indeed, preliminary experiments indicate that disfluencies occur more frequently in longer sentences. The results are shown in Table 3. On disfluent sentences, disfluencies degrade the Labeled Attachment Score (LAS) by 4.1 points. The two other CoNLL scores given are the Unlabeled Attachment Score (UAS) and the Label Accuracy score (LAC).

| | W/o disfl | With disfl | Δ(w,w/o) |
|---|---|---|---|
| LAS | 70.2% | 66.1% | +4.1 |
| UAS | 77.2% | 73.5% | +3.7 |
| LAC | 76.5% | 72.7% | +3.8 |

Table 3: Comparing parsing scores on test sentences with disfluencies (*with*) and after manual deletion of disfluencies (*without*). Only that part of the test corpus that contains disfluencies is considered here.

In the test corpus, 41% of the sentences are sentences with disfluencies and 59% sentences without disfluencies. To evaluate the impact of disfluencies on the whole test corpus, we compare the parsing scores on the raw test corpus and on the same corpus after manual correction of disfluencies. The results are shown in Table 4. As expected, the impact of disfluencies is qualitatively the same than in the previous test, but it is quantitavely lower, because the majority of sentences do not have disfluencies at all in this corpus. This experiment gives a better idea of the actual, effective impact of disfluencies on parsing in real conditions. Roughly, disfluencies account for a decrease in performance of 1.6 points.

| | W/o disfl | With disfl | Δ(w,w/o) |
|---|---|---|---|
| LAS | 67.3% | 65.7% | +1.6 |
| UAS | 74.2% | 73.0% | +1.2 |
| LAC | 74.2% | 72.6% | +1.6 |

Table 4: Parsing scores on test sentences with disfluencies (*with*) and after manual deletion of disfluencies (*without*). Scores are computed on the whole test corpus.

### 4.2 Impact of speaking style

A marked characteristics of the ETB corpus, and more generally of broadcast news, is that it mixes professional radio speech with freer, less prepared, guest speech in interviews. While the utterances of radio announcers are prepared utterances spoken by professional journalists, guest utterances are less polished and embody unplanned speech, closer to an every day setting.

To evaluate the impact of the journalist/guest style difference, we additionally annotated each utterance with an annotation indicating whether the utterance was that of the radio announcer or of a guest. 72% of the corpus is tagged as journalist style, and 28% as guest style. For the next experiment, 60% of the whole corpus is reserved for training, and 40% for testing. This train/test division is carefully made so that the global proportion of journalist/guest speech is preserved in both the training and test corpus. The test corpus is further split into two smaller test corpora, containing respectively only journalist and guest speaker utterances.

| | Journalist | Guest | Δ(S,G) |
|---|---|---|---|
| LAS | 70.8% | 65.2% | -5.6 |
| UAS | 76.5% | 71.8% | -4.7 |
| LAC | 77.5% | 72.0% | -5.5 |

Table 5: Parsing scores of guest and journalist utterances with disfluencies kept.

As expected, the professional radio speaker style is easier to parse. Obviously though, disfluencies are more frequent in guest than in speaker speech. In order to remove the effect of disfluencies, and so only evaluate the impact of the other speaking style factors (lexicon used, syntactic structures, etc.), we manually fix the disfluencies in both parts of the corpus (speaker and guest) and redo the experiment.

| | Speaker | Guest | Δ(S,G) |
|---|---|---|---|
| LAS | 71.2% | 67.8% | -3.4 |
| UAS | 77.2% | 74.1% | -3.1 |
| LAC | 78.2% | 74.5% | -3.7 |

Table 6: Parsing scores for guest and journalist styles with disfluencies removed.

Two remarks can be made. First, correcting disfluencies improves parsing more on guest speech (+2.6%) than on radio journalist speech (+0.4%), which conforms to intuition, because disfluencies are much more frequent in guest speech than in prepared speech. Second, there is still a significant difference in parsing performance between both styles, which is not due to disfluencies but results from other factors, most probably the lexicon and syntactic patterns typical of spontaneous speech. More precisely, disfluencies explain about 40% of the additional parsing errors in guest speech, while these other factors explain 60% of these errors.

### 4.3 Impact of headers

Finally, we consider the impact on parsing of a construct typical of radio announcers, namely headline utterances (headers) which structure the news by preparing or announcing the forthcoming subject, often in a telgraphic style with missing functional elements.

This next experiment is realized in 10 fold-cross-validation because there are few “headers” in the ETB (14% of all utterances). Guest utterances are removed and training is performed on journalist utterances including both headers and normal utterances. Two tests are then realized, one on all journalist utterances and the other only on headers. This test shows that headers are much more difficult to parse, which is probably due to the unbalanced training corpus, which largely favors common, non-header utterances. This suggests that parsing of radio speech could be improved by training models that are dedicated to parsing headers and explicitly detecting this speaking style. Such an experiment, however, would require collecting enough examples of header, and thus recognizing them automatically based on their general patterns.

| | Normal | Headers | Δ(-H,+H) |
|---|---|---|---|
| LAS | 70.6% | 61.7% | -8.9 |
| UAS | 76.2% | 69.7% | -6.5 |
| LAC | 77.4% | 67.5% | -9.9 |

Table 7: Comparing performance on headers vs. common speaker utterances.

Although we have shown that headers have a clear impact within the radio speaker style, it is worth noting that their impact on the global baseline performances is not significant, because the relative proportion of headers is quite low.

# 5 Conclusion

One mid-term objective of the work presented here is the study and comparison of the impact on parsing accuracy, of specific oral constructs in broadcast news. Note however that for now, the finalized treebank is quite small and the conclusions derived from experimental results should be interpreted with care. We distinguish between the prepared speech of professional journalists and the more spontaneous speech of guest speakers, and we show that there are about 20% more parsing errors in the latter than in the former. Obviously, disfluencies occur more frequently in the more spontaneous speech, but interestingly the data shows that disfluencies only account for 40% of these additional errors. This suggests that, in order to improve spontaneous speech parsing, it is not sufficient to treat only disfluencies. Phenomena typical of this kind of speech, such as dislocations and ellipses also need to be considered and better handled. Furthermore, within the subcorpus composed of professional journalist utterances, we distinguish between normal journalistic speech and “header” constructs, whose purpose is to manage and structure the dialog and radio transitions. We show that there are about 30% more parsing errors in header constructs and that these additional errors are mainly due to two factors: the specific structures of these headers, which are often non-verbal utterances with several juxtapositions, and their relatively low number of occurrences in the corpus. Yet, it does not seem superfluous to specifically detect and improve the analysis of these relatively rare segments. Indeed, headers play an important structuring function in radio news and adequately parsing them might markedly benefit interpretation. They could be used, for instance, to infer the identity of the next or of the previous speaker in the audio stream [5].

# References

[1] A. Abeille, L. Clément, and F. Toussenel. *Building a treebank for French*. Kluwer, Dordrecht, 2003.

[2] E. Bick. Parsing and evaluating the french europarl corpus. In *Méthodes et Outils pour L'évaluation Des Analyseurs Syntaxiques (Journée ATALA)*, pages 4–9, Paris, May 2004.

[3] Didier Bourigault. Un analyseur syntaxique opérationnel : Syntex. Mémoire d'habilitation, Université Toulouse-Le Mirail, June 2007.

[4] S. Galliano, E. Geoffrois, D. Mostefa, K. Choukri, J.-F. Bonastre, and G. Gravier. The ester phase ii evaluation campaign for the rich transcription of french broadcast news. In *Proc. of the European Conf. on Speech Communication and Technology*, 2005.

[5] V. Jousse, S. Petitrenaud, S. Meignier, Y. Estève, and C. Jacquin. Automatic named identification of speakers using diarization and ASR systems. In *Proc. of ICASSP*, 2009.

[6] Candito M.-H., Crabbé B., and Denis P. Statistical french dependency parsing: treebank conversion and first results. In *Proceedings of LREC'2010*, La Valletta, Malta, 2010.

[7] M. Meteer, R. Taylor, and R. Iver MacIntyre. Dysfluency annotation stylebook for the switchboard corpus. Technical report, Distributed by LDC, 1995.

[8] Joakim Nivre. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT*, pages 149–160, 2003.

[9] Helmut Schmid. Improvements in part-of-speech tagging with an application to german. In *Proc. of the ACL SIGDAT-Workshop*, pages 47–50, Dublin, 1995.

[10] Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL 2008: Proceedings of the Twelfth Conference on Natural Language Learning*, pages 159–177, Manchester, United Kingdom, 2008.

# LFG without C-structures

Özlem Çetinoğlu[1], Jennifer Foster[1], Joakim Nivre[2],
Deirdre Hogan[1], Aoife Cahill[3], Josef van Genabith[1]

[1]Dublin City University, School of Computing
[2]Uppsala University, Department of Linguistics and Philology
[3]University of Stuttgart, IMS
E-mail: [1]{ocetinoglu,jfoster,dhogan,josef}@computing.dcu.ie
[2]joakim.nivre@lingfil.uu.se [3]aoife.cahill@ims.uni-stuttgart.de

**Abstract**

We explore the use of two dependency parsers, Malt and MST, in a Lexical Functional Grammar parsing pipeline. We compare this to the traditional LFG parsing pipeline which uses constituency parsers. We train the dependency parsers not on classical LFG f-structures but rather on modified dependency-tree versions of these in which all words in the input sentence are represented and multiple heads are removed. For the purposes of comparison, we also modify the existing CFG-based LFG parsing pipeline so that these "LFG-inspired" dependency trees are produced. We find that the differences in parsing accuracy over the various parsing architectures is small.

## 1 Introduction

Phrase structure parsing has come a long way in the last decade. Techniques such as iterative chart pruning (Charniak et al., 2006) and cell-closing (Roark and Hollingshead, 2008) have been used to speed up parsing, and discriminative reranking (Charniak and Johnson, 2005) and latent variable grammar induction (Matsuzaki et al., 2005; Petrov et al., 2006) have pushed Parseval f-scores on the standard Wall Street Journal test set over the 90% mark. In the same time period, dependency parsing has also flourished (Yamada and Matsumoto, 2003; Nivre and Nilsson, 2005; McDonald et al., 2005; Nivre et al., 2007). Dependency trees are often viewed as a more intuitive and less anglocentric way of representing syntactic phenomena, and a multilingual dependency parsing system such as Malt allows such structures to be produced in linear or at worst quadratic time (Nivre et al., 2006; Nivre, 2009). In the context of these two types of linguistic representation, Lexical-Functional Grammar (LFG) (Bresnan, 2001) is an interesting linguistic theory since it has a foot in both camps, encoding constituent structure (c-structure) using context-free constituency trees and grammatical dependencies between the main words in a sentence using directed acyclic graphs (f-structures).

In recent years there have been two main approaches within the LFG community to parsing text into LFG c-structures and f-structures: one in which handcrafted LFG grammars are used in conjunction with unification-based parsing (Kaplan et al., 2004; Maxwell and Kaplan, 1996) and another (Cahill et al., 2004, 2008) which uses treebank-based resources and an LFG f-structure Annotation Algorithm (AA) which decorates nodes in treebank or parser-output CFG trees with LFG functional equations which are then passed to a constraint solver to produce f-structures. The advantages of the treebank-based approach to LFG-parsing include substantially reduced grammar development time, high-quality, wide-coverage and robust output, and the fact that the approach can be applied to languages for which no hand-crafted wide-coverage LFG grammar is available but for which a CFG treebank exists. LFG parsing pipelines (both those based on hand-crafted and treebank resources) are CFG-based: a constituency parser produces trees and these trees carry functional annotations from which a constraint solver can produce an f-structure. Strong advances in dependency parsing in terms of both speed and accuracy, and the fact that, for some languages (e.g., Turkish (Oflazer et al., 2003)), only a dependency bank is available, raise interesting research questions: *is it possible to directly parse strings into LFG f-structures, obviating the CFG-parsing step in traditional LFG parsing architectures?*[1] *Do current dependency parsing technologies require changes in the f-structure level of representation? What are the accuracy results for direct dependency-parsing-based LFG models as compared to CFG-based pipelines?*

We attempt to answer these questions by experimenting with Malt and MST in treebank-based LFG parsing and comparing the results we obtain to those obtained using the Brown and Berkeley parsers. In the direct dependency LFG parsing pipeline, the dependency parsers are trained on LFG-inspired dependency trees, obtained by modifying the f-structures produced by the AA from the original PTB trees. In the CFG-based LFG parsing pipeline, constituency parsers are trained on PTB trees and the AA is applied to the parser output, yielding f-structures. For evaluation, these are then converted to dependency trees using the same procedure which was used to produce the training material for the dependency parsers. We document and discuss the implications of all design decisions which were applied in order to convert the AA output so that it can be used with Malt and MST.

The paper is organised as follows: in Section 2, we describe LFG in more detail and we show how the AA is applied to the output of constituency parsers to obtain LFG f-structures; Section 3 details the changes that were made to the AA-based LFG f-structure DAGs so that they could be used with parsers which directly produce dependency trees; the parsing experiments are described in Section 4; related work is discussed in Section 5; finally, Section 6 summarises the main points of the paper and suggests pointers for future work.

[1] Guo et al. (2008) explore a similar question for LFG-based generation.

## 2 LFG and Treebank-based LFG Acquisition

Lexical-Functional Grammar (Bresnan, 2001) is a constraint-based theory of grammar with minimally two levels of representation: c(onstituent)- structure and f(unctional)-structure. C-structure (CFG trees) captures language specific surface configurations such as word order and the hierachical grouping of words into phrases, while f-structure represents more abstract, language independent grammatical relations (essentially bilexical labelled dependencies with some morphological and semantic information, approximating to basic predicate-argument structures) in the form of attribute-value structures or directed acyclic graphs (DAGs). F-structures are defined in terms of equations annotated to nodes in c-structure (grammar rules).

Treebank-based LFG acquisition was originally developed for English (Cahill et al., 2004, 2008) and is based on an f-structure annotation algorithm pipeline. The pipeline has four main components: a constituency parser creates PTB style trees. The function labeller FunTag (Chrupała et al., 2007) enriches the parser output trees with the PTB function labels. This component is optional. The core annotation algorithm annotates trees with f-structure equations, which are read off the tree and passed on to a constraint solver producing a proto f-structure for the given sentence. Finally, the long-distance dependency component resolves non-local dependencies using automatically acquired subcategorisation frames and finite approximations of functional-uncertainty equations (Cahill et al., 2004).

## 3 LFG-Inspired Dependencies

The LFG AA takes PTB style trees and generates LFG f-structures. In order to use the output of the AA to train the dependency parser, we convert the LFG f-structure DAGs to dependency trees in the CoNLL format. LFG f-structures are recursive attribute-value structures close to but not exactly the same as the bilexical dependencies assumed in CoNLL format: LFG f-structures are somewhat more abstract and, unlike for CoNLL-style dependencies, not every token in a string is represented as a node (i.e. as the value of a PRED attribute) in the f-structure (e.g. auxiliary sequences in languages with analytic tense are represented in terms of abstract tense/aspect features). Words are represented as lemmas rather than surface forms and properties of strings (tense, mood, statement-type, person etc.) are encoded in terms of features. Non-local dependencies are represented in terms of coindexation (reentracies in the graph) and dependencies can be multi-headed.

In order to use the output of the AA to directly train Malt and MST, we convert the LFG f-structures to dependency trees by carrying out the following modifications:*i)* representing each token in the f-structure *ii)* removing dependencies that result in multiple heads *iii)* avoiding multiple roots.

**Representing each token in the f-structure:** in LFG some words map to atomic-valued features in the f-structures, rather than semantic forms. Moreover, punctua-

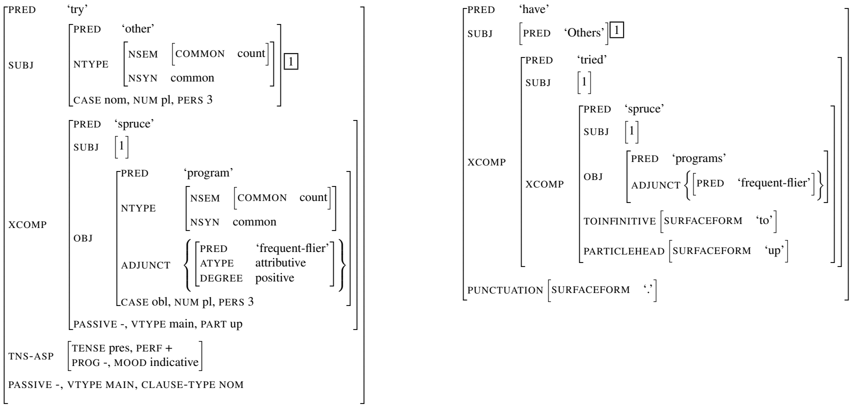


Figure 1: LFG f-structure for *Others have tried to spruce up frequent-flier programs.* in the original form (left) and as bilexical dependencies (right)

tion is not explicitly represented. In the CoNLL format, every token of a sentence and its dependency relation has to be explicitly stated. In order to obtain a bilexical dependency for every token in the input, we modify the AA to turn f-structures into full bilexical dependencies, making sure that every token in the sentence is explicitly represented as a PRED (or SURFACEFORM)-valued node. for the sentence *Others have tried to spruce up frequent-flier programs.* The original AA produces the f-structure on the left in Figure 1. The auxiliary *have* is not explicitly represented in the f-structure but contributes to the values of the TENSE, PERF and MOOD features. Similarly, the particle *up* does not have a PRED but is represented as a feature of the verb *spruce*. The modified f-structure for the same sentence is given on the right. The converted f-structure has an explicit representation for the auxiliary *have*, the verbal particle *up*, the infinitival marker *to* as well as for the punctuation marker. Atomic-valued features like CASE, NUM are removed. The PRED features now represent the surface form of the tokens rather than word lemmas.

**Removing dependencies that result in multiple heads:** For many non-local dependencies such as wh-elements in relative clauses, topicalisation, and subject/object control, LFG (and the LFG AA) assigns multiple heads to a word. In the example on the right in Figure 1 *others* is the subject of both *have*, *tried* and *spruce up*. Multiple heads are not supported in the dependency tree representations used by, for example, Malt.

In order to avoid multiple heads we follow two simple strategies. If the multi-headed construction involves a discourse function (TOPIC, TOPICREL, FOCUS), we remove this dependency, but (crucially) keep the dependency to the local head to capture the non-local dependency. Otherwise we keep the dependency relation with the head at the outermost level of the f-structure and remove the other depen-

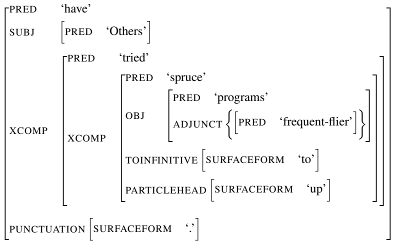

PRED 'have'
SUBJ [PRED 'Others']
XCOMP
PRED 'tried'
XCOMP
PRED 'spruce'
OBJ
PRED 'programs'
ADJUNCT {[PRED 'frequent-flier']}
TOINFINITIVE [SURFACEFORM 'to']
PARTICLEHEAD [SURFACEFORM 'up']
PUNCTUATION [SURFACEFORM '.']

Figure 2: F-structure as bilexical dependencies with multiple heads removed

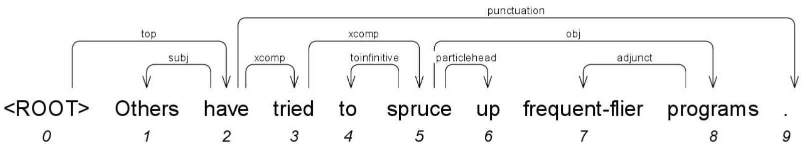


Figure 3: Dependency representation of the example sentence

dencies, as e.g. for cases of control. Our intention is to keep more informative dependencies and ones that cause less non-projectivity.

The f-structure on the right in Figure 1 contains a multi-headed dependency of the latter type. We keep the relation between *others* and *have*, and remove the other dependencies. The resulting simplified f-structure is given in Figure 2. The corresponding dependency representation for this f-structure is given in Figure 3.

**Avoiding multiple roots:** In the current version of the LFG-converted training data, there are tokens without heads, arising from inadequacies in the LFG AA or the f-structure-to-dependency-tree conversion procedure (detailed above). This leads to f-structure fragments and corresponding multiple roots in the dependency trees for a particular string. We automatically make heads of such fragments dependent on a dummy root node ROOT with a dummy label `dep`.

In the LFG-inspired dependency conversion, we exclude PTB trees with FRAG-(ment) constituents from the training data, as the LFG AA is not designed to deal with such structures. For the phrases marked as X (unknown, uncertain, or unbracketable) in the PTB, we use the dummy dependency relation `dep`. In some cases, the AA produces f-structures for only a small fragment of the tree and there are no explicit dependency relations for the remaining words in the sentence. Since we cannot obtain dependency relations for all tokens, we omit these sentences. At this stage, the number of trees/f-structures in Sections 02-21 of the PTB correctly converted to dependency trees is 39,163, i.e. 99.51% of the standard training set. The work on the LFG dependency conversion is ongoing with the objective to

|  | **Multi-head** | **Single-head** |
|---|---|---|
| # sentences | 39132 | 39163 |
| # dependency types | 25 | 25 |
| multi-headed dependencies | 7% | 0% |
| non-projective dependencies | 4.16% | 0.15% |
| non-projective sentences | 63.76% | 2.52% |
| head left of modifier | 51% | 53% |

Table 1: WSJ sections 02-21 conversion statistics for LFG bilexical f-structures with multi-headed and single-headed dependencies

|  |  |  |  |
|---|---|---|---|
| adjunct | adjunct | poss | possesive |
| app | apposition | *possmarker | possesive marker *'s* |
| comp | complement | *prepositionhead | used for *so that*, *so as to*, *as if*,... these are MWEs in LFG |
| coord | coordination item | *punctuation | punctuation |
| *dep | dependency (dummy) | quant | quantifier |
| det | determiner | relmod | relative modifier |
| focus | focus | subj | subject |
| obj | object | *toinfinitive | to infinitive |
| obj2 | 2nd object (obj-th in LFG) | *top | top (root of a dependency tree) |
| obl | oblique object | topic | topic |
| *obl2 | 2nd oblique object | topicrel | relative topic |
| obl-ag | oblique agent | xcomp | open complement |
| *particlehead | head of a particle |  |  |

Table 2: LFG-inspired conversion tagset

eventually cover the whole training set.

The training set has different characteristics when multi-headed dependencies are allowed and when multi-heads are removed to obtain single-headed dependencies. The differences can be observed in Table 1. The percentage of multi-headed dependencies is 7% and when they are removed, the non-projective dependencies decrease to 0.15% from 4.16%. This has a drastic effect on the number of sentences with at least one non-projective dependency, which drops down to 2.52% from 63.76%. The LFG-inspired conversion tagset consists of 25 dependencies. Table 2 lists those dependencies with their descriptions. The tagset is based on core LFG grammatical functions present in the original LFG DAGs. The labels marked with an asterisks indicate the additional dependencies that are not originally present in the LFG theory. The additional labels are used when the more abstract f-structure DAGs are modified to represent more surfacy bilexical dependencies in dependency trees that cover all tokens in the input strings.

# 4 Parsing Experiments

In this section, we describe our LFG parsing experiments. The four parsers we use are described in Section 4.1, the experimental procedure is detailed in Section 4.2 and the results are presented in Section 4.3.

## 4.1 Parsers

**Berkeley:** The Berkeley parser (Petrov et al., 2006) is a generative constituency parser which parses using an unlexicalised yet fine-grained smoothed PCFG. This PCFG is obtained in an iterative process of splitting the treebank non-terminals into subcategories, estimating the parameters of the resulting grammar using Expectation Maximisation and merging the less useful category splits. For efficient parsing, multi-stage coarse-to-fine pruning using the intermediate grammars obtained during training is carried out (Petrov and Klein, 2007). We train a grammar using 6 split-merge cycles and we run the parser in *accurate* mode, meaning that the pruning thresholds are tuned for accuracy at the expense of speed.

**Brown:** The Brown parser (Charniak, 2000) is a generative constituency parser which uses a head-lexicalised smoothed PCFG which is conditioned on the parse history and which combines five probability models fine-tuned for English. In our experiments, we use both this parser and the reranking version in which the n-best list returned by the generative parser is re-ordered using a discriminative reranker trained on features which are unavailable to the original parser (Charniak and Johnson, 2005). We employ these parsers in their *out-of-the-box* settings.

**MaltParser:** MaltParser is a flexible multi-lingual dependency parsing system (Nivre et al., 2006). During training a classifier is induced to predict a parsing action at a particular parsing configuration using information from the parse history and the remaining input string. During parsing, the classifier is used to drive the deterministic construction of a dependency tree. MaltParser can be used with several parsing algorithms including variants of shift-reduce parsing. We use the *stacklazy* algorithm, which employs a swap operation so that non-projective structures can be handled (Nivre, 2009). Following Attardi and Ciaramita (2007) we train a linear classifier where the feature interactions are modelled explicitly.

**MSTParser:** While MaltParser learns to predict parsing actions, MST (McDonald et al., 2005), learns to predict entire dependency trees. The parser finds the maximum spanning tree in a multi-digraph using one of several algorithms described in McDonald (2006). For our experiments, we use the second-order approximate non-projective parsing model introduced in McDonald and Pereira (2006), which parameterizes dependency trees by pairs of adjacent sibling arcs and uses hill-climbing to find the highest scoring (possibly non-projective) tree, starting from the highest-scoring projective tree derived by dynamic programming (Eisner, 1996). MSTParser can be run in one or two stages. In the two-stage model, an unlabelled tree is predicted and the labeling of dependency arcs is carried out during the second stage. We employ the one-stage parser which directly predicts labels.

## 4.2 Procedure

In the **LFG constituency parsing pipeline**, *i)* The constituency parsers are trained in the usual way on the PTB (function tags and traces are excluded from the training trees). *ii)* Input sentences are parsed with the parsers. *iii)* The parse trees are passed through the FunTag labeller (Chrupala et al., 2007) which assigns Penn-II treebank function tags to raw CFG parser output trees. We also carry out the experiment with this step omitted. *iv)* The LFG AA is applied to the constituency trees producing LFG f-structures. *v)* Parser output f-structures are converted to dependency trees. *vi)* The output is evaluated against the LFG-style dependency trees for WSJ Sections 00 and 23.

In the **LFG dependency parsing pipeline**, *i)* The AA is applied to the PTB training data producing f-structures. *ii)* The f-structures are converted into dependency trees. *iii)* The dependency parsers are trained on the LFG-inspired dependency trees. *iv)* Input sentences are parsed with the dependency parsers. *v)* The output is evaluated against the LFG-style dependency trees for Sections 00 and 23.

Our training data consists of Sections 02-21 of the WSJ section of the PTB. We use Section 00 as our development set to tune the MaltParser feature model and to perform error analysis, and present final results on Section 23. We experiment with the use of gold POS tags, POS tags obtained using a POS tagger (Giménez and Màrquez, 2004) and, for Brown and Berkeley, POS tags produced by the parsers themselves.[2] We use the CoNLL evaluation metrics of labelled attachment score (LAS) and unlabelled attachment score (UAS).[3]

### 4.3 Results

Evaluation results on Section 00 and on Section 23 are given in Table 3. We observe that using FunTag leads to a 3% increase for constituency parsers. The Brown+Reranker+FT architecture has the highest scores, outperforming even the gold-POS-tagged systems. Constituency parsers with FunTag rank higher than dependency parsers, but differences between the systems are small. The trends for Section 00 carry over to Section 23. The main difference is that the two dependency parsers, Malt and MST, suffer a greater drop in accuracy when using predicted rather than gold POS tags.

### 4.4 Discussion

We examine Section 00 accuracy scores broken down by dependency type for all parsing systems by picking the best non-gold-POS-tagged architecture for each parser. All four systems have over 95% scores for subjects and objects. The performance on adjuncts is almost the same ($\sim$ 88%) in all systems. MST outperforms all other systems in identifying thematic objects (`obj2`), followed by Malt. Constituency parsers suffer on `obj2` with scores lower than 50%. Coordination f-score for constituency parsers is 85% while dependency parsers perform slightly worse with a 80% f-score. The accuracy of all parsers drop down to the 70%-80% range

[2] Note that the Brown parsers always perform their own POS tagging.

[3] For replicability, we provide all experimental settings at http://www.nclt.dcu.ie/gramlab/experiments.html

|  |  | Section 00 |  | Section 23 |  |
|---|---|---|---|---|---|
| **POS** | **Parser** | **LAS** | **UAS** | **LAS** | **UAS** |
| Gold | Berkeley | 87.38 | 91.71 | 87.63 | 91.48 |
|  | Berkeley+FT | **90.51** | **92.10** | **90.81** | **92.27** |
|  | Malt | 89.02 | 90.64 | 89.01 | 90.59 |
|  | MST | 89.79 | 91.68 | 89.97 | 91.73 |
| OwnTag | Berkeley | 86.82 | 91.38 | 87.12 | 91.19 |
|  | Berkeley+FT | 89.97 | 91.8 | 90.29 | 91.95 |
|  | Brown | 86.30 | 90.93 | 86.56 | 90.81 |
|  | Brown+FT | 89.54 | 91.3 | 89.75 | 91.48 |
|  | Brown+Reranker | 87.55 | 92.24 | 87.72 | 92.04 |
|  | Brown+Reranker+FT | **90.81**$^{1}$ | **92.59**$^{3}$ | **91.13**$^{*}$ | **92.81**$^{*}$ |
| Predicted | Berkeley | 86.97 | 91.42 | 86.51 | 90.60 |
|  | Berkeley+FT | 90.11$^{2}$ | 91.83$^{4}$ | 89.61$^{*}$ | 91.33$^{*}$ |
|  | Malt | 88.66$^{*}$ | 90.41$^{*}$ | 87.57$^{*}$ | 89.47$^{*}$ |
|  | MST | 89.43$^{*}$ | 91.47$^{*}$ | 88.51$^{*}$ | 90.59$^{*}$ |

Table 3: Accuracy scores on Sections 00 and 23 for the LFG-inspired conversion (*p-value (1)&(2)=0.004; p-value (3)&(4)=0.002; For all other comparisons marked with *, p-value $\ll$ 0.001*)

for relative modifiers. The dummy `dep` relation is rarely used by the LFG-inspired conversion and only identified by Brown+Reranker+FT albeit with very low scores.

The breakdown shows that thematic objects, relative clauses and coordination are harder to recover regardless of the parsing system. Since the `dep` relation does not represent a specific linguistic phenomenon, it is not possible to identify it either for constituency parsers or dependency parsers. The small difference between the constituency and dependency parsing results is promising for applications that can sacrifice some accuracy for speed[4]. Another important issue is the granularity of the information represented. The LFG-inspired dependencies omit almost half of the features of the original LFG f-structures during the conversion. Some of those features, such as morphological information can easily be incorporated into CoNLL trees when needed. But information lost on functional relations such as non-local dependencies are potentially harder to recover from dependency trees. In future work, we will explore an approach that allows us to recover such functions after parsing.

# 5 Related Work

Early research on LFG without full-fledged c-structures is presented in (Frank, 2003) and (Schneider, 2005). Both project f-structures (or f-structure like dependencies) from chunks rather than full CFG trees.

Recent research which attempts to combine ideas from LFG and data-driven dependency parsing is described in Schluter and van Genabith (2009) as well as Øvrelid et al. (2009). Schluter and van Genabith (2009) are concerned with French parsing: they convert LFG f-structures into pseudo-projective dependencies and

[4] The Berkeley parser (run in *accurate* mode) takes 10m25s to parse Section 23 compared to 3m17s for MaltParser on the same machine.

train both MSTParser and MaltParser on these. They conclude that the use of a dependency parser is a reasonable alternative to a c-structure parser in an LFG parsing pipeline. The approach of Øvrelid et al. (2009) is quite different: in contrast to our approach they do not produce LFG-inspired dependency trees but adapt the feature set used by MaltParser's parser-action classifier so that it includes features obtained from the hand-crafted ParGram English and German LFGs to improve parser output in the format of CoNLL-style dependency trees.

## 6 Concluding Remarks

LFG f-structures are close to but not the same as the bilexical dependencies used in dependency parsers and we have shown how f-structures can be systematically converted into bilexical dependencies for use in direct parsing into f-structure, obviating the c-structure component in classical LFG parsing. This makes theoretically motivated abstract LFG dependency representations available to the dependency parsing community and fast dependency parsing technology available to the LFG community. This does not come without a price however. A number of pragmatic decisions needed to be made in order to allow existing dependency parsers to be trained, and these decisions include deciding what information can be lost in moving from graphs to trees. How important the lost information is to downstream applications remains an open question. We plan to investigate this, to compare our LFG-inspired dependency scheme to other dependency schemes such as the Stanford dependencies (de Marneffe and Manning, 2008), the Pennconverter dependencies (Johansson and Nugues, 2007) and Tésniére-inspired dependencies (Sangati and Mazza, 2009), and to experiment with dependency parsers which can handle multiple-headed constructions (Sagae and Tsujii, 2008).

### Acknowledgements

This research is partly supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University, School of Computing.

## References

Attardi, Giuseppi and Massimiliano Ciaramita (2007). Tree Revision Learning for Dependency Parsing. In *Proceedings of HLT-NAACL-07*. Rochester, NY.

Bresnan, Joan (2001). *Lexical-Functional Syntax*. Oxford: Blackwell Publishers.

Cahill, Aoife, Michael Burke, Ruth O'Donovan, Stefan Riezler, Josef van Genabith and Andy Way (2008). Wide-Coverage Deep Statistical Parsing using Automatic Dependency Structure Annotation. *Computational Linguistics* 34(1).

Cahill, Aoife, Michael Burke, Ruth O'Donovan, Josef van Genabith and Andy Way (2004). Long-Distance Dependency Resolution in Automatically Acquired Wide-Coverage PCFG-Based LFG Approximations. In *Proceedings of ACL-04*.

Charniak, Eugene (2000). A Maximum-Entropy-Inspired Parser. In *Proceedings of NAACL-00*. Seattle.

Charniak, Eugene and Mark Johnson (2005). Course-to-fine n-best-parsing and MaxEnt discriminative reranking. In *Proceedings of ACL-05*. Ann Arbor.

Charniak, Eugene, Mark Johnson et al. (2006). Multilevel Coarse-to-fine PCFG Parsing. In *Proceedings of HLT-NAACL-06*. New York.

Chrupała, Grzegorz, Nicolas Stroppa, Josef van Genabith and Georgiana Dinu (2007). Better training for function labeling. In *RANLP 2007*. Bulgaria.

de Marneffe, Marie-Catherine and Christopher D. Manning (2008). The Stanford typed dependencies representation. In *Proceedings of the COLING Workshop on Cross-Framework and Cross-Domain Parser Evaluation*. Manchester.

Eisner, Jason (1996). Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of COLING-96*. Denmark.

Frank, Anette (2003). Projecting F-structures from Chunks. In *Proceedings of LFG-03*. Albany, NY.

Giménez, Jesús and Lluís Màrquez (2004). SVMTool: A general POS tagger generator based on Support Vector Machines. In *Proceedings of LREC-04*. Lisbon.

Guo, Yuqing, Haifeng Wang and Josef van Genabith (2008). Accurate and robust LFG-based generation for Chinese. In *Proceedings of INLG-08*. Salt Fork.

Johansson, Richard and Pierre Nugues (2007). Extended Constituent-to-Dependency Conversion for English. In *Proceedings of NODALIDA-07*.

Kaplan, Ron, Stefan Riezler, Tracy King, John Maxwell, Alexander Vasserman and Richard Crouch (2004). Speed and Accuracy in Shallow and Deep Stochastic Parsing. In *Proceedings HLT-NAACL-04*. Boston, MA.

Matsuzaki, Takuya, Yusuke Miyao and Jun'ichi Tsujii (2005). Probabilistic CFG with latent annotations. In *Proceedings of ACL-05*. Ann Arbor.

Maxwell, John and Ron Kaplan (1996). An Efficient Parser for LFG. In *Proceedings of LFG-96*. Grenoble.

McDonald, Ryan (2006). Discriminative Learning and Spanning Tree Algorithms for Dependency Parsing. Ph.D. thesis, University of Pennsylvania.

McDonald, Ryan, Koby Crammer and Fernando Pereira (2005). Online Large-Margin Training of Dependency Parsers. In *Proceedings of ACL-05*. Ann Arbor.

McDonald, Ryan and Fernando Pereira (2006). Online Learning of Approximate Dependency Parsing Algorithms. In *Proceedings of EACL-06*.

Nivre, Joakim (2009). Non-Projective Dependency Parsing in Expected Linear Time. In *Proceedings of ACL-AFNLP-09*. Singapore.

Nivre, Joakim, Johan Hall, Sandra Kübler, Ryan Mac Donald, Jens Nilsson, Sebastian Riedel and Deniz Yuret (2007). The CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of EMNLP-CoNLL-07*. Prague.

Nivre, Joakim, Johan Hall and Jens Nilsson (2006). MaltParser: A Data-Driven Parser-Generator for Dependency Parsing. In *Proceedings LREC-06*.

Nivre, Joakim and Jens Nilsson (2005). Pseudo-Projective Dependency Parsing. In *Proceedings of ACL-05*. Ann Arbor.

Oflazer, Kemal, Bilge Say, Zeynep Hakkani-Tür and Gökhan Tür (2003). Building a Turkish treebank. In *Treebanks: Building and Using Parsed Corpora*, Kluwer.

Øvrelid, Lilja, Jonas Kuhn and Kathrin Spreyer (2009). Improving data-driven dependency parsing using large-scale LFG grammars. In *Proceedings of ACL-AFNLP-09*. Singapore.

Petrov, Slav, Leon Barrett, Romain Thibaux and Dan Klein (2006). Learning Accurate, Compact and Interpretable Tree Annotation. In *Proceedings of ACL-06*.

Petrov, Slav and Dan Klein (2007). Improved Inference for Unlexicalized Parsing. In *Proceedings of HLT-NAACL-07*. Rochester, NY.

Roark, Brian and Kristy Hollingshead (2008). Classifying chart cells for quadratic complexity context-free inference. In *Proceedings of COLING-08*. Manchester.

Sagae, Kenji and Jun'ichi Tsujii (2008). Shift-Reduce Dependency DAG Parsing. In *Proceedings of COLING-08*. Manchester.

Sangati, Federico and Chiara Mazza (2009). An English Dependency Treebank á la Tesniére. In *Proceedings of TLT8*. Milan.

Schluter, Natalie and Josef van Genabith (2009). Dependency Parsing Resources for French: Converting Acquired Lexical Functional Grammar F-Structure Annotations and Parsing F-Structures Directly. In *Proceedings of NODALIDA 2009*. Odense.

Schneider, Gerold (2005). A Broad-Coverage, Representationally Minimal LFG Parser: Chunks and F-structures are Sufficient. In *Proceedings of LFG-05*. Bergen, Norway.

Yamada, Hiroyasu and Yuji Matsumoto (2003). Statistical Dependency Analysis with Support Vector Machines. In *Proceedings of IWPT-03*. France.

# Robust Constituent-to-Dependency Conversion for English

Jinho D. Choi and Martha Palmer

Institute of Cognitive Science
University of Colorado at Boulder
E-mail: choijd@colorado.edu, mpalmer@colorado.edu

**Abstract**

This paper suggests a robust way of converting constituent-based trees in the Penn Treebank style into dependency trees for several different English corpora. For English, there already exist conversion tools. However, these tools are often customized enough for a specific corpus that they do not necessarily work as well when applied to different corpora involving newly introduced POS-tags or annotation schemes. The desire to improve conversion portability motivated us to build a new conversion tool that would produce more robust results across different corpora. In particular, we have modified the treatment of head-percolation rules, function tags, coordination, gapping, and empty category mappings. We compare our method with the LTH conversion tool used for the CoNLL'07-09 shared tasks. For our experiments, we use 6 different English corpora from OntoNotes release 4.0. To demonstrate the impact our approach has on parsing, we train and test two state-of-the-art dependency parsers, MaltParser and MSTParser, and our own parser, ClearParser, using converted output from both the LTH tool and our method. Our results show that our method removes certain unnecessary non-projective dependencies and generates fewer unclassified dependencies. All three parsers give higher parsing accuracies on average across these corpora using data generated by our method; especially on semantic dependencies.

## 1 Introduction

There has been growing interest in statistical dependency parsing. For those who need to parse, in addition to newswire, a large amount of less structured text (e.g., web-blogs or automatic translations), dependency parsing has advantages over constituent-based parsing because it is simple and fast, yet gives useful information (Shen et al. [19], Cui et al. [3]). Moreover, since dependency structure is not constrained by word-order, it is considered to be more domain or language independent than phrase structure.

Most current state-of-art dependency parsers use a supervised learning approach (McDonald et al. [13], Nivre et al. [16]), which usually requires a large amount of annotated data. For English, there are some manually annotated dependency Treebanks available (Rambow et al. [18], Čmejrek et al. [20]); nonetheless, constituent-based Treebanks such as the Penn Treebank (Marcus et al. [11]) are more dominant. It has been shown that these Penn Treebank style constituent-based trees can reliably be converted into dependency trees using heuristics (Johansson and Nugues [9]). The result of such a conversion is that statistical dependency parsers have access to larger amounts of annotated data in dependency structure.

There already exist tools that convert phrase structure to dependency structure. The most popular one is the LTH constituent-to-dependency conversion tool[1] that had been used for the CoNLL'07-09 shared tasks (Hajič et al. [6]) and gave useful results. The LTH tool makes several improvements over its predecessor, Penn2Malt[2]: it adds semantic dependencies extracted from function tags in the Penn Treebank (e.g,. LOC, TMP; Marcus et al. [10]) and remaps dependencies related to empty categories, producing non-projective dependencies. Although the tool works well in many ways, it is somewhat customized to the Penn Treebank (mainly for the Wall Street Journal corpus), so it does not necessarily work as well when applied to different corpora. We tested the LTH tool on the OntoNotes English data (Hovy et al. [7]). These corpora contain part-of-speech tags not introduced in the original Penn Treebank (e.g., EDITED, META) and show occasional departures from the original guidelines (e.g., inserting NML phrases, separating hyphenated words). Unfortunately, these new formats affect the LTH tool's ability to find correct dependency relations, motivating us to aim for a more resiliant approach.

In this paper we present a robust method for doing constituent-to-dependency conversion across different corpora. In particular, we show improvements due to modifications of head-percolation rules, function tags, coordination, gapping relations, and empty category mappings. For our experiments, we use 6 different English corpora from the latest release of the OntoNotes Treebank. To demonstrate the advantages of our approach for dependency parsing, we train and test two state-of-the-art dependency parsers, MaltParser and MSTParser, and our own parser, ClearParser, using converted output from both the LTH tool and our method. Our results show that our method removes certain unnecessary non-projective dependencies and generates fewer unclassified dependencies. Moreover, all three parsers give higher parsing accuracies on average across these corpora using data generated by our method; especially on semantic dependencies. The improvement in parsing accuracy is even more significant when parsing models are tested on corpora different from their training corpora, leading us to believe that our method also gives more robust results across different corpora.

[1] The LTH tool: http://nlp.cs.lth.se/software/treebank_converter/

[2] Penn2Malt: http://stp.lingfil.uu.se/ nivre/research/Penn2Malt.html

# 2 Improved constituent-to-dependency conversion

Our work was inspired by Johansson [8, Chap. 4], which gives descriptive explanations about how the LTH tool converts Penn Treebank style constituent-based trees to CoNLL style dependency trees. We carefully followed their steps and modified certain heuristics to generate more robust output. This section describes some of the key changes we made to Johansson's approach.

## 2.1 Head-percolation rules

| | | |
|---|---|---|
| ADJP | r | JJ*\|VB*\|NN*\|ADJP;IN;RB\|ADVP;CD\|QP;FW\|NP;* |
| ADVP | r | VB*;RB\|JJ*;RB+;ADJP;ADVP;QP;IN;NN;CD;RP;NP;* |
| CONJP | l | CC;TO;IN;VB;* |
| EDITED | r | VB*\|VP;NN*\|PRP\|NP;IN\|PP;S*;* |
| FRAG | l | NN*\|NP;W*;S;SBAR;IN\|PP;JJ\|ADJP;RB\|ADVP;* |
| INTJ | l | VB;NN*;UH;INTJ;* |
| LST | l | LS;NN;CD;* |
| META | r | VP;NP;* |
| NAC | r | NN*;NP;S;SBAR;* |
| NML | r | NN*\|NML;CD\|NP\|QP\|JJ*\|VB*;* |
| NP | r | NN*\|NML;NX;PRP;FW;CD;NP\|QP\|JJ*\|VB*;ADJP;S;SBAR;* |
| NX | r | NN*;NX;NP;* |
| PP | l | TO;IN;VBG\|VBN;RP;PP;NN*;JJ;RB;* |
| PRN | r | * |
| PRT | l | RP;PRT;* |
| QP | l | JJR\|RBR;JJS\|RBS;CD;NN*;PDT\|DT;ADVP;JJ;* |
| RRC | l | VBG\|VBN;VP;NP\|NN*;ADJP;ADVP;PP;* |
| S | r | TO;MD;VB*;VP;*-SBJ;*-TPC;*-PRD;S\|SINV\|S*Q;SBAR;NP;PP;* |
| SBAR | r | IN\|TO;DT;MD;VB*;VP;*-PRD;S\|SINV\|S*Q;SBAR;* |
| SBARQ | r | MD;VB*;VP;S*Q;S\|SINV;* |
| SINV | r | MD;VB*;VP;*-SBJ;*-TPC;*-PRD;S\|SINV;NP;* |
| SQ | r | MD;VB*;VP;*-PRD;SQ;S;* |
| UCP | l | * |
| VP | l | TO;MD;VB*;VP;*-SBJ;*-TPC;*-PRD;NN;NNS;NP;QP;JJ*;ADJP;* |
| WHADJP | r | JJ*\|VBN\|VBG;ADJP;* |
| WHADVP | l | WRB;WHADVP;WDT;RB;* |
| WHNP | r | NN*\|NML;CD;VBG;NP;JJ*;WP;QP;WHNP;WHADJP;* |
| WHPP | l | IN\|TO;* |
| X | r | * |

Table 1: New head-percolation rules. l/r implies to look for the leftmost/rightmost item. */+ implies 0/1 or more characters and *-TAG implies any POS tag with the function tag. | implies a logical OR and ; is a delimiter between POS tags. Each rule gives higher precedence to the left (e.g., TO takes the highest precedence in PP).

We use head-percolation rules (from now on, headrules) to find the head of each constituent in phrase structure. Although there are other headrules available (Yamada and Matsumoto [21]), we designed our own (Table 1) for two reasons. First, previous rules could not handle POS tags not included in the original Penn Treebank (e.g., EDITED, NML). Second, we found it useful to make more use of function tags; Johansson used one function tag for his rules, PRD (predicative), whereas we used two additional tags, SBJ (subject) and TPC (topic) (e.g., Figure 1). Furthermore, we made minor modifications to some rules. For example, we changed the rule for ADJP (adjective phrase) such that adjectives now get higher priority than nouns.

## 2.2 Small clauses

Figure 1 shows a constituent-based tree (left) with a small clause, *us happy*, a dependency tree generated by the LTH tool (right-top), and one generated by our approach (right-bottom). According to our headrules, NP-SBJ becomes the head of ADJP-PRD in S-1. However, the LTH tool treats ADJP-PRD as an object predicative (OPRD) of VBP. Both approaches are valid; we take this approach because we find it easier to integrate this structure with a semantic corpus like PropBank (Palmer et al. [17]): *us happy* is annotated as a single argument of *made* in the PropBank, and by making *happy* a child of *us*, we can simply treat the subtree of *us* as an argument of *made* (ARG$_1$), whereas the treatment gets more complicated when *happy* is also a child of *made*.

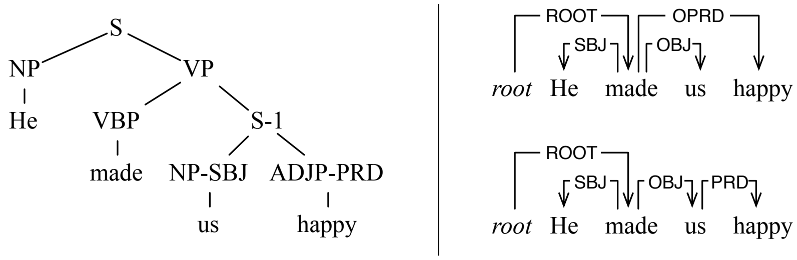


Figure 1: Small clause example.

## 2.3 Function tags

We use 14 function tags to generate dependency labels. Some of them are joined together (e.g., LOC-TMP); the LTH tool converts each joined function tag into a single dependency label. However, most statistical parsers do not often find joined tags correctly (cf. Table 9), so it may be better to select just one of the tags from the joined tag pair. For example, if LOC and TMP are joined (LOC-TMP), we keep only LOC as a dependency label.[3] We follow the precedence table described below when choosing which tag to keep. There are cases where that the choice becomes difficult; however, such cases are rare enough that they can be ignored.

```
DTV|EXT|LGS|SBJ > LOC > BNF|DIR|MNR|PRP|TMP > SEZ|VOC > PRD > ADV
IGNORE ::= CLF|CLR|ETC|HLN|IMP|NOM|PUT|TPC|TTL|UNF
```

Table 2: Function tag precedences.

[3] In a sentence, [ADVP There] [VP goes [PP *ICH*-1]] [NP all your payments] [PP-1 down in the toilet], [ADVP There] is marked with a joined tag, LOC-TMP.

Table 2 shows how we set the precedences for selecting a member of a joined function tag. For example, SBJ takes precedence over LOC, which again takes precedence over MNR and so on. IGNORE shows a list of function tags that we do not use as dependency labels. There are several reasons for this; mainly, we keep only tags with more semantic values, which usually appear as modifiers in the PropBank.

## 2.4 Coordination

We take a right branching approach for coordination (the left conjunct becomes the head of the conjunction, which becomes the head of the right conjunct). It sometimes gets hard to decide whether or not a phrase contains coordination. We consider a phrase contains coordination if it is tagged as an unlike coordinated phrase (UCP), if it contains a child annotated with a function tag ETC (et cetera), or if it contains at least one conjunction (CC) or a conjunction phrase (CONJP). Even if there is a conjunction, if either the left or the right conjunct does not appear within the same phrase, we do not consider there to be a coordination.

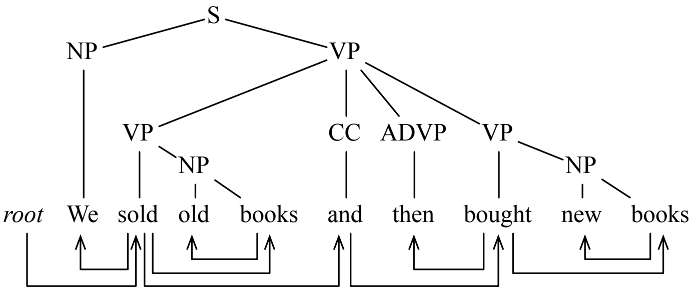


Figure 2: Coordination example.

Within a coordinated phrase, we use commas and semicolons as separators (similar to Johansson's). In addition to Johansson's approach, we apply the following heuristics to check if the left and right conjuncts have the same phrasal type. SKIP shows a list of POS tags that can be skipped to find the correct conjuncts (e.g., ADVP in Figure 2).

```
NounLike  ::= NN*|PRP|NML|NP|WHNP|*-NOM
AdjLike   ::= JJ*|ADJP
WhAdvLike ::= WHADVP|WRB|WHPP|IN
SKIP      ::= PRN|INTJ|EDITED|META|CODE|ADVP|SBAR

if ((parent.pos == UCP) || (left.pos == right.pos) || (left.tag == ETC) ||
    (left.pos == NounLike && right.pos == NounLike) ||
    (left.pos == AdjLike  && right.pos == AdjLike) ||
    (parent.pos == WHADVP && (left.pos == WhAdvLike && right.pos == WhAdvLike))
     return true;
else return false;
```

### 2.5 Gapping relations

Most statistical parsers perform poorly on gapping relations because it is hard to distinguish them from coordination, and they do not appear frequently enough to be trained on. Johansson's approach usually generates correct dependencies for gapping relations; however, it sometimes produces a very flat structure with many long-distance dependencies. The top tree in Figure 3 shows how gapping relations are handled by the LTH tool.

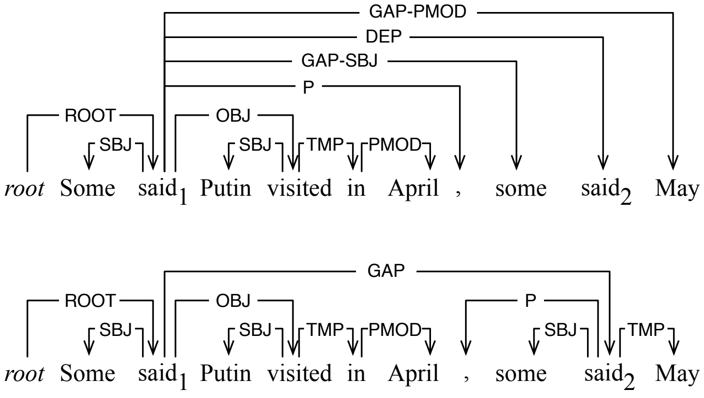

GAP-PMOD
DEP
GAP-SBJ
P
ROOT
OBJ
SBJ
SBJ
TMP
PMOD
root Some said1 Putin visited in April , some said2 May
GAP
ROOT
OBJ
P
SBJ
SBJ
TMP
PMOD
SBJ
TMP
root Some said1 Putin visited in April , some said2 May

Figure 3: Gapping relation example.

As illustrated, *comma*, *some*, *said*$_2$, and *May* become children of *said*$_1$. Although this is an accurate representation, automatic parsers almost never find these dependencies correctly. The bottom tree in Figure 3 shows our approach of handling gapping relations. In our case, *comma*, *some*, and *May* become children of *said*$_2$, which then becomes a child of *said*$_1$. This way, parsers can easily learn the local information, yet still can recover the original representation by using the gapping relation (GAP).

### 2.6 Empty category mappings

Our approach to empty category mappings is similar to Johansson's, except for our handling of *RNR* (right node raising; see Figure 4). *RNR* appears in coordinated phrases such that there are always at least two *RNR* nodes mapped to the same antecedent. In this case, we map the antecedent to its closest *RNR* as illustrated in Figure 4. This way, we can eliminate many non-projective dependencies without sacrificing loss of semantic interpretation.

## 3 Experiments

We evaluate the changes we have made to the conversion process by training several parsers and testing them on various corpora.

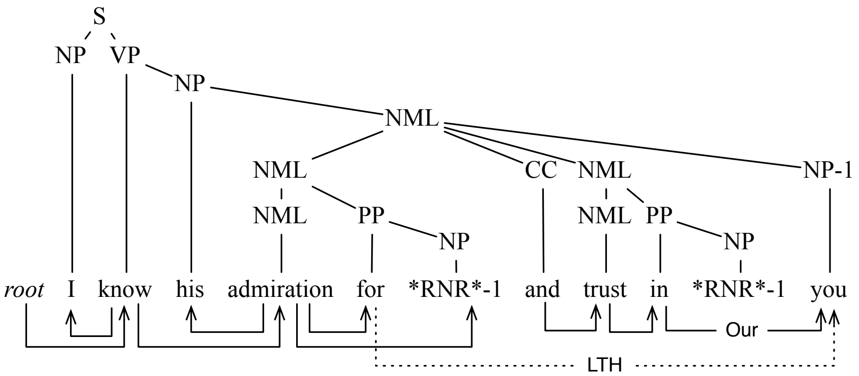


Figure 4: Right node raising (`*RNR*`) example. The solid dependency link labeled as 'Our' shows our way of handling `*RNR*`, and the dashed dependency labeled as 'LTH' shows Johansson's way of handling `*RNR*`. After the conversion, `*RNR*` nodes are dropped from the dependency tree.

## 3.1 Corpora

For our experiments, we use OntoNotes release 4.0 (Hovy et al. [7]).[4] We choose, from among all data, six different English corpora: GALE Broadcast Conversation (EBC), GALE Broadcast News (EBN), the Sinorama Chinese Treebank English Translation (SIN), the Xinhua Chinese Treebank English Translation (XIN), GALE Web Text (WEB), and the Wall Street Journal (WSJ). EBC and EBN contain broadcast conversations and news from various sources like CNN, NBC, etc. SIN contains English translations of the Sinorama magazine from Taiwan. XIN contains English translations of the Xinhua newswire from Beijing. WEB contains documents from web-blogs and newsgroups. Finally, WSJ contains non-financial news from the Wall Street Journal.[5]

| | EBC | EBN | SIN | XIN | WEB | WSJ | ALL |
|---|---|---|---|---|---|---|---|
| Train | 14,873 | 11,968 | 7,259 | 3,156 | 13,419 | 12,311 | 62,986 |
| Eval. | 1,291 | 1,339 | 1,066 | 1,296 | 1,172 | 1,381 | 7,545 |
| Avg. | 15.21 | 19.49 | 23.36 | 29.77 | 22.01 | 24.03 | 21.02 |

Table 3: Corpora distributions (in # of sentences). Avg.-row shows the average sentence length of each corpus (sentence length = # of words in sentence).

Table 3 shows how the corpora are divided into training and evaluation sets. For WSJ, we use Sections 0-21 for training, Section 24 for development, and Section 23 for evaluation. We do not separate development sets for the other corpora because

[4] At this moment, the OntoNotes release 4.0 is not available to the public, but it will soon to be available through the Linguistic Data Consortium (LDC).

[5] WSJ from OntoNotes contains only a subset of the entire Penn Treebank, extracting non-financial news only. Thus, the data is not bias from financial news.

all parsing models are optimized only for WSJ. ALL-column shows the total number of sentences when all corpora are combined.

### 3.2 Converting phrase structure to dependency structure

All corpora are annotated in Penn Treebank phrase structure style (Marcus et al. [11]). For each corpus, we generate two sets of dependency trees, one converted by the LTH tool, and the other converted by our approach. For the LTH tool, we mostly use their default settings, but exclude the GLARF labels (e.g., NAME, POSTHON, SUFFIX, TITLE; Meyers et al. [14]) and the APPO label for appositions. However, we include the QMOD label for quantifier phrases as we include this label in our approach. During conversion, we discard sentences of length 1 to avoid a bias to such a trivial case.

| | EBC | EBN | SIN | XIN | WEB | WSJ | ALL |
|---|---|---|---|---|---|---|---|
| LTH- dep | 1.44 | 0.81 | 0.70 | 0.29 | 0.95 | 0.51 | 0.82 |
| Our - dep | 1.29 | 0.73 | 0.69 | 0.21 | 0.83 | 0.46 | **0.73** |
| LTH- sen | 11.14 | 8.66 | 8.47 | 5.30 | 11.29 | 7.27 | 9.27 |
| Our - sen | 9.19 | 7.39 | 8.22 | 3.75 | 9.02 | 6.24 | **7.78** |

Table 4: Distributions of non-projective dependencies (in %). 'LTH' indicates output from the LTH tool, and 'Our' indicates output from our approach. The top two rows show % of non-projective dependencies among all dependencies. The bottom two rows show % of dependency trees containing at least one non-projective dependency. The numbers do not account for non-projective dependencies caused by punctuation.

Table 4 shows the distribution of non-projective dependencies in each corpus. Our approach generates fewer non-projective dependencies because of our new method of handling *RNR* nodes (cf. Section 2.6).

| | EBC | EBN | SIN | XIN | WEB | WSJ | ALL |
|---|---|---|---|---|---|---|---|
| LTH | 4.77 | 1.51 | 1.16 | 1.63 | 1.93 | 1.93 | **2.20** |
| Our | 0.86 | 0.57 | 0.33 | 0.44 | 1.03 | 0.25 | **0.60** |

Table 5: Distributions of unclassified dependencies (in %).

Table 5 shows the distribution of unclassified dependencies (labeled as DEP). Some of these dependencies may not be errors; they can be rather ambiguous in nature. However, reducing the number of unclassified dependencies as much as possible is preferred because they can appear as noise during training. The numbers show that our approach reduces the percentage of unclassified dependencies from 2.2% to 0.6%, a reduction of 72.7%.

### 3.3 State-of-art dependency parsers

To show the impact each conversion method has on dependency parsing, we train and test two state-of-art dependency parsers, MaltParser (Nivre et al. [16]) and

MSTParser (McDonald et al. [13]), and our own parser, ClearParser. First, we train all parsers on each corpus and test on the same corpus. Then, we train all parsers on WSJ, and test on the other corpora. In addition, we test all parsers on WSJ using models trained on EBN. No extensive optimization is made for any parser, so the performances of these parsers are expected to improve with further optimizations.

For MaltParser, we choose Nivre's *swap* algorithm for parsing (Nivre [15]), and LibLinear multi-class SVM for learning (Fan et al. [5]).[6] For MSTParser, we choose Chu-Liu-Edmonds' algorithm for parsing (McDonald et al. [13]), and the Margin Infused Relaxed algorithm (MIRA) for learning (Mcdonald and Pereira [12]). For ClearParser, we choose Choi-Nicolov's approach to Nivre's *list-based* algorithm for parsing (Choi and Nicolov [2]), and LibLinear L2-L1 SVM for learning.

## 3.4 Accuracy comparisons

### 3.4.1 Overall parsing accuracies

| | EBC | EBN | SIN | XIN | WEB | WSJ | ALL |
|---|---|---|---|---|---|---|---|
| Malt - LTH | 82.91 | 86.38 | 86.20 | 84.61 | 85.10 | 86.93 | 85.44 |
| Malt - Our | 83.20 | 86.40 | 86.03 | 84.85 | 85.45 | 87.40 | **85.65** |
| MST - LTH | 81.64 | 85.47 | 85.02 | 84.10 | 84.05 | 85.93 | 84.49 |
| MST - Our | 82.54 | 85.68 | 85.11 | 83.85 | 84.03 | 86.43 | **84.69** |
| Clear - LTH | 83.36 | 86.32 | 86.80 | 85.50 | 85.53 | 87.15 | 85.88 |
| Clear - Our | 84.06 | 86.77 | 86.55 | 85.41 | 85.70 | 87.58 | **86.09** |

Table 6: LAS (in %) when trained and tested on the same corpora.

| | EBC | EBN | SIN | XIN | WEB | WSJ | ALL |
|---|---|---|---|---|---|---|---|
| Malt - LTH | 74.80 | 82.40 | 81.74 | 79.39 | 80.42 | 80.59 | 80.01 |
| Malt - Our | 75.60 | 83.05 | 81.81 | 81.46 | 80.81 | 81.17 | **80.85*** |
| MST - LTH | 76.65 | 82.45 | 82.29 | 80.46 | 80.64 | 80.02 | 80.49 |
| MST - Our | 77.20 | 83.06 | 82.52 | 80.88 | 80.82 | 81.04 | **81.01*** |
| Clear - LTH | 76.37 | 83.16 | 83.53 | 81.29 | 81.83 | 81.29 | 81.36 |
| Clear - Our | 77.14 | 84.16 | 83.66 | 82.45 | 82.26 | 82.32 | **82.16*** |

Table 7: LAS (in %) when trained and tested on different corpora.

Table 6 shows labeled attachment scores (LAS) of each parser when trained and tested on the same corpora. All parsers give higher parsing accuracies on average using data generated by our approach. It is not clear which significance tests are appropriate for our data; the data contains too many dependencies (over 165K) so even a 0.2% improvement becomes statistically significant using the Chi-square test (thus, all improvements made in ALL-column are significant with $p \leq 0.025$).

[6] MaltParser comes with a default feature template designed for LibSVM (Chang and Lin [1]), so we contacted the MaltParser team to get a different feature template for LibLinear.

If we use the Wilcoxon signed-ranks test, pretending that each corpus is an independent sample, these 0.2% improvements are not statistically significant.

Table 7 shows LAS of each parser when trained and tested on different corpora. All parsers use models trained on WSJ except they use models trained on EBN when testing on WSJ. All parsers make statistically significant improvements (by the Wilcoxon signed-ranks test, $p \leq 0.03$) on average using data generated by our approach. This suggests that our method might create more uniform structures across different corpora, improving domain adaptation.

### 3.4.2 Parsing accuracies on semantic dependencies

Table 8 shows LAS on semantic dependencies (BNF, DIR, EXT, LOC, MNR, PRD, PRP, TMP) using the ALL evaluation set. There are a total of 11,583 and 11,934 semantic dependencies in the LTH and our data, respectively. In terms of F1-score, all parsers again give consistently higher parsing accuracies using data generated by our approach.

| | Precision | | Recall | | F1-score | |
|---|---|---|---|---|---|---|
| | LTH | Clear | LTH | Clear | LTH | Clear |
| Malt - Same | 67.97 | **68.68** | 62.67 | **63.00** | 65.21 | **65.72** |
| MST - Same | 66.86 | **67.67** | 60.80 | **60.82** | 63.69 | **64.06** |
| Clear - Same | 69.56 | **70.22** | 65.62 | **65.74** | 67.53 | **67.91** |
| Malt - Diff | 57.33 | **57.83** | 55.99 | **56.38** | 56.65 | **57.10** |
| MST - Diff | 57.65 | **58.53** | **54.64** | 54.49 | 56.10 | **56.44** |
| Clear - Diff | 58.81 | **60.22** | 58.60 | **59.69** | 58.70 | **59.95** |

Table 8: LAS on semantic dependencies using the ALL evaluation set. Same/Diff-rows show results using the same/different corpora for training and testing.

Table 9 shows LAS on joined semantic dependencies (e.g., LOC-PRD) using the LTH-ALL evaluation set (our data does not include those dependencies; Section 2.3). There are a total of 240 of these dependencies. As shown, all parsers give below 50% accuracies on these dependencies.

| | Same | | | Diff | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1-score | Precision | Recall | F1-score |
| Malt | 56.29 | 35.42 | 43.48 | 45.54 | 21.25 | 28.98 |
| MST | 58.82 | 33.33 | 42.55 | 60.49 | 20.42 | 30.53 |
| Clear | 66.43 | 39.58 | 49.60 | 51.49 | 28.75 | 36.90 |

Table 9: LAS on joined semantic dependencies using the LTH-ALL evaluation set.

# 4 Conclusion and future work

We present a more robust method of doing constituent-to-dependency conversion across different corpora. Our results show that parsers generally perform more accurately using data generated by our approach than by the LTH tool; especially for semantic dependencies. The improvements are even more significant when parsing models are tested on corpora different from their training corpora, leading us to believe that our method generates data that is more suitable for domain adaptation.

In the future, we are planning to apply automatic error detection techniques (Dickinson [4]) to make our data more consistent. We will also generate dependency trees with empty categories for dropped arguments. Finally, we will try to integrate semantic roles from PropBank directly into the dependency trees.

## Acknowledgments

We gratefully acknowledge the support of the National Science Foundation Grants CISE-CRI-0551615, Towards a Comprehensive Linguistic Annotation and CISE-CRI 0709167, Collaborative: A Multi-Representational and Multi-Layered Treebank for Hindi/Urdu, and a grant from the Defense Advanced Research Projects Agency (DARPA/IPTO) under the GALE program, DARPA/CMO Contract No. HR0011-06-C-0022, subcontract from BBN, Inc. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## References

[1] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[2] Jinho D. Choi and Nicolas Nicolov. K-best, locally pruned, transition-based dependency parsing using robust risk minimization. In *Recent Advances in Natural Language Processing V*, pages 205–216. John Benjamins, 2009.

[3] Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. Question answering passage retrieval using dependency relations. In *Proceedings of ACM-SIGIR'05*, pages 400–407, 2005.

[4] Markus Dickinson. Correcting dependency annotation errors. In *Proceedings of EACL'09*, pages 193–201, 2009.

[5] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification journal of machine learning research. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[6] J. Hajič, M. Ciaramita, R. Johansson, D. Kawahara, M. A. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, P. Straňák, M. Surdeanu, N. Xue, and Y. Zhang. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of CoNLL'09 Shared Task*, pages 1–18, 2009.

[7] Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. Ontonotes: the 90 In *Proceedings of HLT-NAACL'06*, pages 57–60, 2006.

[8] Richard Johansson. *Dependency-based Semantic Analysis of Natural-language Text*. PhD thesis, Lund University, 2008.

[9] Richard Johansson and Pierre Nugues. Extended constituent-to-dependency conversion for english. In *Proceedings of NODALIDA'07*, 2007.

[10] M. Marcus, G. Kim, M. Ann Marcinkiewicz, R. Macintyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. The penn treebank: Annotating predicate argument structure. In *ARPA Human Language Technology Workshop*, pages 114–119, 1994.

[11] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19 (2):313–330, 1993.

[12] Ryan Mcdonald and Fernando Pereira. Online learning of approximate dependency parsing algorithms. In *Proceedings of EACL'06*, pages 81–88, 2006.

[13] Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of HLT-EMNLP'05*, pages 523–530, 2005.

[14] Adam Meyers, Ralph Grishman, Kosaka Michiko, and Shubin Zhao. Covering treebanks with glarf. In *Proceedings of the EACL'01 Workshop on Sharing Tools and Resources for Research and Education*, 2001.

[15] Joakim Nivre. Non-projective dependency parsing in expected linear time. In *Proceedings of ACL-IJCNLP'09*, pages 351–359, 2009.

[16] Joakim Nivre, Johan Hall, and Jens Nilsson. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC'06*, pages 2216–2219, 2006.

[17] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.

[18] Owen Rambow, Cassandre Creswell, Rachel Szekely, Harriet Taber, and Marilyn Walker. A dependency treebank for english. In *Proceedings of LREC'02*, 2002.

[19] Libin Shen, Jinxi Xu, and Ralph Weischedel. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL:HLT'08*, pages 577–585, 2008.

[20] M. Čmejrek, J. Cuřín, and J. Havelka. Prague czech-english dependency treebank: Any hopes for a common annotation scheme? In *HLT-NAACL'04 workshop on Frontiers in Corpus Annotation*, pages 47–54, 2004.

[21] Hiroyasu Yamada and Yuji Matsumoto. Statistical dependency analysis with support vector machine. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT'03)*, pages 195–206, 2003.

# Semi-automatic Propbanking for French

Claire Gardent and Christophe Cerisara

CNRS/LORIA, Nancy
Firstname.Lastname@loria.fr

**Abstract**

Although corpora annotated with both syntactic and semantic role annotations are now available for many languages, no such corpus is currently available for French. To address this shorcoming, we present a methodology for pre-annotating the semantic roles of verb arguments semi-automatically. We discuss the results obtained and give pointers for improving the approach.

## 1 Introduction

Corpora annotated with both syntactic and semantic role annotations permits training semantic role labellers i.e., systems which can identify and characterise (usually verbal) predicate/argument dependencies in text. For English, Propbank has been widely used [7] as well as Framenet [2]. As witnessed by the 2009 ConLL shared task "Syntactic and Semantic Dependencies in Multiple Languages", Propbank style corpora are available for many other languages such as in particular, German, Spanish, Catalan, Chinese, Korean. For French however, no such resource is currently available.

In this paper, we describe a methodology for pre-annotating verb arguments with semantic roles semi-automatically. Section 2 presents the methodology used, Section 3 discusses the results obtained and Section 4 concludes by summarising what remains to be done in order to obtain a fully annotated Propbank for French.

## 2 Methodology

We take as a starting point a corpus of newspaper articles annotated with dependency structures namely, the Paris 7 Dependency Treebank (P7Dep, [3]). This corpus gathers articles from Le Monde and contains 350 931 tokens, 12 351 sentences and 25 877 verb instances. The corpus was semi-automatically annotated with phrase structure trees [1] and manually verified, and the resulting treebank automatically converted to dependency structures [3]. The phrase structure annotations were furthermore exploited to automatically extract the TreeLex syntactic lexicon [6], which was then manually corrected.

To enrich the P7 dependency corpus with role labels, we first manually enrich the TreeLex lexicon with thematic grids so that each (verb, subcategorisation frame) pair is enriched with the appropriate syntax-to-semantics linking information (each syntactic argument is mapped to the appropriate semantic role). We then automatically label each verb instance with the subcategorisation frame used by that verb instance. Finally, we project from the enriched TreeLex lexicon, the thematic roles registered in this lexicon for that particular (verb, frame) pair. More specifically, we proceed in three steps as follows:

**Adding thematic grids to Treelex.** We use various existing resources (i.e., Dicovalence [8] and Propbank frame files [7]) to manually enrich the (verb,frame) pairs listed in Treelex with a linking between syntactic arguments and thematic roles. Since Treelex is a subcategorisation lexicon extracted from the P7 corpus, the verbs covered in this lexicon cover the verbs to be labelled in the corpus.

**Associating P7 verb instances with subcategorisation frames.** This is a preliminary step which permits projecting the thematic grid information contained in the enriched Treelex onto each verb instances in the P7 corpus. It consists in identifying the deep grammatical functions of each verb instance in the corpus. For instance, given the sentence *The cat is chased by the rat*, the surface agentive phrase *the rat* will be labelled as deep subject and the surface subject *the cat* as deep object.

**Projecting Treelex thematic grids onto P7 verb instances.** This steps builds on the previous two steps. For each verb instance in the P7 corpus, it projects the thematic grid information contained in the enriched Treelex onto the deep grammatical functions identified by the subcategorisation frame identification step. For instance, given the above sentence, it will project the $a_0$ label onto the deep subject *the rat* and the $a_1$ label onto the deep object *the cat*.

The procedure builds both on the parsed structure already present in the treebank and on the subcategorisation information present in Treelex which was extracted from this parsed corpus. The parse information facilitates the identification for each verb instance occurring in the corpus of its deep grammatical arguments. The subcategorisation information contained in Treelex once enriched with thematic grid permits an automated projection of thematic roles onto the parsed structure via the deep grammatical functions identified by the second step of the procedure. We now describe in more detail each of these steps.

## 2.1 Adding thematic grids to Treelex.

The aim of this first step is to associate each lexical entry (i.e., each (verb, subcategorisation frame) pair) in Treelex with a thematic grid and a mapping between grammatical functions and thematic roles. For instance, given the following lexical entry:

| | |
|---|---|
| abîmer | SUJ:NP, OBJ:NP |
| *(to damage)* | *Ce champignon abîme les graines.* |
| | *(This fungus damages the seeds.)* |

the aim is to produce the following enriched lexical entry:

| | |
|---|---|
| abîmer | SUJ:NP:0 OBJ:NP:1 |
| | *Ce champignon abîme les graines.* |
| | abîmer.01 damage.01 to harm or spoil |
| | 0 agent, causer |
| | 1 entity damaged |

**Resources used.** To produce such entries, we use information from the P7 corpus, Dicovalence [8] and Propbank [7].

The *P7 corpus* gives us information about the usages of the verb in the form of sentences containing instances of it. We use this to help determine the meaning associated with each (verb, frame) pair.

*Dicovalence* is a subcategorisation lexicon which covers the most common French verbs and contains extensive information about each verb including in particular a translation to English. We use the Dicovalence translations of a verb as an indicator of its meaning and a bridge to the English Propbank.

Finally, the English *Propbank frames* associate a verb with a so-called roleset consisting of a verb meaning, a thematic grid and some illustrating examples. We use the Propbank frames to determine the thematic grid to be associated with a (verb,frame) pair in TreeLex given the verb meaning suggested by the English translation.

**Manual editing.** Given the information extracted from the P7 corpus (verb usage), Dicovalence (English translation for the verbs) and the Propbank frames (thematic grids), TreeLex is manually edited to associate each (verb,frame) pair with a meaning identifier, an english translation and an English gloss of that meaning, a thematic grid and a mapping between syntactic arguments and thematic role as illustrated by the enriched lexical entry for *abîmer* given above. The resulting files form the frame files of the French P7-Propbank.

This step of the procedure is time intensive with an average processing speed for a qualified linguist of 10 verbs per hour. Since there are 2 006 verbs in the Treelex lexicon, only a fraction of the verbs could so far be assigned a frame file thereby impacting semantic role labelling. We actually believe that a better way to proceed would be to first create verb classes and in a second step, to assign thematic grids to these classes rather than to isolated verbs. The automatic acquisition of verb classes from existing lexicons described in [5] is here particularly relevant. Indeed, we plan to apply this acquisition method to Treelex and to investigate in how far, the classes thus created group together verbs with identical thematic grids and more particularly, identical mapping between syntactic arguments and thematic roles. In this way, instead of individually annotating 2 000 verbs, we would only need to annotate a few hundred classes.

## 2.2 Associating P7 verb instances with subcategorisation frames.

This step labels each verb argument with a deep grammatical function and a category consistent with the Treelex signature[1]. It then checks whether the resulting subcategorisation frame assigned to the verb is assigned to this verb by Treelex. Verbs labelled with a Treelex frame and verbs not labelled with a Treelex frame can then be distinguished and processed separately e.g., for debugging puroposes. More specifically, the frame labelling process proceeds in three steps namely, argument extraction and processing ; normalisation e.g. of passive and causative structures ; comparison with Treelex frames.

### 2.2.1 Argument extraction and processing

For each verb, a verb description is first produced which, based on the verb mood, on the verb auxiliary (if any) and on its arguments describes the verb environment (passive/active, infinitive/participial/finite form, causative embedding) and its arguments. For instance, given the P7 dependency annotations of the sentence shown at the top of Figure 2, the description associated with the verb *succèdera* (*to succeed*) will be as given in the lower part of the Figure. Additionally (though not shown by the graphical interface), the verb is marked as active.

This conversion from dependency annotations to verb description is implemented by a set of rewrite rules which assign each word related to the verb by an argumental relation, an argument description in the Treelex format i.e., a pair FUNCTION:CATEGORY where FUNCTION and CATEGORY are as listed in the Treelex part of Table 1. As indicated in this Table, the argumental relations taken into account to identify the arguments of a verb are the P7 relations suj, obj, de_obj, a_obj, p_obj, ats, ato and aff. For instance, the subject rule is as follows:

If *F = suj(V)* :

- If *cat(F)* $\in \{A, N, ET, CL, D, PRO, P + PRO, P + D\}$ then *SUJ:NP*
- If *cat(F)* = $P$ then *SUJ:PP*
- If *cat(F)* = $C$ then *SUJ:Ssub*
- If *cat(F)* = $VINF$ then *SUJ:VPinf*

Additionally, verb features are used to assign one or more of the following features to the verb description: infinitival, participial, passive and causative.

[1] The signature used to specify syntactic categories and functions in the P7 dependency treebank differs from that used in TreeLex. Hence the rules must map the P7Dep functions and categories to those used in TreeLex. This mapping is given in Table 1

| **TreeLex** | description | **P7DEP** |
|---|---|---|
| SUJ | subject | suj |
| OBJ | object | obj |
| DE-OBJ | de-prepositional object | de_obj |
| A-OBJ | à-prepositional object | a_obj |
| P-OBJ | other prepositional object | p_obj |
| ATS | subject attribute | ats |
| ATO | object attribute | ato |
| refl | reflexive pronoun | aff |
| obj | affix | aff |

| **TreeLex** | **P7DEP** |
|---|---|
| NP | N |
| Ssub | C |
| PP | P |
| VPinf | VINF |
| il | il |
| en | en |
| CL | CL |
| AdP | ADV |
| y | y |
| VPpart | VPR |
| AP | A |

Figure 1: Mapping P7/Treelex

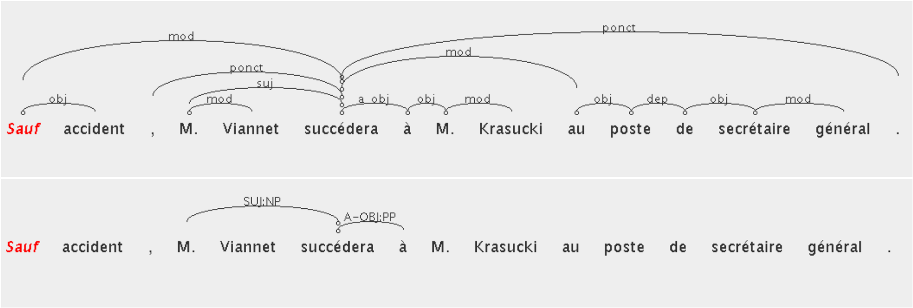

mod
ponct
mod
ponct
suj
obj
mod
a_obj
obj
mod
obj
dep
obj
mod
Sauf accident , M. Viannet succédera à M. Krasucki au poste de secrétaire général .
SUJ:NP
A-OBJ:PP
Sauf accident , M. Viannet succédera à M. Krasucki au poste de secrétaire général .

Figure 2: P7 dependency annotation and the resulting verb description for sentence *Barring accidents, M. Viannet will succeed M. Krasucki as general secretary.*

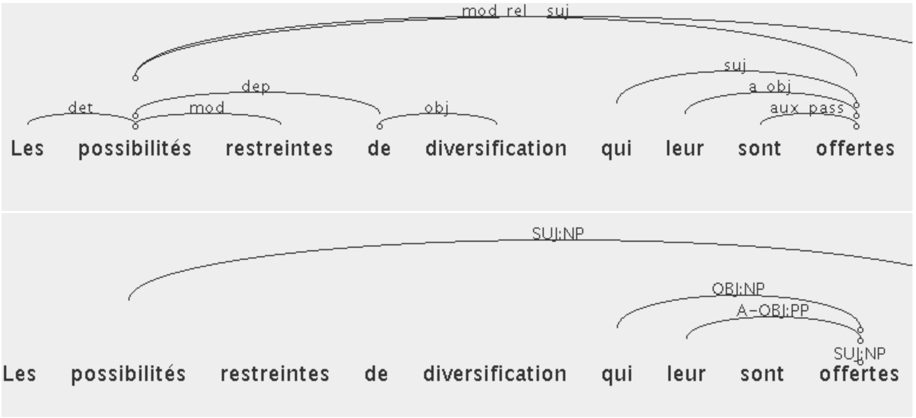

mod_rel
suj
suj
dep
a_obj
det
mod
obj
aux_pass
Les possibilités restreintes de diversification qui leur sont offertes
SUJ:NP
OBJ:NP
A-OBJ:PP
SUJ:NP
Les possibilités restreintes de diversification qui leur sont offertes

Figure 3: Normalising a passive in sentence *The limited diversification possibilities that are offered to them*

### 2.2.2 Normalisation

Given the verb description produced for each verb instance by the preceding step, the normalisation phase rewrites the frames of all verbs occurring in a passive, infinitival, participial or causative environment. The result is a frame assignment which relate each verb instance in the P7 dependency corpus to its arguments by an edge labeled with a deep grammatical function and a Treelex syntactic category. For instance, the frame assignment derived from the P7 dependency annotations for the verb *offertes* (*offered*) shown in the upper part of Figure 3 is as shown in the lower part of this figure. The surface subject *qui* (*that*) is labelled as an object NP, the dative clitic *leur* (*to them*) as a prepositional à -object and a subject NP is added.

### 2.2.3 Comparison with Treelex frames.

Finally, for each verb instance occuring in the P7 dependency corpus, the frame found by the above extraction procedure is checked against the frames associated with that verb by Treelex. If the frame exists in Treelex, the frame assignment is validated. Otherwise, the verb token is marked as having a non validated syntactic frame.

## 2.3 Projecting Treelex thematic grids onto P7 verb instances

The final step of the procedure assigns thematic roles to the deep arguments assigned to verb instances by the previous step using the Treelex lexicon enriched with thematic information described in section 2.1. For instance, given the en-

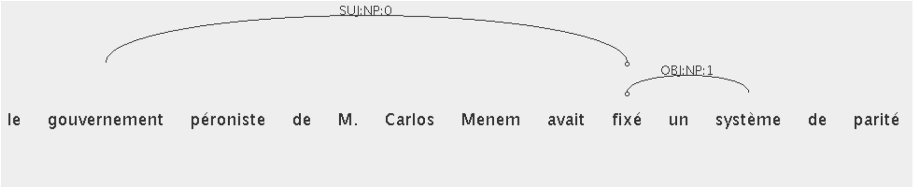

SUJ:NP:0
OBJ:NP:1
le gouvernement péroniste de M. Carlos Menem avait fixé un système de parité

Figure 4: Final output

riched Treelex lexical entry for *fixer* shown below, the final output of our labelling procedure is as shown in Figure 4.

fixer SUJ:NP:0, (OBJ:NP:1)
fixer.01 establish, set
0 agent, setter
1 thing set
2 location, position, attribute
*En mars , le gouvernement péroniste de M. Carlos Menem avait fixé*
*un système de parité de 10000 australs pour 1 dollar,*
*en vertu de la loi de convertibilité approuvée par le Congrès.*
(*In march, the peronist government of M. Carlos Menem had set a*
*parity system of 10000 australs for 1 dollar, under the convertibility*
*low approved by the Congress.)*

# 3 Results and Evaluation

We applied the pre-annotation procedure described in the previous section to the P7 corpus annotated with dependency structures. This corpus contains 350 931 tokens, 12 351 sentences and 25 877 verb instances. 78% (25 113) of the verb instances were assigned a Treelex frame by the first step of the procedure and 42% (13815 tokens) could be labelled with semantic roles.

To analyse the output of each step of the role labelling procedure (frame extraction, frame validation by TreeLex, grid assignment), we developped some visualisation and annotation tools. We then carried out a pilote evaluation to assess precision (the correctness of results found) and recall (the proportion of correct results found).

## 3.1 Visualisation and annotation tools

To visualise and analyse the results of the semi-automatic annotation procedure described in the previous section, we developed a graphical interface which permits visualising intermediate and final results and separating sentences for which all verb tokens were successfully processed from sentences where at least one verb token could not be processed [2]. More specifically, the menu provides 10 distinct views of the results, each view being named after (i) the annotations shown and (ii) the sentences they contain. The annotations can be one of the following. DEPS are the dependency annotations present in the initial P7 dependency corpus. In intermediate result files, dependency annotations are useful for checking whether a missing frame/grid stems from a parse error. RES are all the annotations produced by the annotation procedure described in the previous section. FRAMES are the subcategorisation frames extracted by the frame assignment procedure but not present in Treelex. This annotation level is useful for checking whether the frames found but not present in Treelex are either missing in Treelex or an incorrect result of the extraction procedure. TLFRAMES are the subcategorisation frames extracted by the frame assignment procedure and present in Treelex for the verb considered. These annotation level permits checking the precision of the extraction procedure (are the frames found and validated by Treelex actually the correct frames for the given verb tokens?). Finally, ROLES annotations are the thematic grids extracted by the SRL procedure. This annotation level when merged with the dependency annotations permits constructing the output Propbank.

Furthermore, the sentences contained in a file viewed can be any of the following. A P7 view will contain the entire P7 corpus; a NOTINTL view gathers sentences containing at least one verb whose extracted subcategorisation frame does not occur in Treelex. The ALLINTL views groups together sentences such that all verb tokens in those sentences were assigned a Treelex frame. A NOGRID view contain all the sentences where there is at least one verb for which no thematic grid could be extracted. Finally, the ALLINPBK view gathers sentences such that all verb tokens in those sentences were assigned a thematic grid.

### 3.2 Missing information (low recall)

There can be several reasons for the non identification of a frame or of a thematic grid.

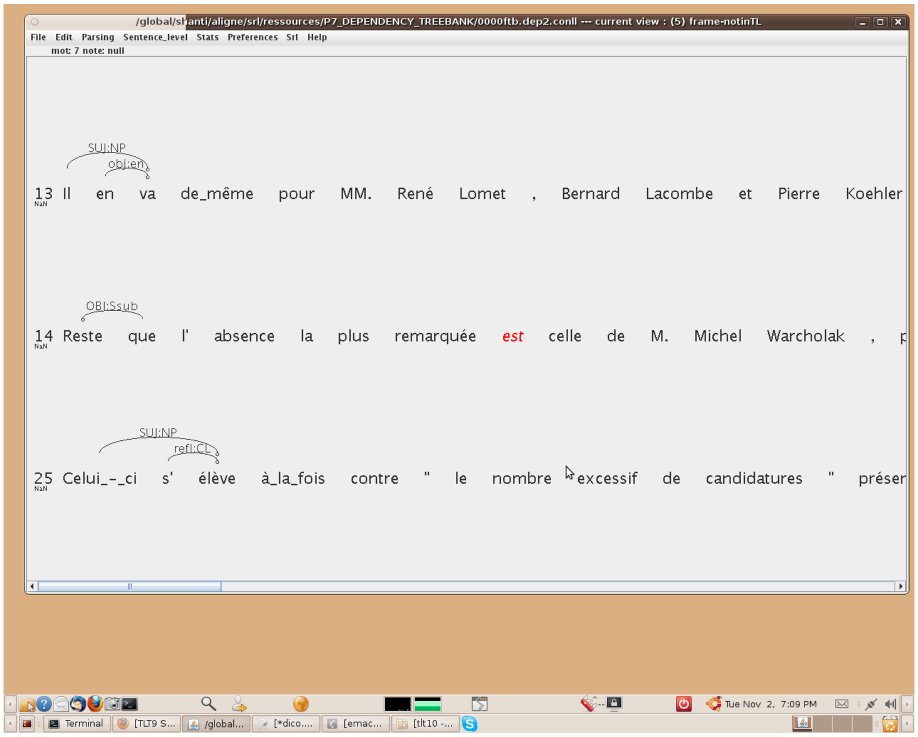
A missing frame may stem from an incorrect dependency structure[3], a missing frame in Treelex or an incorrect/missing frame rewrite rule.

Missing thematic grids stem either from a missing frame (the verb token was not assigned a frame by the frame assignment procedure) or from a missing frame file (cf. section 2.1).

Decreasing the number of missing thematic grids requires improving the frame extraction step and extending the coverage of the frame files. As discussed in section 2.1, the latter is time intensive and will require a few more months for completion. Improving the former (the frame extraction step) requires analysing, quantifying and correcting the three possible sources of missing data (incorrect dependency

[2] This tool is available for download at `http://www.loria.fr/~cerisara/jsafran/index.html`

[3] This is turn may be due either to an incorrect annotation of the P7 treebank or to errors in the conversion script which project dependency structures from the initial constituency annotations.



13 Il en va de_même pour MM. René Lomet , Bernard Lacombe et Pierre Koehler

14 Reste que l' absence la plus remarquée *est* celle de M. Michel Warcholak , p

25 Celui_-_ci s' élève à_la_fois contre " le nombre excessif de candidatures " préser

Figure 5: A FRAMES-NOTINTL view with the beginning of three sentences: (1) *It is the same for M. René Lornet, Bernard Lacombe and Pierre Koehler...* (2) *Still the most notable absence is that of M. Michel Wecholak, ...* (3) *He protested against both "the excessive number of candidates" ...* In the first and third case, the treebank fails to record a complement relation between the verb and a prepositional phrase (Pour- (*for*) and Contre-PP (*against*) respectively) so that the correct frame cannot be matched with any of the frames listed for "aller" (*to be*) and "s'élever" (*to protest*) respectively. In the second case, the frame found is probably correct but not listed in TreeLex.

structure, missing Treelex frame, incorrect/missing frame rewrite rule). To carry out such an investigation, we use the NOTINTL views. The FRAMES-NOTINTL view shows the frames found for those verb tokens for which the found frame is not in Treelex while the DEPS-NOTINTL view shows their dependency annotation. We use the second view (DEPS-NOTINTL) to identify incorrect frame assignment due to a parse error and the first (FRAMES-NOTINTL) to identify both errors in the extraction procedure and missing frames in Treelex. On a random sample of 50 verb tokens in the FRAMES-NOTINTL view (i.e., for which no TreeLex frame could be found), the results are as given in the following table.

| | | |
|---|---|---|
| Treebank error | 22 | 44% |
| Missing Frame in Treelex | 16 | 32% |
| Incorrect/missing frame rewrite rule | 12 | 24% |
| TOTAL | 50 | 100% |

Manual inspection shows that treebank errors include erroneous dependency structures (often noun modifiers classified as de-objects or complements classified as modifiers) and incorrect lemmatisations (e.g., *secoué* (*shaked*) instead of *secouer* (*to shake*)). Missing Treelex frames often involves a mismatch between Treelex treatment of infinitival complements introduced by the preposition “de” and the treebank dependency structure annotation. Finally, incorrect/missing frame rewrite rules fall mainly into two cases namely, coordination and causative structures. We plan to extend the rewrite rules so as to correctly handle these structures too which should further increase the ratio of verb tokens for which a Treelex frame can be found. Provided Treelex and rewrite rule errors are fixed, the upper bound on the automatic identification of the subcategorisation frame of a verb token would thus approximates 88% the remaining errors being due to incorrect dependency annotations and missing information in TreeLex.

### 3.3 Erroneous frame assignment (precision)

Similarly, we analyse erroneous frame assignment by examining the ALLINTL views i.e., those sentences for which all verb tokens are assigned a frame validated by Treelex. On a sample of 50 verb tokens, 9 verb tokens were assigned an incorrect frame. Manual investigation showed the following distribution:

| | | |
|---|---|---|
| Treebank error | 7 | 14% |
| Incorrect/missing frame rewrite rule | 2 | 4% |
| Correct frame identification | 41 | 82% |

Again many of the treebank errors are noun complements categorised as verb de-objects.

### 3.4 Creating a training corpus for semantic role labelling

Our graphical interface also provides a functionality for merging dependency and thematic grid annotations so as to provide a training corpus for semantic role labelling. The format and content of this corpus is similar to the ConLL format [4].

## 4 Discussion, Conclusion and Perspectives

We have presented a semi-automated procedure for pre-annotating verb arguments with thematic roles in a dependency treebank for French. The automated part of the procedure (the identification of the deep grammatical functions of the verb arguments) accounts for 78% of the verb instances whereby the missing identifications are due mostly to errors in the dependency annotations (44% of the missing cases) and to missing information in the TreeLex lexicon (32%) of the missing cases). Only 12% of the missing identifications are due to errors in our procedure and these errors can relatively easily be fixed as the methods used (rewrite rules mapping surface to deep syntactic functions) are symbolic and the visualisation tools we developed, permit a detailed and systematic investigation of the error cases. Further, on the small sample we examined, precision (the proportion of correct mappings between surface and deep grammatical functions) reaches 82% with only 4% of the cases being due to errors in the annotation procedure, the remaining 14% being due to errors in the dependency annotations.

In sum, the automated part of our pre-annotation procedure displays a coverage and a precision which suggests that it can effectively support the development of a propositional bank for French. To ensure that the resulting annotated corpus supports the training of semantic role labellers, two points must be further pursued however.

First, Treelex must be fully augmented with thematic roles. As mentioned in section 2.1, this step could be enhanced by first producing a classification of French verbs which, as in the English VerbNet, groups together verbs, syntactic frames and thematic grids. [5] reports on an experiment in acquiring verb classes for French from existing lexical resources. This preliminary investigation suggests that Formal Concept Analaysis is an appropriate framework for bootstraping a verb classification for French from existing lexical resources and thereby to quickly associate thematic grids with sets of verb/frame pairs. In ongoing work, we are currently exploring how additionally taking into account syntactico-semantic features present in Dicovalence and in the LADL tables affects the classification and more specifically, whether such features permit creating verb classes that are sufficiently semantically homogeneous to contain mostly verbs that share the same thematic grid.

Second, adjuncts need to be dealt with. Indeed the present proposal focuses on so-called core arguments while Propbank style annotation requires that temporal, manner and locative adjuncts also be annotated. It remains to be seen in how much the combination of adjunct rewrite rule with taxonomical knowledge about the semantic type of the arguments suffices to correctly label verb adjuncts.

## References

[1] A. Abeillé, L. Clément, and A. Kinyon. Building a treebank for french. In *In Proceedings of the LREC 2000*, 2000.

[2] C. F. Baker, C. J. Fillmore, and J. B. Lowe. The berkeley FrameNet project. In *Proceedings of the 17th International Conference on Computational Linguistics*, volume 1, pages 86–90, Montreal, Quebec, Canada, 1998. Association for Computational Linguistics.

[3] M.-H. Candito, B. Crabbé, and M. Falco. Dépendances syntaxiques de surface pour le français. Technical report, Université de Paris 7, 2009.

[4] X. Carreras and L. Marquez. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the CoNLL-2005 Shared Task: Semantic Role Labeling*, pages 152–164, Ann Arbor, Michigan, June 2005.

[5] I. Falk and C. Gardent. Bootstrapping a classification of french verbs using formal concept analysis. In *Interdisciplinary workshop on verbs*, Pisa, Italy, 2010.

[6] A. Kupsc and A. Abeillé. Growing treelex. In Alexander F. Gelbukh, editor, *CICLing*, volume 4919 of *Lecture Notes in Computer Science*, pages 28–39. Springer, 2008.

[7] M. Palmer, P. Kingsbury, and D. Gildea. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.

[8] Karel van den Eynde and Piet Mertens. La valence: l'approche pronominale et son application au lexique verbal. *Journal of French Language Studies*, 13:63–104, 2003.

# Treebanking Finnish

Katri Haverinen,[1,3] Timo Viljanen,[1] Veronika Laippala,[2]
Samuel Kohonen,[1] Filip Ginter[1] and Tapio Salakoski[1,3]

[1]Department of Information Technology,
[2]Department of French studies
[3]Turku Centre for Computer Science (TUCS)
20014 University of Turku, Finland
first.last@utu.fi

### Abstract

In this paper, we present the current version of a syntactically annotated corpus for Finnish, the Turku Dependency Treebank (TDT). This is the first publicly available Finnish treebank of practical size, currently consisting of 4,838 sentences (66,042 tokens). The treebank includes both morphological and syntactic analyses, the morphological information being produced using the FinCG analyzer, and the syntax being human-annotated in the Stanford Dependency scheme. Additionally, we conduct an experiment in automatic pre-annotation and find the overall effect positive. In particular, pre-annotation may be tremendously helpful in terms of both speed and accuracy for an annotator still in training, although for more experienced annotators such obvious benefit was not observed.

In addition to the treebank itself, we have constructed a custom annotation software, as well as a web-based interface with advanced search functions. Both the treebank, including the full edit-history with exact timings, and its associated software are publicly available under an open license at the address http://bionlp.utu.fi.

## 1 Introduction

The applications of treebanks and their benefits for natural language processing (NLP) are numerous and well-known. Many languages, regardless of how widely spoken, already have a treebank, and for many others one is currently being developed. Finnish is among the less fortunate languages in the sense that it previously long lacked a publicly available treebank entirely. Even now, prior to this work, the only such treebank is our previously published small-scale treebank [3], which does not yet truly enable NLP research.

In this work, we aim to address the serious lack of NLP resources for Finnish, by extending our previous work into a freely available, practically sized treebank for Finnish, the Turku Dependency Treebank (TDT). The current, extended version of the treebank presented in this paper includes 4,838 sentences. The whole treebank has manually created syntax annotations in the well-known Stanford Dependency (SD) scheme [1, 9] and automatically created morphological analyses. The text of the treebank is drawn from four sources: the Finnish Wikipedia and Wikinews, popular blogs and a university web-magazine.

As a second contribution, we also conduct an experiment on the effect of automated pre-annotation on annotation speed and quality, by using a preliminary statistical parser induced from the treebank to produce an initial analysis.

The linguistic aspects of the work, such as the choice of the annotation scheme and the modifications needed to accommodate the specific features of the Finnish language have been thoroughly discussed in our previous paper on the first release of the treebank [3]. In particular, we have found the Stanford Dependency scheme suitable for the Finnish language, with only minor modifications needed. Thus this paper will rather focus on the annotation process point of view of the work.

## 2 Related Work

The only publicly available treebank of general Finnish is our previously released treebank version [3]. This version only consists of 711 sentences, and, unlike the extended treebank release presented here, lacks morphological information. Also, no inter-annotator agreement figures were presented for this previous release. In addition to the general Finnish treebank, there exists a recently published small-scale treebank and PropBank of clinical Finnish [4]. The size of this corpus is 2,081 sentences (15,335 tokens), and it includes morphological, syntactic and semantic annotation.

Due to the lack of a large, publicly available treebank, also Finnish NLP tools are scarce. Tools targeted at Finnish morphology include FinTWOL and FinCG, a commercial morphological analyzer and a constraint grammar parser that resolves morphological ambiguity [5, 6]. These tools are used in this work to provide morphological analyses for the treebank. The only previously available broad-coverage syntactic parser for Finnish is Machinese Syntax,[1] which is a closed-source commercial parser.

The syntactic representation scheme used in this work, the Stanford Dependency (SD) scheme [1, 9], is relatively widely used in NLP applications. Both the above mentioned treebanks of Finnish use this scheme and additionally, there is a third treebank that has native SD annotation. The BioInfer [13] treebank is an English language corpus of scientific abstracts in the biomedical domain. In addition to these native corpora, also any English language treebank that uses the Penn Treebank scheme [8] can be automatically converted into the SD scheme using existing tools[2].

[1] http://www.connexor.eu

[2] http://nlp.stanford.edu/software/lex-parser.shtml

## 3 Treebank Text

In the current version of the treebank, there are four distinct sources of text: the Finnish Wikipedia and Wikinews, popular blogs and a university web-magazine. These sources are selected on the basis of two criteria.

First, it is our fundamental belief that the treebank should be freely available under an open license, which restricts our choice of texts to those which either are published under an open license originally or for which we can, with reasonable effort, negotiate such a license. Three of the current sources, Wikipedia, Wikinews and the university web-magazine, were originally published under an open license, and for the blog texts, we have obtained the permission to re-publish the text from individual authors.

Second, we have strived for linguistic variety in the texts. We have specifically limited the amount of text chosen about the same topic, and by the same author. In Wikipedia, this is naturally achieved by choosing the articles randomly, as both the amount of articles and the amount of authors are large. In the Finnish Wikinews, the number of authors is substantially smaller, and thus we have, when choosing an article from this source, first randomly selected an author, and only after that randomly selected one individual article by them. This selection process was repeated until a sufficient amount of articles had been chosen.

When selecting the blog texts, we have used several lists of most popular blogs and only selected blogs where the entries appeared to be of sufficient grammatical quality to allow proper annotation of syntax. In addition, the blogs were divided into categories, based on which topic the majority of the entries were about, and the amount of blogs selected from each category was limited. The current selection consists of two blogs from the category *personal and general*, one from the category *style and fashion* and one from the category *relationships and sex*. Naturally, this selection was affected by the permissions given by the authors. The individual texts were selected starting from the newest entries, discarding entries containing certain problematic properties, such as long quotes which could cause copyright issues. We limited the amount of text to be selected from one blog author to be approximately 200 sentences, so that individual entries from a blog were selected in order until the total amount of sentences surpassed 200. Thus the amount of entries chosen from each blog varies according to the length of the entries in that blog.

In the case of the university web magazine, articles were selected starting from the newest writings. Given the relatively limited topics, this section was restricted to a total of 50 articles, which results in a total of 942 sentences.

The breakdown of articles and sentences in different sections of the treebank is shown in Table 1. The table shows that the largest section of the treebank is currently the Wikipedia section, followed by the Wikinews section and the university web-magazine section. The smallest section is at the moment the blog section, which is due to the difficulty of gaining re-publication permissions from individual authors. Altogether the current version of the treebank consists of 4,838 sentences

| **Section** | **articles** | **sentences** | **tokens** |
|---|---|---|---|
| Wikipedia | 199 | 2,260 | 32,111 |
| Wikinews | 67 | 760 | 9,724 |
| Blogs | 32 | 876 | 10,918 |
| Web-magazine | 50 | 942 | 13,289 |
| **Total** | 348 | 4,838 | 66,042 |

Table 1: Breakdown of the treebank sections. As the annotation work still continues, it should be noted that the current breakdown of sections does not reflect the final composition of the treebank.

(66,042 tokens). Out of all sentences, 5.8% are non-projective. For comparison, the clinical Finnish treebank [4] is reported to have a non-projectivity rate of 2.9% of sentences.

In all sections of the treebank, we have annotated each selected text that is shorter than 75 sentences in its entirety. As for instance some Wikipedia articles may be as long as 300 sentences, longer texts have been truncated after the first 75 sentences to avoid biasing the treebank towards the topics of long texts. This strategy was also used in the construction of the first treebank release.

# 4 Syntax and Morphology Annotation in the Treebank

## 4.1 Syntax in the SD Scheme

Our choice for the syntactic representation scheme, the established Stanford Dependency (SD) scheme of de Marneffe and Manning [1, 9], is naturally the same as that used in our previous work. It is a dependency scheme, where syntax is represented as a graph of directed, labeled dependencies between words. The scheme has four different representation variants, which include a different subset of dependency types each. In effect, these variants are layers of dependencies that can be added on top of the basic dependency tree, to offer deeper information on the structure of the sentence. Therefore, the structures in all variants are not necessarily trees. The reader is referred to the original work of de Marneffe and Manning for further details on the SD scheme.

The current annotation of the treebank is based on the so called *basic* variant, where the analyses are trees and the dependencies are for the most part syntactic. The original *basic* variant of the SD scheme includes 55 dependency types, and our modified version 44 types. An example of a syntactic analysis of a Finnish sentence in the SD scheme is given in Figure 1.

In our previous work [3], we have shown that although originally designed for English, the SD scheme is well-suited for Finnish as well, with some minor changes. The reader is referred to this work for details of the modifications made to the original SD scheme, as the version of the scheme used in the current work is identical.

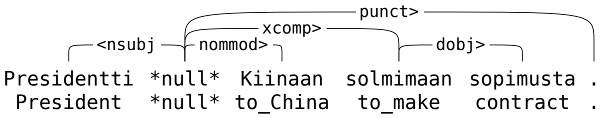

punct>
xcomp>
<nsubj
nommod>
dobj>
Presidentti *null* Kiinaan solmimaan sopimusta .
President *null* to_China to_make contract .

Figure 1: An example of a Finnish sentence annotated in the SD scheme. The sentence can be translated as *The president to China to make a contract*. The *null token* present in the analysis stands for the main verb which this fragmentary sentence lacks but which is necessary for the construction of a dependency analysis in the SD scheme.

| word | transl. | lemma | POS | comp. | case | tense | voice | num. | person |
|---|---|---|---|---|---|---|---|---|---|
| Ryöstäjä | Burglar | **ryöstäjä** | **N** | | **NOM** | | | **SG** | |
| poistui | leave | **poistua** | **V** | | | **PAST** | **ACT** | | **SG3** |
| pimeän | darkness | **pimeä** | **N** | | **GEN** | | | **SG** | |
| | dark | pimeä | A | POS | GEN | | | SG | |
| turvin | chub | turpa | N | | INS | | | PL | |
| | safety | **turva** | **N** | | **INS** | | | **PL** | |

Figure 2: FinTWOL and FinCG analyses. The words of the sentence and their translations are given in the two leftmost columns and the lemma in the third column, followed by all tags given to the word by FinTWOL. The readings selected by FinCG are shown in bold. The example sentence as read from the leftmost column can be translated as *The burglar left in the safety of the darkness*.

## 4.2 Morphology with FinTWOL and FinCG

We also add to the whole treebank, including our previously released subcorpus, morphological analyses created using two Finnish morphology tools: FinTWOL and FinCG[3]. For each word, FinTWOL gives all possible readings, each of which includes a detailed morphological analysis. Given the analysis by FinTWOL, FinCG aims to disambiguate which of the readings is correct in the current context. When unable to fully disambiguate a word, FinCG may select multiple readings. In the treebank, each token is given all of its FinTWOL readings, and those selected as correct by FinCG are marked. An illustration of the morphological information present in the treebank is given in Figure 2.

Manually annotated morphological analyses are currently left as future work, pending further investigation of the various issues involved, such as morphological analyzer licensing, defining a suitable annotation scheme and, naturally, funding.

[3] http://www.lingsoft.fi

| **Section** | **Annotator 1** | **Annotator 2** | **Annotator 3** | **Annotator 4** | **Overall** |
|---|---|---|---|---|---|
| Wikipedia | 95.1 | 84.0 | 90.4 | - | 89.5 |
| Wikinews | 96.3 | 87.7 | - | - | 92.0 |
| Blogs | 94.6 | 86.4 | - | - | 90.5 |
| Web-magazine | 96.6 | 89.5 | 92.0 | 70.6 | 88.6 |
| **Overall** | 95.5 | 86.2 | 90.8 | 70.6 | **89.9** |

Table 2: The inter-annotator agreement of different annotators across the sections of the treebank. All agreement figures are given in labeled attachment scores (%). Note that the LAS is calculated across all tokens and the averages in the table are thus implicitly weighted by the size of the various sections and annotator contributions. Therefore the overall figures are not the same as the averages of the individual annotator or section figures.

# 5 Annotating the Treebank

## 5.1 Annotation Process and Quality

Our annotation method for all sections of the treebank is the so called *full double annotation*. Each sentence is first independently annotated by two different annotators, and the resulting annotations are automatically merged into a single analysis, where all disagreements are marked. Disagreements are then jointly resolved, typically by all annotators in the group, and this results in the *merged annotation*. These annotations are further subjected to consistency checks, the purpose of which is to ensure that even old annotations conform to the newest annotation decisions. The result of these consistency checks is called the *final annotation*.

This annotation procedure allows us to measure the quality of the annotation and the suitability of the SD scheme for its purpose, using *inter-annotator agreement*. Rather than the *final annotation*, the agreement is measured for each annotator against the *merged annotation*, so as to avoid unfairly penalizing an annotator on decisions that were correct at annotation time but have later become outdated due to changes in the annotation scheme. Additionally, the *final annotation* may differ from the individual annotations in terms of numbers of tokens and sentences, as sentence boundaries and tokenization are corrected at this level where necessary. We use as the measure of inter-annotator agreement *labeled attachment score (LAS)*, which is the percentage of tokens that receive the correct head and dependency label. On average, our annotators achieved an inter-annotator agreement of 89.9% over the entire treebank. Figure 3 illustrates the development of inter-annotator agreement over time and Table 2 lists the agreements of individual annotators across the different sections of the treebank.

## 5.2 The Effect of Pre-annotation

On a 45 article subset of the web-magazine section, we have performed an experiment regarding annotation speed and quality, in order to find whether automatic

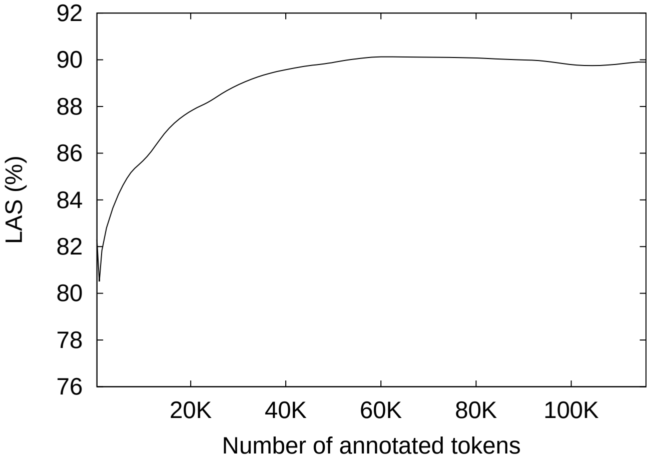


Figure 3: Development of inter-annotator agreement as the amount of annotated tokens grows. Note that the total number of tokens is twice the size of the treebank due to the double-annotation protocol.

| Speed [sec/token] | | | | LAS [%] | | | |
|---|---|---|---|---|---|---|---|
| **Annotator** | **plain** | **pre.** | ***p*** | **Annotator** | **plain** | **pre.** | ***p*** |
| Annotator 1 | 3.63 | 3.01 | 0.01 | Annotator 1 | 97.4 | 95.3 | <0.001 |
| Annotator 2 | 4.61 | 4.45 | 0.60 | Annotator 2 | 89.0 | 89.6 | 0.64 |
| Annotator 3 | 5.60 | 5.39 | 0.65 | Annotator 3 | 92.3 | 91.7 | 0.63 |
| Annotator 4 | 6.92 | 4.59 | 0.001 | Annotator 4 | 64.0 | 78.7 | <0.001 |
| Δ | | -0.76 | <0.001 | Δ | | 2.29 | 0.034 |

Table 3: Results of the pre-annotation experiment. The left-hand side shows annotation speed and the right-hand side the LAS. For each annotator are given their base speed, averaged across all plain documents, their pre-annotated speed, and the p-value for the difference. Similarly for LAS. The Δ values are the change of speed or LAS across annotators, corrected for each annotator's base speed or labeled attachment score.

pre-annotation would be helpful (or possibly harmful) for our annotation process. Previously, beneficial effects have been reported by for instance Rehbein et al. [14] and Fort and Sagot [2], on different linguistic annotation tasks.

For the purposes of this experiment, we have produced the first baseline statistical parser of Finnish. Our parser was built using the MaltParser system of Nivre et al. [11], which can be used to automatically induce a parser for a new language, given a treebank. The parser was developed using the body of annotated data available before commencing the experiment discussed in this section, in total 3,648 sentences (48,950 tokens), gathered from all sections of the treebank, including also a small portion of the web-magazine section. From this data, 80% was used for parser training, 10% for parameter estimation, and 10% for testing. Substantial effort was invested into parameter and feature selection in inducing the parser; the LAS of 70% achieved by the parser thus forms a non-trivial parsing baseline for Finnish. The parser was used to provide automatically pre-annotated versions of each of the documents in our experiment set. The division of documents among annotators was then performed so that for each document, one annotator was assigned the pre-annotated version and the other annotator was to start from an unannotated one (referred to as *plain* hereafter) exactly as in our regular annotation setting. The automatically produced dependencies were visually marked so that the annotator could easily distinguish between dependencies already considered and those still awaiting confirmation or correction.

We calculate the annotation speed in seconds per token and annotation accuracy in terms of LAS. To evaluate the effect of pre-annotation for individual annotators, we compare their speed and LAS on *pre-annotated* vs. *plain* documents and establish statistical significance using the unpaired, two-tailed t-test. The results are shown in Table 3. Our Annotator 4, who has only recently started annotation training and consequently receives the lowest base speed and LAS by far, benefited from the pre-annotation by a tremendous amount, with regard to both speed and LAS. In fact, this annotator's LAS on the plain documents is worse than that of the baseline parser, but given a pre-annotated text, the annotator's LAS clearly exceeds the parser performance. Our most experienced annotator, Annotator 1, gained a small benefit in speed, but suffered a small but statistically significant decrease in LAS. Annotators 2 and 3 did not have a statistically significant difference in speed or LAS.

Since each document was annotated by two different annotators, once as plain and once as pre-annotated, we can further establish the overall effect of pre-annotation across all documents regardless the annotator, using the paired, two-tailed t-test to test for statistical significance. This, however, involves comparing speeds and accuracies between different annotators, which are not directly comparable since individual annotators differ notably in their typical annotation speed and LAS. To take this into account, we establish the base speed and LAS of each annotator across all plain documents (columns *plain* in Table 3) and subtract these from the per-document values before performing the comparison. We are thus comparing changes in speed and LAS, rather than directly their values. We find that pre-annotated documents were on average annotated 0.76 seconds/token faster than plain documents (significant with $p<0.001$) and their LAS was on average 2.29 percentage points higher (significant with $p=0.034$).

Therefore, we conclude that whether pre-annotation is beneficial or harmful depends strongly on the annotator. It would seem that an inexperienced annotator can greatly benefit from a starting point for their work, but for more experienced annotators there was no similar benefit. The risk of overlooking mistakes in a pre-annotated text may contribute to this, and additionally at least Annotator 2 reported difficulties in adapting to the new style and technique of annotation. Naturally, it could also be suggested that a parser with a better performance could potentially be more helpful for even more experienced annotators. This matter is worth investigating further.

## 6 Released Data and Software

When releasing the treebank, we do not merely release the text together with its final annotation, but rather the full history leading to the final data. That is, in addition to the final data, we release the independent annotations of both annotators on each document, as well as the *merged* annotation, which is the result of discussing the disagreements between annotators.

Our most important reason for releasing this intermediate data is that each annotated document contains the full edit history of that document, including the exact times (at the resolution of a millisecond) of each edit action performed by an annotator. We believe that this kind of data could potentially be very useful for research, especially for studies on the difficulty of different phenomena encountered in an annotation task, such as the recent work by Tomanek et al. [15]. To our knowledge such detailed data included in a treebank is unique, and it may be a useful resource for future research. In addition, the data makes our own work more transparent. For instance, it allows the replication of the results presented in this paper.

We note that a fraction of approximately 10% of the treebank data in this as well as future releases will be held private, for the purposes of possible future shared tasks on Finnish parsing and parser comparison in general.

Finally, we release a web-based interface for the treebank. This interface allows the user to browse the treebank, as well as make advanced searches. It is possible to search in the text of the treebank, in the morphological analyses, and in the syntactic trees. Morphological and syntactic searches can also be combined, by for instance searching for present tense third singular form verbs that have as their subject a noun that is in partitive. Also searches with a more complex dependency structure are possible, using a syntax akin to TRegex [7] and Tgrep[4]. Detailed documentation of the search features is beyond the scope of this paper and can be found on the project web-page.

## 7 Conclusion and Future Work

In this work, we have presented an extended version of a freely available treebank for Finnish, the Turku Dependency Treebank (TDT). The size of the current treebank is 4,838 sentences (66,042 tokens), and it consists of four sections, with text from different sources: the Finnish Wikipedia and Wikinews, assorted blogs and a university web-magazine. These sources were selected with the aim to keep the

[4] http://crl.ucsd.edu/software/

treebank freely available under an open license, as well as to ensure a sufficient variation of topics and authors.

The treebank has two levels of analysis: morphological and syntactic. The morphological analyses are created automatically, using existing tools for Finnish. In the manually created syntax annotation, we have used the well established Stanford Dependency scheme, and in order to ensure high quality of the annotation, we have used the *full double annotation* protocol. The average inter-annotator agreement across the treebank was 89.9%. Such a high agreement suggests that annotator training is sufficient and that the annotation scheme is well-defined.

We have also performed an experiment on the effect of pre-annotation on annotation speed and quality and observed greatly improved performance, in terms of both speed and accuracy, for an annotator still in training. An expert annotator achieved a statistically significant gain in speed, although at the cost of a decrease in accuracy.

The treebank, our custom annotation software, detailed data on the annotation process, and a web-based interface of the treebank are available at the address `http://bionlp.utu.fi`.

This work has several important future work directions. The first and most obvious one is to further increase the size of the treebank, adding also new text sources. For instance, fiction text would be a valuable addition, and we are searching for fiction published under an open license. Our current goal is to annotate approximately 10,000 sentences, which appears to generally be enough to produce a robust statistical parser, as for example the results of the multiple language parsing study by Nivre [10] indicate. The second future work direction is to investigate the possibilities to improve the performance of the current parser and to release a fast and robust statistical parser for Finnish.

Thirdly, our goal is to enhance the treebank with additional annotation. Such annotation could, for instance, include human-validated morphological analyses. Also additional dependencies on top of the *basic* SD variant, following one of the extended variants of SD, could be a useful extension of the treebank. With such further annotation in place, it would be possible to add semantic information, for example more detailed analysis of the highly common *nominal modifiers*, with labels such as *temporal* and *cause*. Ultimately, these annotations would enable the development of the treebank into a fully fledged PropBank according to the model set by Palmer et al. [12]. The interaction between the SD and PropBank schemes has already been investigated in connection with the clinical Finnish PropBank [4], and the schemes were found compatible.

## Acknowledgements

We are grateful to Lingsoft Ltd. for making FinTWOL and FinCG available to us, as well as the permission to publish their analyses together with the treebank. We would also like to thank all the blog authors who kindly gave us the permission to include their work in the treebank. This work has been supported by the Academy of Finland.

## References

[1] Marie-Catherine de Marneffe and Christopher Manning. Stanford typed dependencies manual. Technical report, Stanford University, September 2008.

[2] Karën Fort and Benoît Sagot. Influence of pre-annotation on POS-tagged corpus development. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 56–63, 2010.

[3] Katri Haverinen, Filip Ginter, Veronika Laippala, Timo Viljanen, and Tapio Salakoski. Dependency annotation of Wikipedia: First steps towards a Finnish treebank. In *Proceedings of The Eighth International Workshop on Treebanks and Linguistic Theories (TLT8)*, pages 95–105, 2009.

[4] Katri Haverinen, Filip Ginter, Veronika Laippala, Timo Viljanen, and Tapio Salakoski. Dependency-based propbanking of clinical Finnish. In *Proceedings of The Fourth Linguistic Annotation Workshop (LAW IV)*, pages 137–141, 2010.

[5] Fred Karlsson. Constraint Grammar as a framework for parsing unrestricted text. In *Proceedings of COLING'90*, pages 168–173, 1990.

[6] Kimmo Koskenniemi. Two-level model for morphological analysis. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 683–685, 1983.

[7] Roger Levy and Galen Andrew. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of LREC'06*, pages 2231–2234, 2006.

[8] Mitchell Marcus, Mary Ann Marcinkiwicz, and Beatrice Santorini. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[9] Marie-Catherine de Marneffe and Christopher Manning. Stanford typed dependencies representation. In *Proceedings of COLING'08, Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, 2008.

[10] Joakim Nivre. Deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553, 2008.

[11] Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryiğit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135, 2007.

[12] Martha Palmer, Dan Gildea, and Paul Kingsbury. The Proposition Bank: an annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105, 2005.

[13] Sampo Pyysalo, Filip Ginter, Veronika Laippala, Katri Haverinen, Juho Heimonen, and Tapio Salakoski. On the unification of syntactic annotations under the Stanford dependency scheme: A case study on BioInfer and GENIA. In *Proceedings of BioNLP'07*, pages 25–32, 2007.

[14] Ines Rehbein, Josef Ruppenhofer, and Caroline Sporleder. Assessing the benefits of partial automatic pre-labeling for frame-semantic annotation. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 19–26, 2009.

[15] Katrin Tomanek, Udo Hahn, Steffen Lohmann, and Jürgen Ziegler. A cognitive cost model of annotations based on eye-tracking data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1158–1167, 2010.

# The TIGER Corpus Navigator

Sebastian Hellmann[1] Jörg Unbehauen[1]
Christian Chiarcos[2] Axel-Cyrille Ngonga Ngomo[1]

[1]University of Leipzig [2]University of Potsdam
E-mail: chiarcos@uni-potsdam.de
{hellmann|unbehauen|ngonga}@informatik.uni-leipzig.de

**Abstract**

Linguistically annotated corpora are a central resource in NLP. The extraction of formal knowledge from these corpora, however, is a tedious process. We introduce the Tiger Corpus Navigator, a Semantic Web system which aids users to classify and retrieve sentences from linguistic corpora – here, the Tiger corpus – on the basis of abstract linguistic concepts.

These linguistic concepts are specified extensionally, thus, independent from the underlying annotation: The user provides a small set of pre-classified sentences that represent instances (positive examples) or counterinstances (negative examples) of the corresponding concept, and the system automatically acquires a formal OWL/DL specification of the underlying concept using an Active Machine Learning approach.

## 1 Introduction

A large number of annotated corpora have become available over the past years. Still, the retrieval of dedicated linguistic knowledge for given applications or research questions out of these corpora remains a tedious process. An expert in linguistics might have a very precise idea of the concepts she would like to retrieve from a corpus. Yet, she faces a number of challenges when trying to retrieve corresponding examples out of a particular corpus:

**access** she needs a tool that is able to process the format of the corpus, that is easy to deploy, and that provides an intuitive user interface

**documentation** she needs to be familiar with the annotations and the query language

**representation** she needs a representation of the results so that these can be studied more closely or that they can be processed further with other NLP tools.

In this paper, we describe a novel approach to this problem that starts from the premise that linguistic annotations can be represented by means of existing standards developed in the Semantic Web community: RDF and OWL[1] are well-suited for data integration, and they allow to represent different corpora and tagsets in a uniform way.

We present the Tiger Corpus Navigator, an Active Machine Learning tool that allows a user to extract formal definitions of extensionally defined concepts and the corresponding examples out of annotated corpora. Based on an initial seed of examples provided by the user, the Navigator learns a formal OWL Class Definition of the concept that the user is interested in. This definition is converted into a SPARQL query[2] and passed to Virtuoso,[3] a triple store database with reasoning capabilities. The results are gathered and presented to the user to choose more examples, to refine the query, and to improve the formal definition. The data basis for the Navigator is an OWL/RDF representation of the Tiger corpus[4] and a set of ontologies that represent its linguistic annotations.

Our tool, available under `http://tigernavigator.nlp2rdf.org`, addresses and circumvents the barriers to the acquisition of knowledge out of corpora presented above:

(i) it does not need any deployment and provides a user interface in a familiar surrounding, the browser,

(ii) the concept descriptions acquired during the classifier refinement represent the (conceptual representation of the) annotations in the corpus in an explicit and readable way, and finally,

(iii) the Navigator uses OWL; the query results are thus represented in a readable, portable and sustainable way.

## 2 Tools and Resources

Several categories of tools and resources need to be integrated to enable the implementation of the goals presented above: We employ the **DL-Learner** [16] to learn class definitions for linguistic concepts; **NLP2RDF** [12] is applied for the conversion and ontological enrichment of corpus data; and the **OLiA ontologies** [5] provide linguistic knowledge about the annotations in the corpus.

### 2.1 DL-Learner

The DL-Learner extends Inductive Logic Programming to Descriptions Logics, OWL and the Semantic Web; it provides a OWL/DL-based machine learning tool

[1] http://www.w3.org/TR/rdf-concepts, http://www.w3.org/TR/owl-ref

[2] http://www.w3.org/TR/rdf-sparql-query

[3] http://virtuoso.openlinksw.com

[4] http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERCorpus

to solve supervised learning tasks and support knowledge engineers in constructing knowledge. The induced classes are short and readable and can be stored in OWL and reused for classification. OWL/DL is based on Description Logics that can essentially be understood as fragments of first-order predicate logic with less expressive power, but usually decidable inference problems and a user-friendly variable free syntax. OWL Class definitions form a subsumption hierarchy that is traversed by DL-Learner starting from the top element (*owl:Thing*) with the help of a refinement operator and an algorithm that searches in the space of generated classes. An example of such a refinement chain is (in Manchester OWL Syntax):

*(Sentence)* ⇝
*(Sentence and hasToken some Thing)* ⇝
*(Sentence and hasToken some VVPP)* ⇝
*(Sentence and hasToken some VVPP and hasToken some (stts:AuxiliaryVerb and hasLemma value "werden"))*

The last class can easily be paraphrased into: A sentence that has (at least) one Token, which is a past participle (*VVPP*), and another Token, which is an AuxiliaryVerb with the lemma *werden* (passive auxiliary, lit. 'to become'). Detailed information can be found in [16] and under `http://dl-learner.org`.

## 2.2 NLP2RDF

NLP2RDF[5] is a framework that integrates multiple NLP tools in order to assess the meaning of the annotated text by means of RDF/OWL descriptions: Natural language (a character sequence) is converted into a more expressive formalism – in this case OWL/DL – that grasps the underlying meaning and serves as input for (high-level) algorithms and applications.

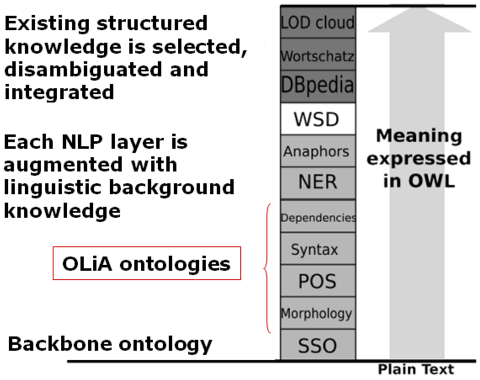


Figure 1: NLP2RDF stack

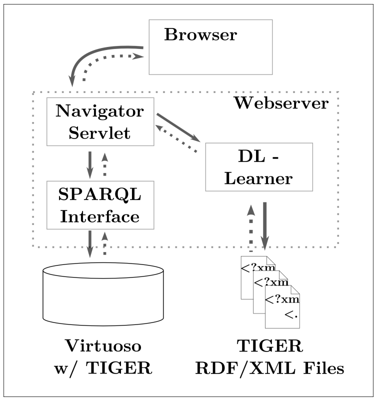


Figure 2: Architecture of the Tiger Corpus Navigator

[5] `http://nlp2rdf.org`, `http://code.google.com/p/nlp2rdf`

In a first step, sentences are tokenized and aggregated in a *Structured Sentence ontology (SSO)*. The SSO consists of a minimal vocabulary that denotes the basic structure of the sentence such as tokens and relative position of a token in a sentence.

As shown in Fig. 1, the SSO serves as the backbone model, which is then augmented additional layers of annotations:

(1) features from NLP tools
in light grey: morphology, parts of speech (POS), syntactic structures and edge labels (syntax, dependencies), named entity recognition (NER), coreference (anaphors)

(2) rich linguistic ontologies for these features (Sect. 2.3)
combined in a *tagset-ontology pair* for every level mentioned in (1)

(3) background knowledge from the Web of Data
examples in dark grey: Linking Open Data (LOD) Cloud,[6] DBPedia,[7] and Wortschatz[8]

(4) additional knowledge
knowledge created by the Navigator (Sect. 2.1) or derived from the steps described above (e.g., in white: word sense disambiguation, WSD)

### 2.3 Linguistic Ontologies

The Ontologies of Linguistic Annotations [5, OLiA] represent an architecture of modular OWL/DL ontologies that formalize several intermediate steps of the mapping between concrete annotations, a Reference Model and external terminology repositories, such as GOLD[9] or the ISO TC37/SC4 Data Category Registry:[10]

- Multiple Annotation Models formalize annotation schemes and tag sets, e.g., STTS for the part of speech tags of the Tiger corpus.

- The Reference Model provides the integrating terminology for different annotation schemes (OLiA Annotation Models).

- For every Annotation Model, conceptual subsumption relationships between Annotation Model concepts and Reference Model concepts are specified in a Linking Model. Other Linking Models specify relationships between Reference Model concepts and external terminology repositories [6].

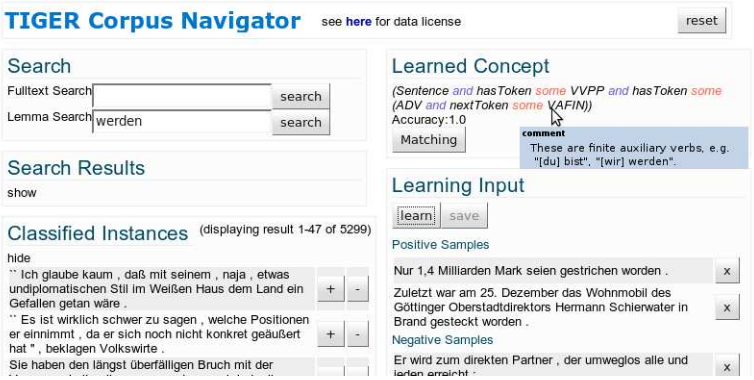
[6] http://richard.cyganiak.de/2007/10/lod

[7] http://dbpedia.org

[8] http://wortschatz.uni-leipzig.de

[9] http://linguistics-ontology.org

[10] http://www.isocat.org



Fulltext Search

Lemma Search: werden

(displaying result 1-47 of 5299)

`` Ich glaube kaum , daß mit seinem , naja , etwas undiplomatischen Stil im Weißen Haus dem Land ein Gefallen getan wäre .

`` Es ist wirklich schwer zu sagen , welche Positionen er einnimmt , da er sich noch nicht konkret geäußert hat " , beklagen Volkswirte .

Sie haben den längst überfälligen Bruch mit der

(Sentence and hasToken some VVPP and hasToken some (ADV and nextToken some VAFIN))

Accuracy:1.0

comment

These are finite auxiliary verbs, e.g. "[du] bist", "[wir] werden".

Nur 1,4 Milliarden Mark seien gestrichen worden .

Zuletzt war am 25. Dezember das Wohnmobil des Göttinger Oberstadtdirektors Hermann Schierwater in Brand gesteckt worden .

Er wird zum direkten Partner , der umweglos alle und jeden erreicht ;

Figure 3: Screenshot of the Tiger Corpus Navigator

For the current paper, we focused on the STTS Annotation Model[11] that covers the morphosyntactic annotations in the Tiger corpus.

The usage of OLiA combined with NLP2RDF offers two major advantages: OLiA provides a growing collection of annotation models for more than 50 languages, that are interlinked with the OLiA Reference Model (and further to community-maintained repositories of linguistic terminology). The adaption of the Navigator to other corpora and other languages is thus easily possible. The interlinking further allows to reuse learned classes on other corpora and even to learn on a combination of different corpora.

## 3 The Tiger Corpus Navigator

Figure 2 shows the architecture of the Tiger Corpus Navigator: The Virtuoso triple store contains the whole corpus in RDF and allows queries over the complete data for retrieval, the data used by DL-Learner consists of one file for the OWL schema and 50,474 RDF/XML files (one per sentence), which it loads on demand according to the given examples.

With the Navigator user interface (Fig. 3), the user starts his research by searching for sentences with certain lemmas or words. The retrieved sentences are presented on the left side. They can be moved to the right panel and classified as positive or negative examples, i.e., as instances or counterinstances of the target concept. Upon pressing the *Learn* button, they are sent to the DL-Learner and the learned OWL Class Definition is displayed (right top). The *Matching* button triggers the retrieval of matching sentences. The user can choose more positive and negative examples from the classified instances and iterate the procedure until the learned definition has an acceptable quality.

To aid the user during this process, the accuracy of the definition on the training

[11] available under `http://nachhalt.sfb632.uni-potsdam.de/owl`

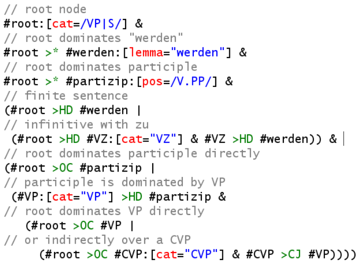
data is given below the definition. Additionally, the number of matching sentences is displayed (in this case 5,299, ≈10% of the corpus). Hovering over a named class in the concept description presents a tooltip explaining the meaning of the construct as specified by the OLiA Annotation Model. This allows to quickly gain insight into the annotations of the corpus and judge whether the learning result matches the needs of the user.

# 4 Evaluation

In this section, we evaluate recall and precision of automatically acquired concepts for passive identification in German. We describe two problems (with 4 experiments each), in which we vary several configuration options: training set size (how many examples a user needs to choose), learning time and usage of lemmas.

## 4.1 Experimental Setup

We consider the German *werden* passive that is formed by the auxiliary *werden* and a past participle [23].



```
// root node
#root:[cat=/VP|S/] &
// root dominates "werden"
#root >* #werden:[lemma="werden"] &
// root dominates participle
#root >* #partizip:[pos=/V.PP/] &
// finite sentence
(#root >HD #werden |
// infinitive with zu
 (#root >HD #VZ:[cat="VZ"] & #VZ >HD #werden)) &
// root dominates participle directly
(#root >OC #partizip |
// participle is dominated by VP
 (#VP:[cat="VP"] >HD #partizip &
// root dominates VP directly
   (#root >OC #VP |
// or indirectly over a CVP
     (#root >OC #CVP:[cat="CVP"] & #CVP >CJ #VP))))
```

Figure 4: Rule for passive sentences in the Tiger Query Language [15]

The task is to distinguish passive clauses from other auxiliary constructions, given only linguistic surface structure (SSO) and morphosyntactic annotations (POS). In the corpus, neither POS nor SSO alone are sufficient to distinguish passive from active clauses, so that information from both sources has to be combined. For our experiment, the DL-Learner was trained on POS and lemmas. Syntax annotation was used only to identify target classifications (with the query in Fig. 4).
Three sets of sentences can be distinguished:

1. finite passive (finite auxiliary *werden*, 6,333 sentences, condition `#root >HD #werden`)

2. infinite passive with particle *zu* (lit. ‘to’) (37 sentences, condition `#root >HD #VZ`)

3. active (44,099 sentences that do not match the query)

From these sets we identified two learning problems to measure how well our approach can separate these sets from each other: (i) learn an OWL class that *covers* all finite passives (set 1) and the remainder (sets 2, 3), and (ii) distinguish between infinite passives (set 2) and the remainder. The second problem is especially difficult, as the number of correct sentences (37) is less than 0.07% of sentences in total.

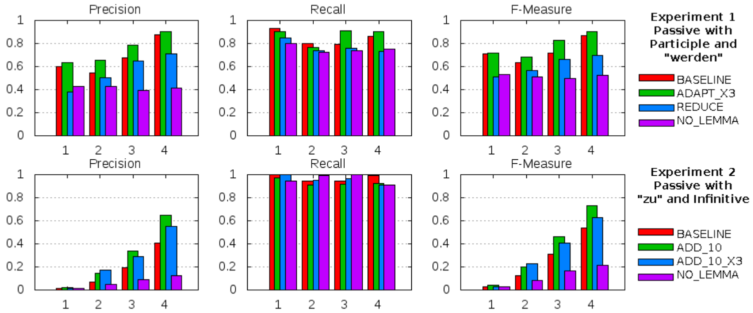


Figure 5: Evaluation results

For each problem, the data is split into training and test data (both positive and negative).[12]

As BASELINE, we randomly drew 5 positive (5p) and 5 negative (5n) sentences from the training data. In the experiments, we performed 4 iterations, starting with 5p+5n initial examples, and adding 5p+5n examples in every iteration. Precision and recall were measured on the intersection of retrieved sentences with the target classification.

We tested three configuration variations for the first problem: (1) we adapted the max. execution time to three times the number of examples (ADAPT, 30s, 60s, 90s, 120s),[13] (2) we reduced the number of initial examples to 2p+2n and added 2p+2n for each iteration (REDUCE, total 4,8,12,16), and (3) we deactivated the inclusion of *owl:hasValue* (lemmas) in the classes (NO_LEMMA).

As for the second problem, (1) we added 10 additional negative examples (ADD_10, total 20, 40, 60, 80), (2) we added 10n but adapted the runtime to 3 times the example size (ADD_10_X3, 60s, 120s, 180s, 240s), and (3) we used again the baseline (BASELINE) with no lemmas (NO_LEMMA).

For the first problem, we conducted a stratified leave-one-out 10-fold cross validation. As it was impossible to create 10 folds for the second set, we used a randomized 70%-30% split averaged over 10 runs (28 sentences for training, 11 for testing).

## 4.2 Results

Our results (summarized in Fig. 5) show that the Tiger Corpus Navigator is capable of acquiring concepts that involve multiple knowledge sources, here, the SSO (lemma) and the OLiA ontologies (for POS) with a high recall and with reasonable speed.

The observed high recall is inherent in the learning algorithm: When exploring

[12] Five overlapping sentences were removed.

[13] DL-Learner is an anytime algorithm, it stops when finding a class with 100% accuracy or a given maximum execution time (default 30 sec) is reached and returns its (intermediate) results.

the search spaces, it automatically discards all classes that do not cover all positive examples, so it produces very general results. High precision, however, can only be achieved after a certain number of iterations or by raising the *noise* parameter (zero in our experiments).

We found that our results are clearly dependent on lemmas, *owl:hasValue* inclusion yields better results. The selection of significant lemmas is done generically by DL-Learner according to a value frequency threshold, set equal to the number of positive examples. Users could also wish to manually configure this parameter or give certain lemmas in advance.

The size of the training set had a great influence on the performance with about 20% lower F-Measure in iteration 4 (REDUCE vs. ADD_10 to BASELINE). We observed marginal effects by increasing the maximum learning time with a slight F-Measure gain of 3.5% (ADAPT_X3 vs. BASELINE) and even a loss of more than 10% in the second experiment (ADD_10 vs. ADD_10_X3).

Although the second experiment amounts to a much lower F-Measure scores in iteration 4, the achieved results are interpretable: 40 % precision and 99 % recall mean that the retrieved set of sentences was reduced to about 100 sentences of which 40 would be correct. Such a small sample would be suitable for manual inspection and postprocessing.

Our implementation fulfills the speed requirements for a web scenario: For the first experiment, the average learning times for BASELINE were 1.8 sec, 22.6 sec, 31.9 sec and 29.5 sec, and for the second experiment 0.5 sec, 2.2 sec, 5.3 sec, 13.3 sec. The SPARQL queries needed 14.6 seconds on average and can be further improved by caching. The last example of the refinement chain in Sect. 2.1 was one of the highest scoring learned classes.

## 5 Related Work and Outlook

In the introduction, we identified three elementary functions a corpus tool has to fulfill, i.e., to **access**, to **document** and to **represent** linguistic annotations. We presented the Tiger Corpus Navigator, which provides *access* via a an intuitive user interface over the Web. The paradigm of navigating a corpus based on example sentences rids the necessity of being familiar with the *documentation* beforehand. Even more so, only the necessary information is presented unobstrusively on-the-fly. Learned classes *represent* the results in a formal, yet easily understandable way and the evaluation has shown that it is possible to extract the desired information without much time or effort.

### 5.1 Access to Linguistic Annotations

Linguistic corpora can be accessed by several corpus tools, e.g., GATE [9], TGrep2 [21], TigerSearch [15], the Stockholm TreeAligner [17], or MMAX2 [18], just to name a few. Newer tools also provide web interfaces, such as the IMS Corpus

Workbench [8], the Linguist's Search Engine [20], or ANNIS [7, 24].

All these tools, however, have in common that they operate on a formal, complex query language that represents a considerable hurdle to their application by non-specialists.

The Tiger Corpus Navigator represents an innovative approach to access corpus data that may complement such traditional corpus interfaces. It provides access to the primary data of specific sentences on the basis of extensionally defined conceptual descriptions, it is thus even possible to search for concepts that are not directly annotated (as shown for the passive concept and the Tiger POS annotation).

### 5.2 Document Linguistic Annotations

In our approach, linguistic annotations are explicitly documented by their linking to repositories of linguistic terminology. These repositories contain descriptions, definitions and examples that are represented to the user as tooltips (Fig. 3). In this way, the OWL representation of linguistic corpora and their linking with existing terminology repositories serves a documentation function.

And more than this, the application of the Tiger Corpus Navigator does not require the users to be familiar with the documentation at all: The automatic acquisition of query concepts allows a relatively uninformed user to run queries against a database without the necessity to be aware of the underlying data format, its expressivity and even the kind of annotations available. Thereby, our approach extends and generalizes approaches to access annotated corpora on the basis of abstract, ontology-based descriptions such as [19, 7]. As opposed to these, however, the concepts are not pre-defined in our scenario, but acquired by the system itself. The Tiger Corpus Navigator thus allows for corpus querying independently from the theoretical assumptions underlying the actual annotations in the corpus.

### 5.3 Represent Linguistic Annotations

As for exchange and representation formats, the linguistic community still struggles to define its own standards; several concurrent proposals are currently in use, e.g., NITE XML [4], UIMA XML [11], LAF/GrAF [14], or PAULA [7]. Here, standards from the Semantic Web community are applied, RDF and OWL, that are maintained by a large community and supported by a number of tools. So far, only few NLP tools working with OWL are available, e.g., [1], but a number of linguistic resources has already been transformed to OWL/DL [22, 3], or linked with ontologies [13]. Also, existing ontologies have been extended with concepts and properties for linguistic features [2, 10]. The Navigator represents another step in this development of convergence of ontological and NLP resources.

## 5.4 Application Scenarios

The Tiger Corpus Navigator may not constitute a full-fledged substitute for existing query tools, as the subsequent refinement of the classifier by the user may turn out to be a time-consuming task. It does, however, represent a prototype implementation of a technology that may be integrated with “traditional” tools to browse, query or access/distribute corpora. If used as a corpus exploration interface of an archive of linguistic resources, for example, the Tiger Corpus Navigator reduces the initial bias to assess the suitability of a corpus with unknown annotations. Such an archive may host different resources that require specialized tools for visualization and querying (e.g., TGrep2 for constituent syntax, MMAX2 for coreference, etc.), so that the efforts required to evaluate the suitability of a resource are enormous (a user has to acquaint itself not only with the annotations and some “standard” query language, but also with several specialized tools and their task-specific query languages). Using the Navigator, a user develops a classifier for a concept of interest, and the correctness of the classifier and the concept description obtained and the tooltips that contain their documentation allow her to assess the suitability of a corpus and its annotations for the task at hand immediately . If indeed a resource appears to be useful for a particular task, the user may decide to obtain the corpus and to process it further with the appropriate corpus tools.

## 5.5 Future Work

Future work includes the ability to save learned OWL classes. They can be collaboratively reused and extended by multiple users (Web2.0). Furthermore, they can be utilized to classify previously untagged text, converted by NLP2RDF in the same manner as here and thus extend the discovery of matching sentences beyond the initial corpora. With a corresponding parser-ontology pair it is even possible to replace the initial full text search by entering any example sentences.

It should be noted here that we aimed primarily for a proof-of-concept implementation. The Tiger Corpus Navigator does currently not come with an appropriate visualization, and it is restricted to sentence-level classification. Given sufficient interest from the community, the corresponding extensions may, however, be possible in subsequent research. Another topic for further research may be the combination of existing corpus management and corpus query tools with the Tiger Corpus Navigator, resp. the underlying technologies.

## Acknowledgements

We would like to thank three anonymous reviewers for helpful comments and feedback. The work of Sebastian Hellmann, Jörg Unbehauen and Axel-Cyrille Ngonga Ngomo was supported by the Eurostars SCMS project, the work of Axel-Cyrille Ngonga Ngomo additionally by the Research Focus Media Convergence at the Johannes Gutenberg University of Mainz. The work of Christian Chiarcos was supported by the German Research Foundation (DFG) in the context of Collaborative Research Center (SFB) 632 “Information Structure”, Project D1 “Linguistic Database”, University of Potsdam.

## References

[1] G. Aguado de Cea, J. Puch, and J. . Ramos. Tagging Spanish texts: The problem of “se”. In *LREC 2008*, Marrakech, Morocco, May 2008.

[2] P. Buitelaar, T. Declerck, A. Frank, S. Racioppa, M. Kiesel, M. Sintek, R. Engel, M. Romanelli, D. Sonntag, B. Loos, V. Micelli, R. Porzel, and P. Cimiano. LingInfo: Design and applications of a model for the integration of linguistic information in ontologies. In *LREC 2006*, Genoa, Italy, May 2006.

[3] A. Burchardt, S. Padó, D. Spohr, A. Frank, and U. Heid. Formalising multilayer corpora in OWL/DL – Lexicon modelling, querying and consistency control. In *IJCNLP 2008*, Hyderabad, January 2008.

[4] J. Carletta, S. Evert, U. Heid, J. Kilgour, J. Robertson, and H. Voormann. The NITE XML Toolkit: flexible annotation for multi-modal language data. *Behavior Research Methods, Instruments, and Computers*, 35(3):353–363, 2003.

[5] C. Chiarcos. An ontology of linguistic annotations. *LDV Forum*, 23(1):1–16, 2008.

[6] C. Chiarcos. Grounding an ontology of linguistic annotations in the Data Category Registry. In *LREC-2010 Workshop on Language Resource and Language Technology Standards (LT&LTS)*, Valetta, Malta, May 2010.

[7] C. Chiarcos, S. Dipper, M. Götze, U. Leser, A. Lüdeling, J. Ritz, and M. Stede. A flexible framework for integrating annotations from different tools and tagsets. *TAL (Traitement automatique des langues)*, 49(2), 2008.

[8] O. Christ. A modular and flexible architecture for an integrated corpus query system. In *Papers in Computational Lexicography (COMPLEX ’94)*, pages 22–32, 1994.

[9] H. Cunningham. GATE, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254, 2002.

[10] B. Davis, S. Handschuh, A. Troussov, J. Judge, and M. Sogrin. Linguistically light lexical extensions for ontologies. In *LREC 2008*, Marrakech, Morocco, May 2008.

[11] T. Goetz and O. Suhre. Design and implementation of the UIMA Common Analysis System. *IBM Systems Journal*, 43(3):476–489, 2004.

[12] S. Hellmann. The Semantic Gap of Formalized Meaning. In *European Semantic Web Conference (ESWC)*, Heraklion, Greece, May 2010.

[13] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. Ontonotes: the 90% solution. In *HLT-NAACL 2006*, pages 57–60, New York, June 2006.

[14] N. Ide. GrAF: A graph-based format for linguistic annotations. In *ACL-2007 Linguistic Annotation Workshop*, Prague, Czech Republic, June 2007.

[15] E. König and W. Lezius. The TIGER language - a description language for syntax graphs, Formal definition. Technical report, IMS, 2003.

[16] J. Lehmann. DL-Learner: learning concepts in description logics. *Journal of Machine Learning Research*, 2009.

[17] T. Marek, J. Lundborg, and M. Volk. Extending the TIGER query language with universal quantification. In *Proceeding of KONVENS-2008*, pages 3–14, Berlin, October 2008.

[18] C. Müller and M. Strube. Multi-level annotation of linguistic data with MMAX2. In *Corpus Technology and Language Pedagogy*, pages 197–214. Peter Lang, Frankfurt am Main, 2006.

[19] G. Rehm, R. Eckart, and C. Chiarcos. An OWL-and XQuery-based mechanism for the retrieval of linguistic patterns from XML-corpora. In *RANLP 2007*, Borovets, Bulgaria, September 2007.

[20] P. Resnik, A. Elkiss, E. Lau, and H. Taylor. The web in theoretical linguistics research: Two case studies using the Linguist's Search Engine. In *31st Meeting of the Berkeley Linguistics Society*, pages 265–276, February 2005.

[21] D. Rohde. TGrep2 user manual, version 1.15. Technical report, MIT, Cambridge, MA, May 2005.

[22] J. Scheffczyk, A. Pease, and M. Ellsworth. Linking FrameNet to the Suggested Upper Merged Ontology. In *Fourth International Conference on Formal Ontology in Information Systems (FOIS 2006)*, pages 289–300, Baltimore, November 2006.

[23] G. Schoenthal. *Das Passiv in der deutschen Standardsprache*. Hueber, München, 1975.

[24] A. Zeldes, J. Ritz, A. Lüdeling, and C. Chiarcos. ANNIS: A search tool for multi-layer annotated corpora. In *Proceedings of Corpus Linguistics*, pages 20–23, Liverpool, UK, July 2009.

# Comparing Conversions of Discontinuity in PCFG Parsing

Yu-Yin Hsu

Indiana University
Bloomington, IN, USA
E-mail: hsuy@indiana.edu

**Abstract**

This paper compares three different types of representations that resolve long-distance dependency into binary representation as it is required in PCFG parsing. Each conversion is applied to the German TIGER Treebank in the PCFG parsing experiments. The examination of data and the labeled dependency evaluation show that the choice of conversion of treebank data in the preprocessing step can influence the F-score up to 2.83% and that each conversion has its own advantages and limits. The result of this paper shows that this preprocessing step is not trivial in parsing free word order languages.

## 1 Introduction

In order to conduct a Probabilistic Context Free Grammar (hereafter PCFG) parsing, it is necessary to resolve crossing branches in the input data, because such parsers only process context-free tree structures, where no long distance relationship is allowed [11]. In the literature, crossing branches are mostly resolved by a node-raising approach (see approaches by Kübler [4]; Maier [6]; Kübler et al. [5]). Boyd [1] proposes a new approach: a node-splitting approach, i.e., mother nodes that are associated with discontinuous daughter nodes are splitted into partial nodes in order to resolve the discontinuity. She argues that such a node-splitting representation is better than the node-raising method, because the converted representation retains the original syntactic information after resolving crossing branches and thus structures are recoverable. However, no work discusses the impact of this conversion step on the parsing performance so far. Nonetheless, this preprocessing step is not trivial, given that German allows free word order and that about 30% of the sentences in the TIGER Treebanks has one or more than one crossing branch. Thus, in this paper, I focus on how different modifications of the TIGER Treebank data affect the parsing results. In addition to the aforementioned two approaches, a new approach is also considered, the node-adding approach. This method modifies the tree structure by copying the mother node information during the conversion.

The remainder of the paper is organized as follows. In section 2, I briefly introduce TIGER Treebank. In section 3, I summarize the preprocessing methods used in the experiments, and in section 4, I show the three different representations at issue. Section 5 explains the experiment followed by evaluation and discussion. Section 6 concludes the paper.

## 2 TIGER Treebank

The TIGER Treebank Release 2 contains 50 474 sentences from the German newspaper *Frankfurter Rundschau*. Part-of-speech, lemma and morphological information, phrase structures and grammatical function are annotated in the treebank. TIGER Treebank can be searched *via* TIGERSearch, which returns tree diagrams of query structures.

Among the four German treebanks that are available, NEGRA [10], TIGER [3], and the Tübingen Treebank of Written German (TüBa-D/Z) [12] are treebanks of written data. All three of them use the Stuttgart-Tübingen-Tagset [8] for part-of-speech annotation. TIGER, like NEGRA yet unlike TüBa-D/Z, does not allow unary branching and its tree structures are flatter, whereas TüBa-D/Z presents more hierarchy and it includes topological fields to avoid long distance dependencies and thus no crossing branches occur in TüBa-D/Z.

In TIGER, subjects and finite verbs are always treated as immediate daughters of the clauses, while non-finite verbs, complements of the verb, PPs and adjuncts are daughters under a single VP. It is also common to have topicalization or extraposition in German sentences. Thus, crossing branches are used in TIGER to account for such long-distance dependencies in German. Sentence structures in TIGER show less hierarchy in order to avoid structural ambiguities and to eliminate the need for traces. The distinction between adjuncts and arguments is shown through labels of syntactic functions, rather than in the structure [3]. Figure 1 illustrates an example of TIGER trees with a discontinuous constituent, i.e., the VP *Bis dahin geheimgehalten* 'until then kept secret'. Therefore, it is necessary to resolve crossing branches by a conversion of the data representation before doing PCFG parsing.

## 3 Data Preparation

Several preprocessing steps were done for the TIGER treebank data in order to prepare appropriate input files that meet the requirements of the parser. LoPar [9] was used in the experiments. It is an implementation of a parser for head-lexicalised probabilistic context-free grammars. Among 50 474 sentences in TIGER treebank, sentences with a length of more than 40 words were excluded in the experiments, because they cause problems with memory load in the parsing process. 90% of the filtered data was used as training data, 5% as development data and 5% as test data.

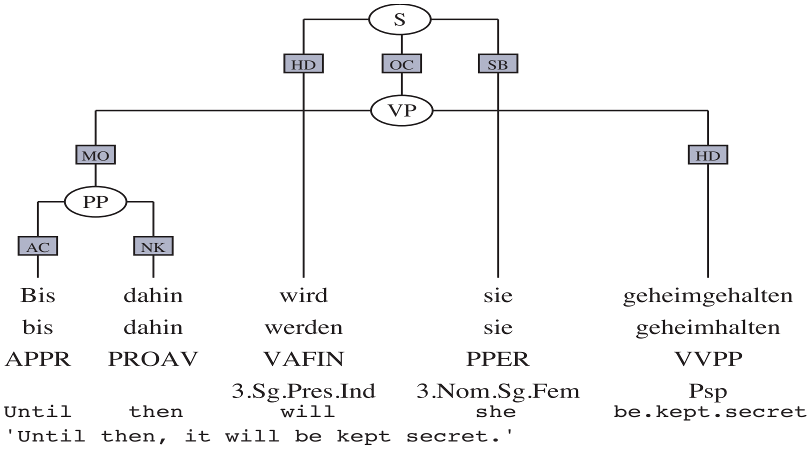

S
HD
OC
SB
VP
MO
HD
PP
AC
NK
Bis bis APPR
dahin dahin PROAV
wird werden VAFIN 3.Sg.Pres.Ind
sie sie PPER 3.Nom.Sg.Fem
geheimgehalten geheimhalten VVPP Psp
Until then will she be.kept.secret
'Until then, it will be kept secret.'

Figure 1: An example of TIGER tree

POS tags along with their labels of grammatical function were extracted directly from the treebank. Because this project focuses on the effect of different representations of crossing branches, gold standard labels were used to avoid unnecessary noise introduced by automatic POS tagging.

Next, three converted sets of treebank data, i.e., data with node-raising, node-splitting and node-adding, were created based on the same 90-5-5 split. In each conversion, a virtual root node was assigned to each sentence, and punctuations were re-attached to their local surrounding nodes.

# 4 Three Different Representations

## 4.1 Node-raising Approach

Following the node-raising approach [4] [5] [6], a script was used to detect crossing branches and to resolve crossing branches by raising non-head sister(s) higher up until no crossing branches were observed in the tree. After the raising conversion, the sentence in Figure 1 is shown in Figure 2. I.e. the prepositional phrase PP, which is the non-head daughter, is raised from the VP to the S node.

The raising approach is good in maintaining the number of nodes involved in a tree, i.e., both Figure 1 and Figure 2 have three phrases: S, VP, PP. However, after raising, the PP is reattached to a new, higher node; the mother node information (i.e., the mother node of PP-MO is VP-OC) is not available in the new structure in Figure 2. This leads to a new rule in the grammar: S-> PP VAFIN PPER VP, which did not occur before.

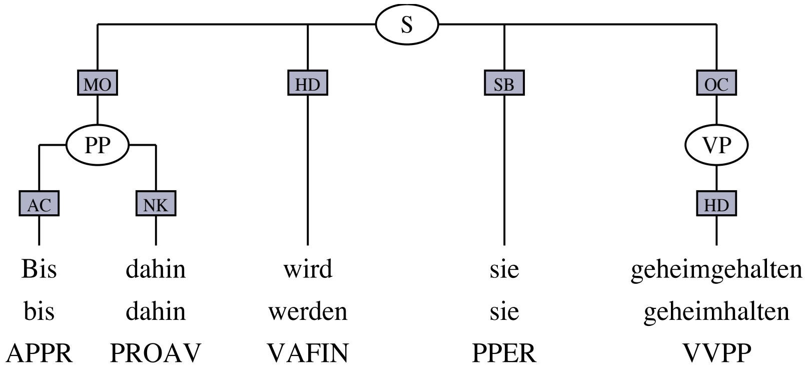


Figure 2: Node-raising conversion

## 4.2 Node-splitting Approach

Unlike the node-raising approach, Boyd [1] suggests a node-splitting representation to resolve crossing branches. That is, the mother nodes of discontinuous daughter nodes are divided into partial nodes marked with an asterisk. She argues that such representation is easily reversible and the original structural information can be maintained in this conversion. I replicated her idea of conversion; a script was used to detect long distance constituents and to split nodes that involve crossing branches. For the sentence in Figure 1, it is converted into the tree in Figure 3 after the splitting conversion. We can see that the node VP splits into two VP*s in Figure 3.

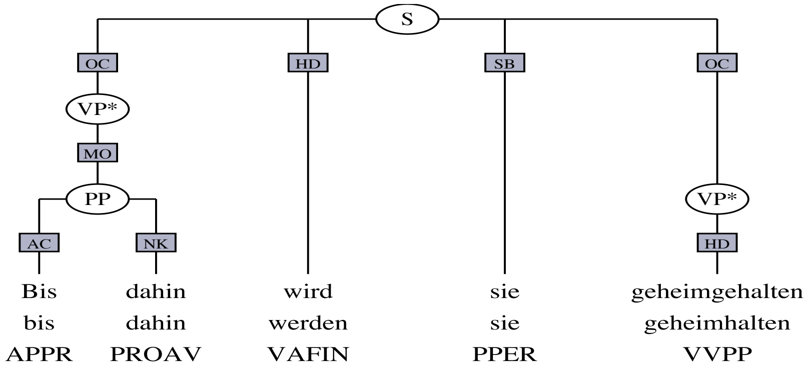


Figure 3: Node-splitting conversion

Boyd [1] also notes that in this node-splitting representation, it is easy to recover the original structure; however, if there are discontinuous nodes that use the same labels, or if there are nodes of the same categories that were split more than once, it is more difficult to find the right pair of split nodes in parsing. Figure 4 shows a parsed output that has four split nodes of VP and it would be more challenging for a program to identify which two VP*s should be combined together in order to recover the original structure. Fortunately, nodes like these can never be sisters in most of the data and thus such conversion is still reversible. An additional problem is that parsers make grouping decisions in individual processes, which means that there is no guarantee that there is always a second starred node.

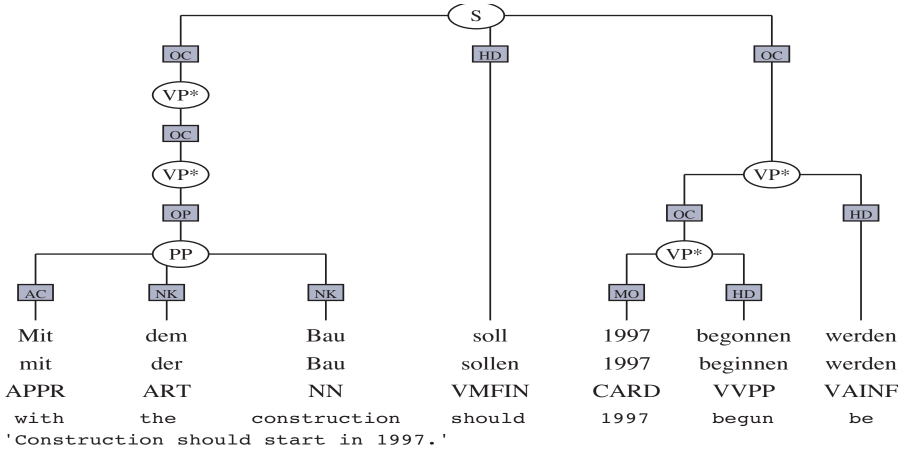


Figure 4: A tree with more than one splitting node of the same label

## 4.3 Node-adding Approach

In addition to raising and splitting approaches, another possibility that has not yet been used is to resolve crossing branches by copying the mother node information and keep such information through additional nodes in the conversion. A script was used to detect crossing branches, duplicate their mother node information, and then linked the new copy of mother node with its non-head daughter to resolve crossing branches. A similar process continued until no discontinuous constituents were found in the structure. After this conversion, the sentence in Figure 1 is represented as in Figure 5.

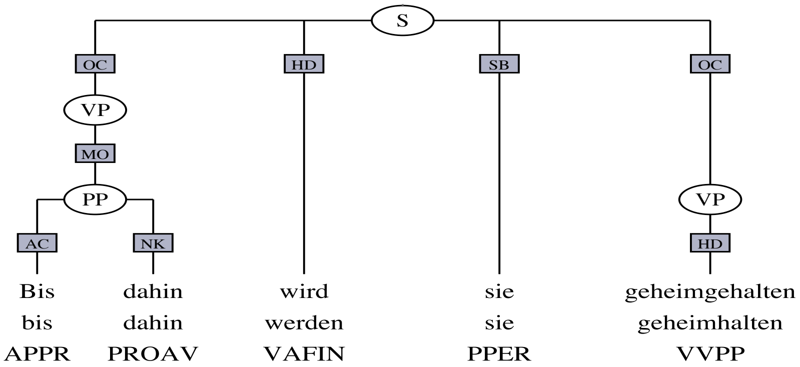


Figure 5: Node-adding conversion

This conversion as well as all other conversions introduces inconsistencies in the grammar, because new and specific unary nodes are introduced into the grammar. However, this method produces fewer inconsistencies than the node-splitting approach because it does not introduce new node labels.

# 5 Experiment

## 5.1 Results

In the LoPar parsing experiments of the TIGER Treebank data, the *viterbi* function was used to return only the best analysis of each test sentence. Given the *viterbi* output of LoPar of each conversion, a series of scripts were used to perform the PARSEVAL measures. The evaluation was computed relative to the number of brackets in a sentence structure and then returned precision, recall and F-score for labeled constituents. Overall, 28% of the test sentences involved crossing branches, and 86% of them had two crossing brackets or fewer. Table 1 shows the results of three different conversions at issue. We can see that the node-adding approach shows a precision slightly higher than the raising approach, but higher than the node-splitting approach by about 3%. In other words, both the node-raising and the node-adding approaches returned more precise parses than the node-splitting approach. In terms of recall, the raising approach is 1.12% higher than the node-adding version and it is 1.74% higher than the splitting version. Such differences can be because the new node labels introduced by the node-splitting approach cannot be estimated reliably by the parser, while the node-adding approach avoids such a problem and therefore reaches the highest precision of all methods (In section 5.2, I further discuss the effect of different representations of node labels).[1]

| | Precision | Recall | F-score |
|---|---|---|---|
| Raise | 60.00 | 64.49 | 62.16 |
| Split | 57.59 | 62.75 | 60.06 |
| Add | 60.74 | 63.37 | 62.03 |

Table 1: PARSEVAL results (%)

| | Precision | Recall | F-score |
|---|---|---|---|
| Raise | 52.94 | 57.73 | 55.23 |
| Split | 51.32 | 53.52 | 52.40 |
| Add | 54.21 | 53.37 | 53.78 |

Table 2: PARSEVAL results of sentences with crossing branches(%)

The same evaluation process then was carried out on only sentences with crossing branches (i.e., 28% of the test sentences). The results are shown in Table 2.

[1] As it is pointed out by a reviewer, we acknowledge that the standard evaluation measures are not perfect and the measure used in this study is specific to one type of treebank formats. The impact on changing the number of nodes in the structure is discussed in section 5.2. It is possible to include other types of measures, such as a dependency evaluation (as discussed by Boyd and Meurers [2]). However, such measures rely on extracting information from gold standard as well as the parser output. Rehbein and van Genabith [7] showed that the conversion from gold standard constituents is very reliable, but it is unclear how reliable it is for parser output, which may contain unexpected structures that will lead to conversion errors. Thus, we leave the comparison of different types of evaluation measures for future work.

Similar to the overall results, the node-adding version shows the highest precision and the raising approach shows the best recall among the three conversions. However, the node-splitting conversion shows a recall slightly higher than the node-adding conversion, but it is much worse than the node-adding conversion in precision. Results are all lower than 60% and the node-raising version shows the best F-score, 55.23%, which is 1.45% higher than the node-adding version, and is 2.83% higher than the node-splitting version. The results show that the choice of conversion in the preprocessing affect the parsing of sentences with crossing branches and the difference could be up to 2.83%. For conversions that preserve original structural information, the node-adding approach shows an F-core about 2% higher than that of the node-splitting approach. In sum, both node-raising and node-adding conversions show better F-score than the node-splitting representation.

### 5.2 Rule Types

The lower scores of the splitting version may be because of the asterisk shown on labels that leads to more unique rules and affects the frequency of rules, and in turn influence the parsing results. I investigated this assumption by looking at the number of rules extracted form the training data by the different methods. The results reported in Table 3. Taking the raising approach as the baseline, we can see that the node-splitting approach has the largest set of phrase structure rules (7.9% more than the raising approach), which is higher than the number of rules created by the node-adding approach (6.7% more than the raising version).

| Raise | Split | Add |
|---|---|---|
| 261682 | 282315 | 279293 |
| | +7.9% | + 6.7% |

Table 3: The number of phrase structure rules in the training data

These differences also reflect on the numbers of rule types for major phrase types. As shown in Table 4, although the exact rules in each set of representation are not identical, in terms of the number of rule types of the major categories, the raising and adding-node approaches have similar numbers of rule types of these categories, but splitting-node approach creates additional rules. Numbers in parentheses indicate the number of rules having partial nodes. Node-splitting conversion increases rule types of each category, and this change also affects the frequency of rules. Table 5 shows the top five frequent rule types in each set of data. Although both node-splitting and node-adding conversions increase the number of nodes in the modified structure, the node-adding approach does not change the ranking of rule frequency much, but the node-splitting approach shows more effects on the frequency of rules and the parsing performance.

|  | PP | VP | S |
|---|---|---|---|
| Raise | 24 | 16 | 18 |
| Split | 32 (8) | 21 (5) | 29 (11) |
| Add | 24 | 16 | 18 |

|  | AP | AVP | NP |
|---|---|---|---|
| Raise | 20 | 22 | 29 |
| Split | 27 (7) | 29 (8) | 46 (17) |
| Add | 20 | 22 | 29 |

Table 4: Rule types

| Frequency<br>Rank | Split<br>Rule Type | Add<br>Rule Type | Raise<br>Rule Type |
|---|---|---|---|
| 1 | PP-MO | VP-OC | PP-MO |
| 2 | NP-SB | PP-MO | VP-OC |
| 3 | VP-OC | NP-SB | NP-SB |
| 4 | VP*-OC | NP-OA | PP-MNR |
| 5 | PP-MNR | PP-MNR | NP-OA |

Table 5: Top five rule types in the frequency rank

| Train |  | Test |  |
|---|---|---|---|
| Phrase | GR | Phrase | GR |
| VP 50.5 | OC 51.9 | VP 49.6 | OC 51.6 |
| PP 16.3 | MO 15.1 | NP 17.3 | MO 14.9 |
| NP 15.5 | SB 8 | PP 15.8 | SB 8 |

Table 6: The most frequently modified nodes (%)

### 5.3 Modified Nodes

Boyd [1] reports that in the splitting conversion, VP (about 55%) and NP (about 20%) are the most frequently split nodes. In this project, a similar phenomenon is found. Table 6 summarizes the most frequently modified phrase types and the grammatical functions (GR). In both train and test data, VP is the most frequently modified category (about 50%); PP and NP come as the second and the third frequent categories. In terms of the grammatical function, clausal objects (OC) are those that underwent modification most frequently (about 52%), and then modifiers (MO) come as the second (about 15%), and the subject (SB) the third. Among nodes being affected in the conversion process, words with labels of VP-OC, PP-MO, NP-OA and PP-OP are the most common nodes that underwent changes. This, in fact, reflects the linguistic properties of German that VPs often involve long-distance dependency, and that PPs and objects often involve extraposition, and thus they are the targets in the conversion process more frequently.

### 5.4 Errors in the Parses

In the parsing output, the most common errors are found with PP modifiers, clausal VP objects, subjects and objects. PPs that are nominal modifiers were often parsed as general PPs, and a general PP might be identified as nominal modifiers sometimes. NP errors were found mostly when the object precedes other NPs in the sentence and was parsed as the subject. Constituent errors happened mostly with the VP clausal object. These errors are not easily avoidable, since the function *viterbi* is calculated based on the probability of rules, and VP-OC rules have a dominating frequency in the data, comparing to other VP rules or clausal rules (cf.

Table 5). Errors of PP adjunction are also predictable based on the frequency of rules, i.e., PP-MO rules have a frequency higher than PP-MNR in all three versions. In addition to these general parsing errors, the node-splitting conversion shows a different problem. Since *viterbi* is calculated based on the probability of rules, the parser does not know that it is necessary to match partial nodes into a complete node in parsing. Therefore, some parses of the splitting version show more partial nodes than necessary. Figures 6 to 8 demonstrate this problem. Figure 6 is one of the sentences in TIGER that involves crossing branches, i.e., the VP has a PP at the beginning of the sentence and the past participle *übriggeblieben* 'left over' at the end of the sentence. Ideally, the crossing branches in this sentence would be resolved through the node-splitting conversion, as in Figure 7, where both the initial PP and the final participle *übriggeblieben* show the same mother node information, i.e., VP*-OC.

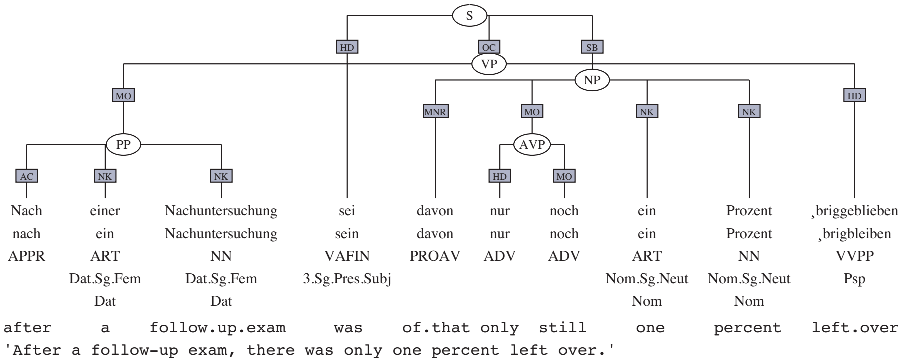

S
HD
OC
SB
VP
NP
MO
MNR
MO
NK
NK
HD
PP
AVP
AC
NK
NK
HD
MO
Nach nach APPR
einer ein ART Dat.Sg.Fem Dat
Nachuntersuchung Nachuntersuchung NN Dat.Sg.Fem Dat
sei sein VAFIN 3.Sg.Pres.Subj
davon davon PROAV
nur nur ADV
noch noch ADV
ein ein ART Nom.Sg.Neut Nom
Prozent Prozent NN Nom.Sg.Neut Nom
¸briggeblieben ¸brigbleiben VVPP Psp
after a follow.up.exam was of.that only still one percent left.over
'After a follow-up exam, there was only one percent left over.'

Figure 6: An example of original TIGER trees

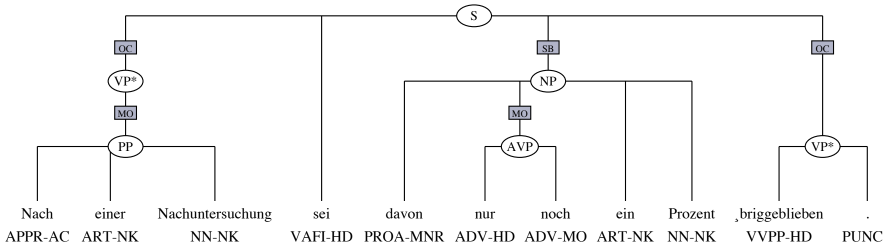


Figure 7: The expected splitting modification of Figure 6

However, since the probabilistic calculation of the best parse is independent of the structure, the parser does not know it is important to match partial nodes in the parsing. For the sentence in Figure 6, the parser returned the structure in Figure 8. We can see that it returned more smaller constituents (and this tendency is less observable in the other two approaches). In addition, a partial node, NP*-PD, for the word *davon* 'of that' occurs in Figure 8, but there is no corresponding NP*-PD partial node in the structure.

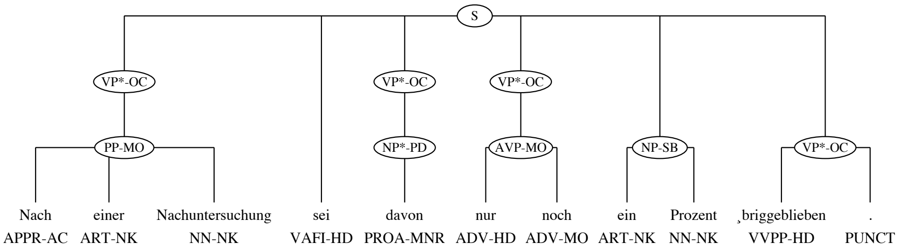


Figure 8: The actual node-splitting parse of the sentence in Figure 6

## 6 Conclusion and Future Work

The results of the experiments show that the choice of a conversion algorithm influences the parsing results by up to 3%. The experiment reports a 1-2% difference in recall, a 3% difference in precision and overall a 2% difference in the F-score when a different modification process is chosen. Although both the node-adding and the node-splitting conversions try to maintain the original information in the structure, the node-spitting version is recoverable, but it is harder to recover the original trees from the node-adding conversion, since there is no indication showing which nodes should be combined together. In terms of recoverability, the node-splitting approach is better than the other two modifications, but in maintaining the original structural information and in terms of the parsing performance, the node-adding approach seems to be preferred. To further improve the parsing results, we may consider adding morphological information. This extra information would be most helpful to distinguish object NP from subject NP in parsing, and avoid the bias introduced by linear order of words. Theoretically, this can also help identify PP adjunctions. I leave these possibilities for the future work.

## Acknowledgements

I am grateful to Sandra Kübler for her insightful suggestions, comments and support for this project. I would also like to thank the anonymous reviewers for their valuable comments.

## References

[1] Boyd, Adriane (2007) Discontinuity revisited: An Improved Conversion to Context-free Representations. In *Proceedings of the Linguistic Annotation Workshop (LAW 2007)*, Prague, Czech Republic.

[2] Boyd, Adriane and Detmar Meurers (2008) Revisiting the Impact of Different Annotation Schemes on PCFG Parsing: A Grammatical Dependency Evaluation. In *Proceedings of the ACL'08 Workshop on Parsing German*, Columbus, Ohio.

[3] Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith (2002) The TIGER Treebank. In *Proceedings of the First Workshop on Treebank and Linguistic Theories (TLT 2002)*, Sozopol, Bulgaria.

[4] Kübler, Sandra (2005) How Do Treebank Annotation Schemes Influence Parsing Results? Or How Not to Compare Apples and Oranges. In *Proceedings of the Fifth International Conference on Recent Advances in Natural Language Processing (RANLP 2005)*, pages 293–300, Borovets, Bulgaria.

[5] Kübler, Sandra, Erhard W. Hinrichs, and Wolfgang Maier (2006) Is It Really That Difficult to Parse German? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, Sydney, Australia.

[6] Maier, Wolfgang (2006) Annotation Schemes and Their Influence on Parsing Results. In *Proceed- ings of the ACL-2006 Student Research Workshop*, Sydney, Australia.

[7] Rehbein, Ines and Josef van Genabith (2007) Treebank Annotation Schemes and Parser Evaluation for German. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic.

[8] Schiller, Anne, Simone Teufel, and Christine Thielen (1995) *Guidelines fü das Tagging deutscher Textkorpora mit STTS*. Universität Stuttgart, Universität Tübingen, Germany.

[9] Schmid, Helmut (2000) Lopar: Design and Implementation. Technical report, Universität Stuttgart, Germany.

[10] Skut, Wojciech, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit (1997) An Annotation Scheme for Free Word Order Languages. In *Proceedings of the Fifth conference on Applied Natural Language Processing (ANLP)*, Washington, D.C.

[11] Telljohann, Heike, Erhard W. Hinrichs, and Sandra Kübler (2004) The TüBa-D/Z treebank: Annotating German with a Context-free Backbone. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 2229–2235, Lisbon, Portugal.

[12] Telljohann, Heike, Erhard W. Hinrichs, Sandra Kübler, and Heike Zinsmeister (2005) *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)*. Universität Tübingen, Germany.

# Adding Context Information to Part Of Speech Tagging for Dialogues

Sandra Kübler, Matthias Scheutz, Eric Baucom, Ross Israel

Indiana University
Bloomington, IN, USA
{skuebler,mscheutz,eabaucom,raisrael}@indiana.edu

**Abstract**

Part-of-speech (POS) tagging for English is often considered a solved problem, with accuracies for POS tagging the Penn Treebank of around 97%. However, POS tagging generally assumes that there is a large in-domain training set available, and that the domain is carefully edited written language. We investigate the performance of Markov model and maximum entropy POS taggers given a small data set of spontaneous dialogues in a collaborative search task. We investigate whether adding information about the speaker or about the dialogue move of the sentence can improve results. Our experiments show that especially the dialogue move information increases accuracy, but the information must be provided in a way that does not cause data sparseness issues. Our best results of 96.55% were reached by an extension of the maximum entropy tagger that uses the dialogue information as additional features in classification.

## 1 Introduction

Part-of-speech (POS) tagging for English is often considered a solved problem. There are well established approaches such as Markov model trigram taggers [1], maximum entropy taggers [10], or Support Vector Machine based taggers [7], and accuracy is around 97%. However, most experiments in POS tagging for English have concentrated on data from the Penn Treebank [9], i.e., based on a well defined genre (financial news) of carefully edited language and with a large training set. In this paper, we investigate POS tagging for a small corpus of spontaneous dialogues, CReST [6], based on an experiment in which humans perform a cooperative, remote search task. This corpus challenges both assumptions that are made when using the Penn Treebank: We have a very limited data set, and since the corpus is based on spontaneous dialogues, the language is more casual than in the Penn Treebank. I.e. this data set exhibits all the characteristics of spontaneous speech, including hesitations, false starts, corrections, and word replacement. Under these conditions, POS taggers do not reach state-of-the-art results. For example, TnT, a Markov model POS tagger reaches 96.7% accuracy on the Penn Treebank [1] but approximately 2 percent points less, 94.8% when trained and tested on the CReST corpus. If TnT is trained on the Penn Treebank and tested on the CReST corpus, it reaches an accuracy of only 85.5%. Consequently, there are significant differences between the Penn Treebank and CReST, which cannot be counterbalanced by the shorter and less complex sentences in CReST, which should be easier to analyze.

In interpreting these numbers, we have to take into consideration that a wrong choice in the POS tag of an ambiguous word will negatively affect following analysis steps such as syntactic parsing. One typical error in our experiments is the confusion of adverbs and adjectives. If an adjective is erroneously tagged as an adverb, then the parser may be forced to resort to a different subcategorization frame for a verb, thus changing the analysis for the complete sentence. This means that even a 2% drop in POS tagging accuracy will result in a serious decrease in performance of a parser that uses the POS tagged text as input.

In order to improve POS tagging results for small training corpora, we investigated whether adding information about the context improves tagging accuracy. The context information that we included was dialogue model information, and more specifically information about the speaker (director or searcher), information about the dialogue move, and a combination of both. Our hypothesis is that the dialogue model influences the types of sentences uttered at a certain point in the dialogue, and consequently also the types of POS tags. We assume that since dialogue moves tend to be associated with certain syntactic structures, they would also help predict certain sequences of tags. For instance, if the move is INSTRUCT, the sentence is more likely to be a command, which informs the POS sequence, especially at the beginning of the sentence. We also assume that speaker information would help predict the tag sequence because different speaker roles tend to produce different syntax structures, especially when the roles are as distinct as director and searcher, which was the case in our corpus. The searcher, for example, may be more likely to ask questions, which have different POS sequences. Our goal in comparing different POS taggers and different ways of integrating the information into the POS tagging process was to find the optimal approach for integrating additional context information into the POS tagging process.

Our results show that adding the additional information is indeed helpful, but only if it is integrated as additional features and not in the POS tags themselves. We also show that maximum entropy tagging is better suited for integrating new types of information.

The remainder of the paper is structured as follows: In section 2, we discuss related approaches, and in section 3, we discuss the corpus in more detail as well as the POS taggers used for the experiments. Section 4 outlines our experimental methodology. We present our results in section 5, and we finish with our conclusion and future work in section 6.

## 2 Related Work

There is a vast literature on many facets of POS tagging. For the purpose of the research presented here, we concentrate on POS tagging approaches that go beyond using 1-2 words on either side of the focus word as information in deciding the POS tag of the focus word. To our knowledge, the only statistical POS tagging approach that does use additional features is the one based on maximum entropy models. Maximum entropy taggers allow for a rich, linguistically motivated feature set to be used in a statistical framework. Accordingly, there have been a number of maximum entropy POS taggers developed in an attempt to further improve upon the accuracy that can be achieved by other approaches.

Ratnaparkhi [10] describes a maximum entropy approach to POS tagging. The tagger learns a log-linear conditional probability model from a training corpus of tagged text using a maximum entropy classifier. Along with contextual features looking at the surrounding words and tags, there are a number of features based on the form of the word, including the nature of affixes and the inclusion of hyphens, apostrophes, numbers, and capital letters. Ratnaparkhi reports 96.6% accuracy on unseen data from the Wall Street Journal.

Toutanova and Manning [13], also working with a maximum entropy tagger, used the model laid out by Ratnaparkhi [10] as a starting point, and raised the effectiveness of the tagger to deal with unknown words, in particular. They added a set of features designed to help identify proper nouns, another set to disambiguate verb forms, and a set to disambiguate participles, adverbs, and prepositions more accurately. By adding these features and excluding some that were used by Ratnaparkhi, Toutanova and Manning were able to achieve higher accuracy overall and specifically for unknown words on the same test set from the Wall Street Journal.

Inspired by the work of Ratnaparkhi and Toutanova and Manning, Denis and Sagot [5] developed another implementation of a maximum entropy tagger, MElt, that they applied to the task of POS tagging French. The authors developed a superset of features that combined those utilized previously. The full set of features is shown in Table 1. Denis and Sagot altered the algorithm by Ratnaparkhi by lifting the restriction that so called lexical features, i.e. features that examine the composition of words, should only apply to rare words. They also added an external lexical resource, *Lefff* [11], to be used in concert with the dictionary learned from the training data. MElt is described further in the following section.

## 3 Corpus and POS Taggers

### 3.1 The CReST Corpus

The CReST corpus [6] is a corpus of natural language dialogues obtained from humans performing a cooperative, remote search task in which one person outside the search environment (director) directed a person inside the environment (searcher). The director guided the searcher through the search environment, for

| **Lexical features** | |
|---|---|
| $w_i = X$ | & $t_i = T$ |
| Prefix of $w_i = P, \|P\| < 5$ | & $t_i = T$ |
| Suffix of $w_i = S, \|S\| < 5$ | & $t_i = T$ |
| $w_i$ contains a number | & $t_i = T$ |
| $w_i$ contains a hyphen | & $t_i = T$ |
| $w_i$ contains an uppercase letter | & $t_i = T$ |
| $w_i$ contains only uppercase letters | & $t_i = T$ |
| $w_i$ does not start a sentence and contains an uppercase letter | & $t_i = T$ |
| **Contextual features** | |
| $t_{i-1} = X$ | & $t_i = T$ |
| $t_{i-2}t_{i-1} = XY$ | & $t_i = T$ |
| $w_{i+j} = X, j \in -2, -1, 1, 2$ | & $t_i = T$ |

Table 1: Denis and Sagot's MElt features.

which the director had a map, in order to find different colored boxes, enter them on the map, and place blocks in them. The director was fitted with a free-head eyetracker, and he was recorded by a microphone positioned between the director and the telephone's speaker. The searcher wore a helmet with a cordless phone and a light-weight digital video camera that recorded his or her movement through the environment as viewed from his or her perspective and provided a second audio recording of the spoken dialogue.

The multi-modal corpus consists of seven dialogues. The text highlights the differences between formal written and naturally occurring language, as it is rife with directives, disfluencies, corrections, ungrammatical sentences, wrong-word substitutions, and various other constructions that are missing from written text corpora. In total, there are 11 317 words in 1 977 sentences.

The corpus contains the speech signals as well as transcriptions of the dialogues, which are additionally annotated for dialogue structure, disfluencies, and for syntax. The syntactic annotation comprises POS annotation, Penn Treebank [9] style constituent annotations, as well as dependency annotations based on the dependencies of *pennconverter* [8].

### 3.2 Annotation

On the dialogue level, the corpus was annotated for dialogue structure and for disfluencies. Utterances were divided into separate dialogue moves, based on the classification developed by Carletta et al. [2] for coding task-oriented dialogues. Their scheme views utterances as moves in a conversational game and classifies utterances into three basic move categories: *Initiation*, *Response*, and *Ready*. *Initiation* is further divided into INSTRUCT, EXPLAIN, QUERY-YN, QUERY-W, CHECK, and ALIGN. The category *Response* includes ACKNOWLEDGE, replies to wh-questions REPLY-WH, and yes or no replies REPLY-Y, REPLY-N.

|  |  |  |  |
|---|---|---|---|
| yeah | AP | you | PRP |
| let | VBI | ’re | VBP |
| ’s | PRP | gonna | VBG+TO |
| do | VB | find | VB |
| that | DDT | a | DT |
| yeah | UH | pink | JJ |
|  |  | box | NN |

Figure 1: Two examples with POS annotation.

The POS annotation is based on the Penn Treebank POS tagset [12], with a small number of new POS tags added to describe typical characteristics of spoken language:

- **AP** for adverbs that serve for answering questions, such as yes, no, or right.
- **DDT** for substituting demonstratives, such as in that is correct.
- **VBI** for imperatives, such as turn left.
- **XY** for non-words or interrupted words.

The first sentence in Figure 1 shows an example of a sentence with three new POS tags.

Another modification of the tagset concerns informal contractions such as in you ’re gonna wanna turn to the right?, which are kept as single words. As a consequence, they are assigned combinations of tags, such as **VBG+TO**. The second sentence in Figure 1 shows an example of such a contraction.

### 3.3 POS Taggers

We used two different POS tagging approaches, Markov models, and maximum entropy models. In the following, we give a short overview of the individual implementations and their characteristics.

**TnT.** TnT [1] is a trigram Markov model POS tagger with state-of-the-art treatment of unknown words. It can be trained on new data sets, and the implementation allows setting parameters such as the order of the Markov model, but it is impossible to add new types of data because the source code for the POS tagger is not available.

**IncT.** In order to incorporate new types of information, we used our own re-implementation of an incremental trigram Markov model POS tagger. The trigram model is interpolated with unigram and bigram models, using $\lambda$ values taken from TnT's optimization. For handling unknown words, IncT uses a simpler version of TnT's suffix trie in combination with Chen-Goodman smoothing [3].

**MElt.** For a maximum entropy tagger, we chose the Maximum-Entropy Lexicon-Enriched Tagger, MElt [5]. MElt is a conditional sequence maximum entropy POS tagger that uses a set of lexical and context features, which are a superset of the features used by Ratnaparkhi [10] and Toutanova and Manning [13]. The features are handled by the MegaM maximum entropy package [4]. The implementation, including the source code, is available from sourceforge.

**MElt+.** In order to integrate new types of information, we modified the MElt source code to add any features that accompany a sentence in a comment line at the beginning of the sentence. The modification only adds these new features so that there is no change in performance or accuracy when no features are added.

## 4 Experiments

We used the seven dialogues as folds in a 7-fold cross-validation. Evaluation was performed on the concatenation of the test data sets, i.e. on the whole data set. As a baseline, we used all POS taggers without modification. This means that MElt and MElt+ are identical, and we report only 3 results. Then we added the dialogue information about the speaker and the dialogue move assigned to the sentence. Both types of information were extracted from the multi-modal corpus annotation. In a first experiment, we added this information to the POS tags, thus creating complex POS tags. This approach has the advantage of not requiring any modification of the POS taggers. In a second experiment, we added the new type of features directly to the algorithms. For these experiments, we experimented with adding the speaker information, the dialogue move information, and a combination of both types of information.

The evaluation was performed on POS tags only; in the experiments using the complex tags, we used the complex tags for training and testing, but for evaluation, we stripped off the additional information and evaluated on the POS tags only. One reason for this procedure is that we needed to ensure comparability between experiments. The more important reason is that we are not interested in how accurately the POS tagger can predict the additional information but rather in whether the additional information is useful for tagging and whether it can be successfully built into the POS tagging process.

### 4.1 Complex POS Tags

The simplest approach to adding the new information is to add it to the POS tags themselves, thus creating more complex tags. For example, given the word-tag combination `left/VBN` spoken by the director during a question move, we create the complex tag `left/VBN_director` when only speaker information is added, `left/VBN_question` with dialogue move information, and `left/VBN_director_question` with both types of information.

Adding the new information to the POS tags increases the size of the tagset and therefore also the risk of data sparseness: While the original POS tagset contains 38 different POS tags, adding the speaker information increases the tagset to 74 tags, and adding the dialogue moves results in a large tagset of 515 different tags. Adding a combination of both types of information results in a tagset of 772 tags. It is obvious that the latter two tagsets can very easily result in data sparseness problems, given that we only have small corpus of 11 317 words.

### 4.2 Modified Algorithms

In order to avoid data sparseness issues with the extended data sets, we pursued a second approach in which we integrated the new information into the algorithms directly. For MElt, this conversion to MElt+ was relatively simple: MElt uses a maximum entropy classifier in the background. Thus, for each word, an instance with independent features is passed to the classifier, which then makes the decision which POS tag should be assigned to the word based on the features. Our modification of the algorithm consists of passing the new types of information to the classifier as additional features.

For IncT, the modification was more extensive. To integrate the new information into the Markov model, we replaced the standard sentence boundary marker by a set of such markers, which model the additional features. Thus, in the training phase, counts were tabulated of how often certain POS tags occurred at the beginning and end of the sentences, in the context of certain dialogue moves and/or speaker types. For the combination of both types of information, we first used a solution in which we combined the labels into a single tag, e.g. `QYN_Searcher` for a yes/no question uttered by the searcher. Since this led to data sparseness issues, we then modified the approach so that the first sentence boundary marker at the beginning of the sentence represents the speaker, and the second sentence boundary the dialogue move. For the sentence boundary marker, we chose again the dialogue move. Thus the trigrams extracted from the INSTRUCT sentence `turn/VBI left/RB` uttered by the director are shown in Figure 2. The sentence boundary markers start with a $ sign.

| | | |
|---|---|---|
| $Director | $INSTRUCT | VBI |
| $INSTRUCT | VBI | RB |
| VBI | RB | $INSTRUCT |

Figure 2: The trigrams extracted from the sentence `turn/VBI left/RB`.

## 5 Results

The results of the experiments described above are shown in Table 2. The first baselines, in which the POS taggers were trained on the Penn Treebank and tested on CReST, show that both the trigram and the maximum entropy tagger do not perform well out of domain; TnT reached 85.49%, and MElt, which relies more heavily on lexical features, reached 83.31%. Since the initial results were so low, we refrained from repeating this experiment with IncT. From these experiments, we can conclude that using an existing model trained out-of-domain does not provide useful results. When the taggers are trained on CReST in 7-fold CV, the baseline shows that although the taggers were trained on a small data set in the order of 9 700 words, they reached results that are only slightly lower than results reported on the Penn Treebank (Brants [1] reports an accuracy of 96.7% on this data set). On our data set, TnT reached 94.80% while MElt reached a slightly higher accuracy of 95.64%. Our own trigram Markov model tagger, IncT, reached an accuracy that is comparable to TnT's 94.50%. The slight difference can be explained by the taggers' different strategies for handling unknown words.

| | | TnT | IncT | MElt | MElt+ |
|---|---|---|---|---|---|
| Baseline | Trained on Penn | 85.49 | * | 83.31 | |
| | Trained on CReST | 94.80 | 94.50 | 95.64 | |
| Complex Tags | Dialogue Move | 94.42 | 89.28 | 94.70 | |
| | Speaker | 94.81 | 93.57 | 95.39 | |
| Modified Algorithm | Dialogue Move | * | 95.03 | * | 96.55 |
| | Speaker | * | 94.52 | * | 95.74 |
| | Dialogue Move & Speaker | * | 94.98 | * | 96.55 |

Table 2: Results of the POS tagging experiments

In the experiments reported as **Complex Tags**, we added the additional information to the POS tags, thus creating complex tags. A closer look at the table corroborates our assumption that such a procedure leads to data sparseness. All taggers performed worse than in the baseline experiments. The only exception is the experiment in which TnT was confronted with POS tags that contained speaker information. In this experiment, TnT reached a non-significant[1] improvement of 0.1 percent points over the baseline. Adding speaker information results in a smaller

[1] McNemar, $p < 0.001$.

loss of accuracy for all taggers than adding dialogue moves. The reason for this difference can be found in the potential increase in POS tags that adding dialogue moves causes. Speaker information consists of 2 labels, director and speaker. Thus it can maximally double the initial POS tagset of 38 tags, and almost does: We observed 74 out of the 76 possible tags. Adding dialogue moves, in contrast, supplies 47 new labels, which can maximally create $38 * 47 = 1\ 786$ complex labels. Even though the actual number is considerably lower at 515, there is still an increase by more than a factor of 13. The fact that this major increase in tags only results in losses in accuracy of less than 1 percent points for TnT and MElt shows that the new information must provide useful information. However, while TnT and MElt suffered minimally, adding the dialogue move information for IncT resulted in a considerable loss of accuracy. The tagger only reached an accuracy of 89.28%, which is more than 5 percent points lower than the baseline accuracy.

Since both types of information result in lower accuracies, we refrained from adding both types simultaneously. This would have increased the size of the tagset even more and thus exacerbated the data sparseness problem. Instead, we investigated whether the information can be successfully integrated into the algorithms. The results of these experiments are reported as **Modified Algorithm**. Since it is not possible to use the original implementations for these experiments, we report results only for IncT and for MElt+.

A closer look at the results of these experiments shows that adding speaker information results in a non-significant improvement for both MElt+ and IncT. The error reduction for MElt+ is 2.4% and for IncT 0.3%. In contrast, adding dialogue move information results in a significant increase for both taggers, with an error reduction of 20.9% for MElt and 9.5% for IncT. This increase shows that dialogue information is more useful in POS tagging CReST than speaker information. Why speaker information is not more helpful is not immediately clear. However, the setup of the search scenario is such that both speakers need to collaborate to perform the tasks. This means that both speakers ask questions or give directions during the completion of the task. For example, the corpus contains 45 questions asked by the director and 89 questions asked by the searcher. However, there are large individual differences between the dialogues; in two dialogues, the director asks more questions than the searcher. Thus, knowing whether a word is part of a question or of an explanation is more useful than knowing which speaker uttered the sentence.

A very clear example where the dialogue moves provide useful information for POS tagging is the word `yeah`. This word is assigned the POS tag `AP` when it occurs in an answer to a yes/no question, i.e. when it is part of a REPLY-Y or REPLY-N move, and it is assigned the POS tag `UH` when it belongs to any other move. There is only one exception this rule: In the sentence `yeah let 's do that yeah`, the second `yeah` is tagged `UH` in spite of being in a REPLY-Y move. In addition, the confusion between `AP` and `UH` is the largest source of errors in our experiments. Adding speaker and dialogue move information results for this word in an error reduction of 91.1% for MElt+ and of 91.3% for IncT.

While adding individual information is either beneficial or slightly detrimental, the picture is much clearer in the experiment where both types of information are added: IncT has a minimal decrease in accuracy to 94.98% in comparison to adding only dialogue move information, MElt+ reaches the same accuracy as in the experiment with dialogue moves, 96.55%. In comparison to the in-domain baseline, however, IncT reaches an error reduction of 9.8%, and for MElt+, there is an error reduction of 20.9%.

This shows that both types of information are potentially useful for POS tagging dialogue data. However, the information must be integrated in a way in which a POS tagger can successfully use the information without encountering data sparseness. Since MElt+ adds both types as individual features, no data sparseness ensues. IncT, in contrast, must use a combination of both tags and thus cannot avoid data sparseness. This leads us to conclude that adding the additional information as sentence boundary markers is not viable. Instead, the information must be integrated into the transition probabilities.

## 6 Conclusion and Future Work

In this paper, we have shown that for dialogue data such as in the CReST corpus, adding information about the speaker and about the dialogue moves improves tagging results. Especially, dialogue move information provides valuable disambiguation information for words that can be ambiguous between different categories that primarily occur in certain dialogue moves. However, in order to avoid data sparseness, we had to provide the data not as part of complex POS tags but rather as information inside the POS tagging algorithm. The results show that adding features to a maximum entropy tagger results in higher accuracy than adding them as sentence boundary markers in a Markov model tagger.

For the future, we are planning to modify IncT, the incremental Markov model tagger, so that the calculation of the transition probabilities is not conditioned on the previous context words but also on the additional information. This modification will allow us to use the additional information in all decisions. In order to avoid data sparseness, we will also modify the interpolation model.

We are also planning on extending our experiments and integrating a classifier for dialogue moves. We do have a preliminary version, which reaches an accuracy of approximately $70\%$ by looking only at previous move information, with disregard of the words in the sentences. While this is a module with state-of-the-art accuracy for dialogue moves, it is likely that the error rate is still too high to have a positive influence on POS tagging.

## Acknowledgment

This work is based on research supported by the US Office of Naval Research (ONR) Grant #N00014-10-1-0140.

# References

[1] Thorsten Brants. Tnt - a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing (ANLP-2000)*, Seattle, WA, 2000.

[2] Jean Carletta, Stephen Isard, Amy Isard, Gwyneth Doherty-Sneddon, Jacqueline Kowtko, and Anne Anderson. The reliability of a dialogue structure coding scheme. *Computational Linguistics*, 23(1):13–31, 1997.

[3] Stanley Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University, 1998.

[4] Hal Daumé III. Notes on CG and LM-BFGS optimization of logistic regression. Paper available at `http://pub.hal3.name#daume04cg-bfgs`, implementation available at `http://hal3.name/megam/`, August 2004.

[5] Pascal Denis and Benoît Sagot. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort. In *Proceedings of the Pacific Asia Conference on Language, Information and Computation (PACLIC 23)*, Hong Kong, China, 2009.

[6] Kathleen Eberhard, Hannele Nicholson, Sandra Kübler, Susan Gunderson, and Matthias Scheutz. The Indiana "Cooperative Remote Search Task" (CReST) Corpus. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, Valetta, Malta, 2010.

[7] Jesús Giménez and Lluís Màrquez. SVMTool: A general POS tagger generator based on Support Vector Machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal, 2004.

[8] Richard Johansson and Pierre Nugues. Extended constituent-to-dependency conversion for English. In *Proceedings of NODALIDA 2007*, Tartu, Estonia, 2007.

[9] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[10] Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP)*, Philadelphia, PA, 1996.

[11] Benoît Sagot, Lionel Clément, Éric De La Clergerie, and Pierre Boullier. The Lefff 2 syntactic lexicon for French: Architecture, acquisition, use. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, pages 1348–1351, Genoa, Italy, 2006.

[12] Beatrice Santorini. Part-of-speech tagging guidelines for the Penn Treebank Project. Department of Computer and Information Science, University of Pennsylvania, 3rd Revision, 2nd Printing, 1990.

[13] Kristina Toutanova and Christopher D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC)*, Hong Kong, 2000.

# Dependency Parsing using Prosody Markers from a Parallel Text

John Lee

Department of Chinese, Translation and Linguistics
City University of Hong Kong
E-mail: jsylee@cityu.edu.hk

**Abstract**

This paper describes a method to automatically generate dependency trees for ancient Greek sentences by exploiting prosodic annotation in a Hebrew parallel text. The head selection accuracy of the resulting trees, at close to 80%, is significantly higher than what standard statistical parsers might be expected to produce, for a resource-poor language such as ancient Greek. Our evaluation suggests that prosodic markers can be reliable indicators of syntactic structures.

## 1 Introduction

Increasingly, researchers in digital humanities are exploiting statistical techniques in the study of ancient languages, including decipherment [1, 2], morphology [3], and syntax [4, 5]. Data-driven syntactic analysis requires large treebanks, which are labour intensive and time consuming to create, especially so when the language in question no longer has any native speakers.

Ancient Greek is an important vehicle of human civilization, but relatively little syntactic annotation has been performed on its literature. Currently, the largest treebanks, both manually crafted, are the 200K-word Perseus Greek Dependency Treebank [6] and the 100K-word *Pragmatic Resources of Old Indo-European Languages* (PROIEL) [7]. An enormous amount of historically significant texts await analysis — the *Thesaurus Linguae Graecae* alone has more than 105 million words in its electronic collection. Although statistical parsers can be trained on these existing treebanks, their performance is unlikely to be adequate. Parsing accuracy has reached the nineties for English, but it is significantly lower for resource-poor languages [8], and lower still for classical Latin, a language with comparable characteristics and digital resources as ancient Greek: the state-of-the-art accuracy is about 54% [4][1].

[1] Accuracy for medieval Latin, however, is higher at about 80% [5].

This paper describes a method to automatically derive dependency trees in ancient Greek, by exploiting prosodic markers from a word-aligned parallel text in ancient Hebrew. The contribution of this paper is two-fold. First, the resulting treebank of the *Septuagint*, the Greek text with which we are concerned, contains 0.6 million words, doubling the size of the existing treebanks. Second, our evaluation shows that prosodic information can help produce parse trees of significantly higher accuracy than would be expected of those produced by a statistical parser, if trained on currently available resources.

# 2 Research Background

## 2.1 Previous Work

There has been much research on transferring the syntactic analysis of a resource-rich language, $L_1$, to a resource-poor language, $L_2$, given sufficient parallel text for the two languages in question. A popular approach is *syntactic projection* [9, 10], founded on the Direct Correspondence Assumption for syntactic structures. $L_1$ parse trees and $L_1$-$L_2$ word alignments are first acquired, either manually or automatically; dependencies between words in $L_1$ are then projected onto their $L_2$ counterparts, possibly followed by some local transformations as required by the linguistic peculiarities of $L_2$. In [10], using gold-standard word alignments and $L_1$ dependency trees, syntactic projection yielded unlabelled attachment F-scores of 70.3% for English-to-Spanish, and 67.3% for English-to-Chinese, a more divergent pair of languages.

In *bilingual parsing* [11, 12], a unified model performs joint inference for the best $L_1$ and $L_2$ parse trees, as well as the word alignments. This approach works well when there is noise in the $L_1$ parse trees, since it is able to find the combination of parse trees and alignments that are collectively more likely. It has been shown to improve Korean parsing when coupled with English [12].

With ancient Hebrew as $L_1$ and ancient Greek $L_2$, our work is analogous to syntactic projection, but with one crucial difference — we do not, in contrast to [10], have the luxury of a high-performing $L_1$ parser. Instead of $L_1$ dependency trees, our prior information will be prosodic annotation known as cantillation marks, for which we now provide some background.

## 2.2 Cantillation Marks

In order to facilitate public chanting of the Hebrew Bible, a group of scholars called the Masoretes added special symbols, called *cantillation marks*, to the text between the 7th and 10th century CE. These marks, written above or beneath a word, may be considered to be very fine-grained punctuation marks. They fall into one of two categories. Simply put, a word bearing a *disjunctive mark* suggests that a prosodic boundary separates it from the following word; a word bearing a *conjunctive mark*, in contrast, indicates that there is no such prosodic boundary.

| Category | Names (listed from strongest to weakest) |
|---|---|
| Disjunctive | passuq [·], atnah [;], segolta, little zaqef, great zaqef, tifha [,], revia, zarqa, pashta, yetiv, tevir, geresh, pazer, great pazer, great telisha |
| Conjunctive | maqqef [≡], munah [=], mehuppakh, merekha-khefula [-], darga, azla, little telisha, galgal |

Table 1: Cantillation marks are either *disjunctive* or *conjunctive*. They are listed above in descending order of strength [13]. The symbols in square brackets are shorthands used in the rest of the paper, and do not represent their actual shapes.

| Hebrew | *wy'mr* [-] | *'lhym* [,] | *yhy* [=] | *'wr* [;] | *wyhy* [≡] | *'wr* [·] |
|---|---|---|---|---|---|---|
| English | and said | the God | let be | light | and became | light |

Table 2: The original Hebrew words from Genesis 1:3, *And God said, "Let there be light", and there was light* (translation from Jewish Publication Society). Cantillation marks are shown in square brackets. See §2.2 for a discussion on how the marks help disambiguate this sentence.

These boundaries, however, should be understood in a relative sense, since these marks are organized in a complex hierarchy according to their levels of "strength"[2], as listed in Table 1 [13].

Cantillation marks can help interpret a sentence. Consider, for example, the Hebrew sentence in Table 2. Based on the words alone, it may be read as:

> *And God said, "Let there be light", and there was light.*

but also possibly as:

> *And God said, "Let there be light and there was light."*

However, the cantillation marks strongly suggest the first interpretation. The disjunctive marker *atnah* at the first occurrence of the word "light" is stronger than the *tifha* at the word "God"; hence the pause following the former should be more substantial than the latter.

Correspondence has been demonstrated between clause structures and tone units in English [14]. Cantillation marks have also been hypothesized to correlate with units of meaning and syntactic phrases in Hebrew [15], although no formal evaluation has been reported. In this paper, we use these marks to infer syntactic dependencies in an ancient Greek parallel text, namely the *Septuagint*, and evaluate the accuracy of the resulting dependency trees.

[2] Among the conjunctive marks, the *maqqef* is the strongest. The differences among the rest are of a musical nature only and are ignored for our purpose.

| Hebrew | *wy'mr* [-] | *'lhym* [,] | *yhy* [=] | *'wr* [;] | *wyhy* [≡] | *'wr* [·] |
|---|---|---|---|---|---|---|
| Greek | *kai eipen* | *ho theos* | *genēthētō* | *phōs* | *kai egeneto* | *phōs* |

Table 3: Hebrew-to-Greek word alignments associate Hebrew words to zero, one or more Greek words; these will be referred to as a "Greek chunk".

| POS combination | Frequency |
|---|---|
| Noun | 18% |
| Verb | 13% |
| Article + Noun | 12% |
| Conjunction + Verb | 8% |
| Pronoun | 6% |

Table 4: The most common parts-of-speech combinations for the Greek chunks extracted from the word alignments (see Table 3). For example, the chunk "*ho theos*" has the combination "Article + Noun".

# 3 Approach

In addition to the Hebrew text annotated with prosodic marks, we have at our dispoal Hebrew-to-Greek word alignments, some samples of which are shown in Table 3. In general the alignments are many-to-many, but most of the time one Hebrew word is associated with zero, one or two Greek words, which will be referred to as a "Greek chunk" in the rest of the paper. These chunks have an average length of 1.6 words, and will serve as the starting blocks of the derivation process of the dependency tree. The most frequent part-of-speech (POS) combinations for the chunks are listed in Table 4.

Following [7] and [18], we adopt the dependency tree [16] as the target syntactic representation. Our approach consists of two main phases. After some preliminary steps (§3.1), the first phase (§3.2) constructs dependency subtrees for each Greek chunk; the second phase (§3.3) merges these subtrees, two at a time, in an order determined by the Hebrew cantillation marks. The division of labor between these two phases is similar to that between the chunker and attacher in [19].

## 3.1 Preliminary Steps

Before parsing can begin, two preliminary steps must be taken to deal with incomplete word alignment and different word orders.

### 3.1.1 Insertion

In about 40% of the sentences, one or more Greek words are not aligned with any Hebrew ones. In some cases, they reflect genuine differences in the content, due to textual variations between the Hebrew *Vorlage* and our particular *Septuagint* version. But far more often, non-alignment is caused by differences in syntax. For example, the Hebrew noun construct chain "*yšby* ⟨place⟩" (literally, "dwellers-of ⟨place⟩") is often rendered in Greek as the participial form of "dwell", followed by the preposition "*en*" ("in") and ⟨place⟩, such as "*katoikountas en tais polesin*" ("those dwelling in the cities", Genesis 19:25).

In other instances, stylistic considerations are responsible for the non-alignment. For example, the Greek verb-to-be "*ēn*" is used in the phrase "*lithos de ēn megas*" ("*the stone ... was large*", Genesis 29:2), where the original Hebrew has none.

These non-aligned Greek words may form their own chunk. Alternatively, they may be amalgamated with the chunk to its left; a common example is the joining of a postpositive particle, such as "*oun*" ("therefore"), to the preceding verb. Lastly, they may join the chunk on their right, as would be appropriate for the preposition "*en*" heading a prepositional phrase, such as in the example above. Among these three options, we choose the one that would yield a chunk whose POS combination has the highest frequency count, based on statistics computed from the rest of the corpus.

### 3.1.2 Re-ordering

The *Septuagint* is a highly literalistic translation; the relatively free word order in the Greek language allowed translators to largely conserve the word order in the original. Indeed, ignoring insertions and deletions, the Greek word order exactly matches the Hebrew word order in 94% of the verses. This high percentage facilitates the preservation of prosodic boundaries from the Hebrew to the Greek.

Two systematic exceptions involve particles and numbers. One is the use of postpositive particles such as "*gar*" and "*de*", which must be placed after the first word, to render the Hebrew sentence-initial "*ky*". For noun phrases involving a number and the word "year" or "day", the number usually comes first in Hebrew but comes last in Greek. Some other differences in word order are caused by the relative positions of verbs and their direct or indirect objects.

## 3.2 Parsing Greek Chunks

After addressing issues with word alignments and word order, we can now derive dependency trees for the Greek chunks. A dozen rules, each targeting chunks of a specific POS combination, were written to assign dependency relationships within the chunk, using the guidelines for [7]. Table 5 shows a rule for chunks of the type "Article + Noun" being applied on the chunk "*ho theos*"; the noun "*theos*" is annotated as the head of the article "*ho*", with the relation AUX.

These rules make use of not only the POS combination but also morphological information. Consider the chunks "*eneteilamēn soi*" ("I commanded you") and

| Greek<br>chunk | *kai eipen* [-]<br>"and said" | *ho theos* [,]<br>"the God" | *kai egeneto* [≡]<br>"and became" |
|---|---|---|---|
| Subtree | [-]<br>*kai*<br>\|<br>*eipen*<br>PRED | [,]<br>*theos*<br>\|<br>*ho*<br>AUX | [≡]<br>*kai*<br>\|<br>*egeneto*<br>PRED |

Table 5: Derivation of dependency subtrees (see §3.2) for the Greek chunks, taken from the sentence in Table 3. (The chunks of length 1 are omitted.) Note that the Hebrew cantillation marks have been projected onto the chunks.

"*ginōskō egō*" ("I know"); both consist of a verb followed by a personal pronoun. In the first, the pronoun "you" is dative and hence is an indirect object of the verb; in the second, in contrast, the pronoun "I" is nominative and is the subject of the verb.

## 3.3 Merging Subtrees

For each sentence, the procedure in §3.2 yields a sequence of subtrees, such as those in Table 5; they must then be merged into a single dependency tree. The merging process requires two kinds of decisions: the merge order, which will be determined by the relative strengths of the cantillation marks; and the attachment site, which will be informed by manually derived rules.

### 3.3.1 Merge Order

Using Hebrew-to-Greek word alignments, cantillation marks are projected from each Hebrew word to its corresponding Greek chunk and subtree (Table 5). Only the mark on the last word of a chunk is retained; the rest are ignored. Following the analogous treatment of the Hebrew in [17], the Greek subtrees are then merged two at a time. First, in descending order of strength, those with conjunctive marks are merged with their right neighbors (step 1 in Table 6); then, in ascending order of strength, those with disjunctive marks are merged with their right neighbors (steps 2 and 3 in Table 6).

To ensure each Greek chunk has one cantillation mark, two issues need to be resolved. First, when a new chunk is inserted via the insertion step (§3.1.1), its cantillation mark is predicted using an *n*-gram model. We trained a trigram model on the existing chunks, treating a chunk's POS combination as the "word" and its cantillation mark as the "tag". Also, if a Hebrew word has a left neighbor which is unaligned but has a stronger disjunctive mark, it will project this stronger mark instead of its own, so as to preserve the prosodic boundary.

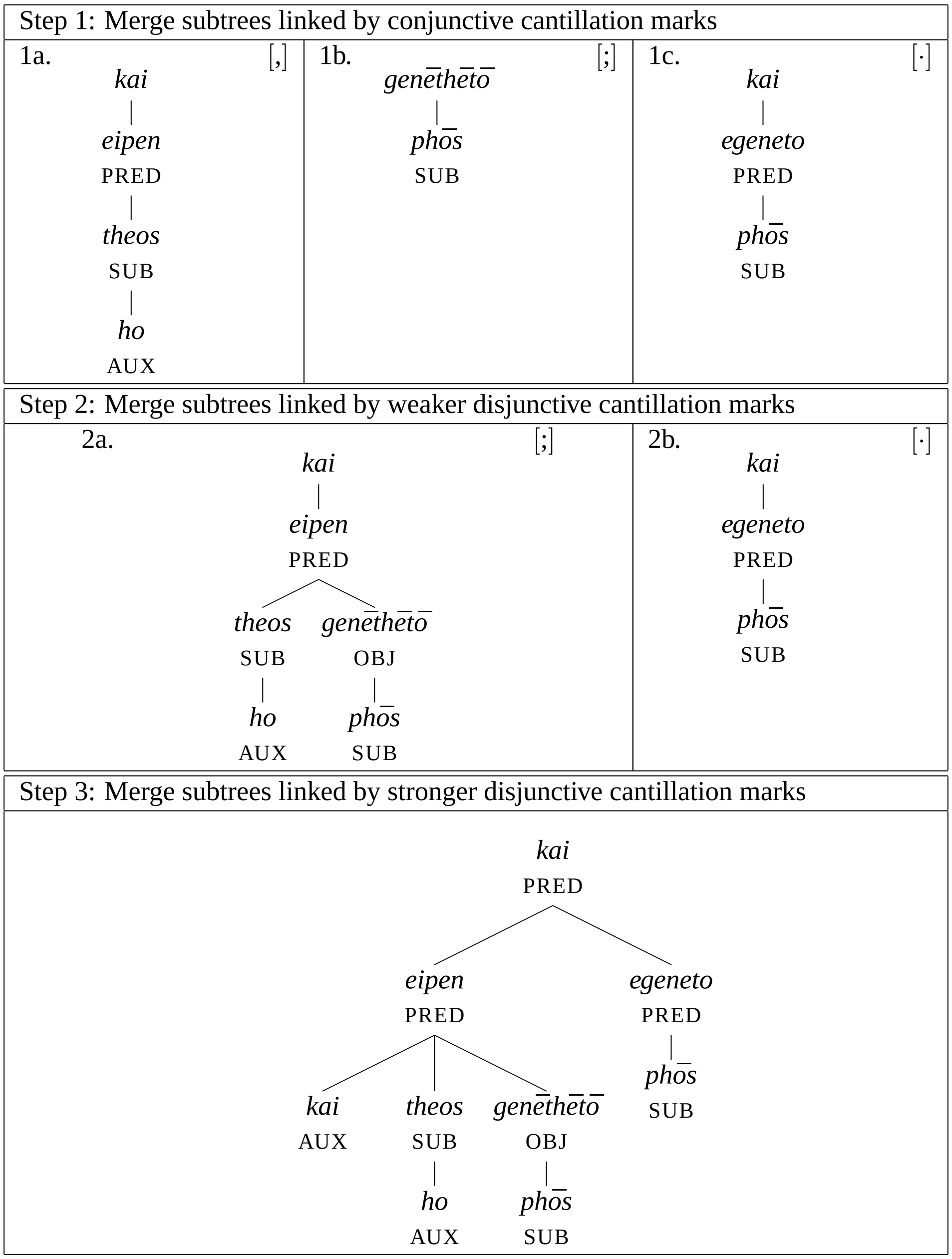


Table 6: The subtree merging process (§3.3) for Genesis 1:3. Step 1 shows the result of merging three pairs of subtrees (Tables 3 and 5) that are connected by conjunctive marks (namely the *merekha*, *munah* and *maqqef*). Step 2 faces two options: merge 1b (“Let there be light”) and 1c (“And there was light”) first, or merge 1a (“And God said”) and 1b first. The first option would have yielded a tree with the interpretation *And God said, “Let there be light and there was light”*. See a discussion in §2.2 on how the stronger disjunctive mark between 1b and 1c rules out this option. Finally, 2a and 2b ar l as coordinated predicates, resulting in the final tree in step 3.

<table>
<tr><td>*eis* [≡]<br>“to”</td><td>*gēn* [=]<br>“land”</td><td>*oikoumenēn*<br>“settled”</td></tr>
<tr><td colspan="2">[=]<br>*eis*<br>|<br>*gēn*<br>OBL</td><td>?<br>|<br>*oikoumenēn*</td></tr>
<tr><td colspan="3">*eis*<br>OBL<br>|<br>*gēn*<br>OBL<br>|<br>*oikoumenēn*<br>ATR</td></tr>
</table>

<table>
<tr><td>*meta* [≡]<br>“with”</td><td>*Laban* [=]<br>“Laban”</td><td>*parōikēsa*<br>“I stayed”</td></tr>
<tr><td colspan="2">[=]<br>*meta*<br>|<br>*Laban*<br>OBL</td><td>?<br>|<br>*parōikēsa*</td></tr>
<tr><td colspan="3">*parōikēsa*<br>PRED<br>|<br>*meta*<br>ADV<br>|<br>*Laban*<br>OBL</td></tr>
</table>

Table 7: Examples of subtree merging involving function words, as discussed in §3.3.2. Both are preposition-noun-verb trigrams with identical cantillation marks, but their merged trees are completely different. The verb in the top example (Exodus 16:35) is a participle which forms part of a long noun phrase, whereas the verb in the bottom (Genesis 32:5) is finite and becomes the root of the new tree. Morphological analysis of the verbs is indispensable for correct parsing.

### 3.3.2 Root Attachment Site

Having specified the order, we now turn our attention to how the dependency subtrees are merged. The most straightforward manner is to assign the root word of one subtree as the head of the root of the other. For example, in Genesis 32:5 in Table 7, the verb "*parōikēsa*" is assigned as head of the preposition "*meta*". A verb-object pair would be treated likewise. This decision is made according to the POS of the root words, expressed through a dozen of deterministic rules for each POS pair. Table 6 continues with our running example, completing the entire merging process.

**Relative clauses** When the prosodic structure differs from the syntactic structure, the appropriate attachment site may not be the root. One such case occurs with a relative clause, whose root is dependent on its antecedent noun; this noun is not necessarily the root of the other subtree. For example, in Genesis 3:3, there are two chunks "*apo de karpou tou xulou*" ("from the fruit of the tree") and "*ho estin en mesō tou paradeisou*" ("that is in the middle of the garden"). The relative pronoun "*ho*" signals that the root "*estin*" ("is") must be attached to its antecedant noun. The algorithm searches within the first subtree in post-order, to find a noun — in this case, "*xulou*" ("tree") — that agrees with the relative pronoun with respect to gender and number. Hence, "*xulou*", rather than the head "*apo*" ("from"), becomes the head of "*estin*".

**Function words and noun phrases** A systematic disagreement between the cantillation marks and phrase boundaries occurs when a long noun phrase depends on a function word, such as a conjunction or a preposition. Consider the phrase "*eis* [≡] *gēn* [=] *oikoumenēn*" ("to a land that was settled", illustrated in Table 7). The strongest conjunctive mark, the *maqqef*, binds the preposition "*eis*" ("to") to "*gēn*" ("a land"), the first word of the noun phrase; it thus tears apart the two-word NP "a land [=] that was settled", which is held together by a weaker conjunctive mark, the *munah*. This makes sense prosodically, since a pause is needed in the middle of a long NP, as suggested in the discussion of a similar phenomenon in Hebrew in [15]. The procedure described above would have produced an incorrect tree; instead, *oikoumenēn* should be dependent on the non-root *gēn*.

Morphological analysis is necessary to decide whether this kind of adjustment is warranted. Consider another phrase with a preposition-noun-verb sequence, "*meta* [≡] *Laban* [=] *parōikēsa*" ("with Laban I have been staying", illustrated in Table 7). Its surface structure and cantillation marks are indistinguishable from the last example. However, the finiteness of the verb suggests that the preposition should be dependent on it.

# 4 Evaluation

## 4.1 Data

The book of Genesis is used as the development set to design the rules for parsing Greek chunks (§3.2) and merging subtrees (§3.3). Three poetic books in the *Septuagint*, namely Job, Psalms and Proverbs, use a system of cantillation marks that differ from the one presented in §2.2. They were therefore excluded from the evaluation. The cantillation marks are extracted from the corpus described in [15]. The Hebrew-to-Greek word alignments and the morphologically analyzed corpus of the *Septuagint* are compiled by the Center for Computer Analysis of Texts at the University of Pennsylvania.

There is no existing treebank for the *Septuagint*. Fortunately, many of its verses appear also in the Greek New Testament, much of which has been analyzed in the PROIEL dependency treebank [7]. Some quotations diverge slightly from the original; to automate the creation of the gold-standard trees, we adopted the simple criterion of including all fragments of at least five consecutive words that are quoted *verbatim*. The gold-standard trees[3] for these fragments were extracted from PROIEL. After these filtering steps, there were altogether 995 words for evaluation.

## 4.2 Result

The unlabeled attachment score is 79.4%. A comparison with [10], a related work in syntactic projection, is difficult due to different language pairs and text genre; nonetheless, the higher score achieved here provides some evidence that prosodic boundaries are reasonable predictors of syntactic boundaries. A chief problem is the analysis of coordinated phrases, especially those embedded in long sentences. Another source of head selection error is the attachment of relative pronouns, as described in §3.3.2, as well as participles, when it can be modifying one of multiple nouns or verbs.

Among words with correctly selected heads, 88.5% are assigned the correct dependency label, yielding an overall labeled attachment score of 70.6%. This level of accuracy is significantly higher than what statistical parsers might be expected to achieve, if the corresponding score of 54% reported in [4] for classical Latin, a language with similarly limited resources, has any value as a reference point. Among the label errors, the most frequent mistake, constituting more than a fifth of the total, is the confusion between adjunct and argument in prepositional phrases (labeled as ADV and OBL, respectively). This difficulty echoes the findings in [20]; indeed the adjunct-argument distinction remains challenging even for resource-rich languages such as English [21].

[3] If the head of a word is located outside the fragment, the word is excluded, since its head may not be the same in the *Septuagint*.

# 5 Conclusion and Future Work

We have described a method to automatically create ancient Greek dependency trees by leveraging prosodic annotation in a parallel text in Hebrew. The resulting treebank for the *Septuagint* is a substantial addition to the relative dearth of syntactic data currently available for ancient Greek. It may be expected to help boost the performance of a statistical parser.

This study also provides evidence that cantillation marks are good indicators of syntactic boundaries. Similar techniques can generate treebanks for other languages into which the Hebrew Bible has been translated.

## Acknowledgments

We thank the J. Alan Groves Center for Advanced Biblical Research for making available the Hebrew Bible corpus.

## References

[1] Lin, Shou-de and Knight, Kevin (2006) Discovering the Linear Writing Order of a Two-Dimensional Ancient Hieroglyphic Script. In *Artificial Intelligence* **170**(4/5):409—421.

[2] Snyder, Benjamin, Barzilay, Regina and Knight, Kevin (2010) A Statistical Model for Lost Language Decipherment. In *Proc. ACL*.

[3] Lee, John (2008) A Nearest-Neighbor Approach to the Automatic Analysis of Ancient Greek Morphology. In *Proc. Conference on Computational Natural Language Learning (CoNLL)*.

[4] Bamman, David and Crane, Gregory (2008) Building a Dynamic Lexicon from a Digital Library. In *Proc. 8th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2008)*.

[5] Passarotti, Marco and Dell'Orletta, Felice (2010) Improvements in Parsing the Index Thomisticus Treebank. In *Proc. LREC*.

[6] Bamman, David, Mambrini, Francesco and Crane, Gregory (2009) An Ownership Model of Annotation: The Ancient Greek Dependency Treebank. In *Proc. TLT*.

[7] Haug, Dag and Jøhndal, Marius (2008) Creating a Parallel Treebank of the Old Indo-European Bible Translations. In *Proc. LREC Workshop on Language Technology for Cultural Heritage Data*.

[8] Nivre, Joakim, Hall, Johan, Kübler, Sandra, McDonald, Ryan, Nilsson, Jens, Riedel, Sebastian and Yuret, Deniz (2007) The CoNLL 2007 Shared Task on Dependency Parsing. In *Proc. EMNLP-CoNLL*.

[9] Yarowsky, David and Ngai, Grace (2001) Inducing Multilingual POS Taggers and NP Bracketers via Robust Projection across Aligned Corpora. In *Proc. NAACL*.

[10] Hwa, Rebecca, Resnik, Philip, Weinberg, Amy, Cabezas, Clara and Kolak, Okan (2005) Bootstrapping Parsers via Syntactic Projection across Parallel Texts. In *Natural Language Engineering* **11**(3):311–325.

[11] Melamed, I. Dan, Satta, Giorgio and Wellington, Benjamin (2004) Generalized Multitext Grammars. In *Proc. ACL*.

[12] Smith, D. A. and Smith, N. A. (2004) Bilingual Parsing with Factored Estimation: Using English to Parse Korean. In *Proc. EMNLP*.

[13] Robinson, David and Levy, Elisabeth (2002) *The Masoretes and the Punctuation of Biblical Hebrew*. British and Foreign Bible Society.

[14] Fang, Alex C., House, Jill and Huckvale, Mark (1998) Investigating the Syntactic Characteristics of English Tone Units. In *Proc. International Conference on Spoken Language Processing (ICSLP)*.

[15] Wu, Andi and Lowery, Kirk (2006) From Prosodic Trees to Syntactic Trees. In *Proc. ACL*.

[16] Mel'čuk, Igor (1988) *Dependency Syntax: Theory and Practice*. State University of New York Press.

[17] Machine Assisted Translation Team (2002) The Masoretes and the Punctuation of Biblical Hebrew. British and Foreign Bible Society. *http://lc.bfbs.org.uk*

[18] Crane, Gregory, Chavez, Robert F., Rydberg-Cox, Jeffrey A., Mahoney, Anne, Milbank, Thomas L. and Smith, David A. (2001) Drudgery and Deep Thought: Designing Digital Libraries for the Humanities. In *Communications of the ACM* **44**(5):175–182.

[19] Abney, Steven. Parsing by Chunks (1991) In Robert Berwick, Steven Abney and Carol Tenny (eds.), *Principle-Based Parsing*, Kluwer Academic Publishers.

[20] Lee, John and Haug, Dag (2010) Porting an Ancient Greek and Latin Treebank. In *Proc. LREC*.

[21] Merlo, Paola and Esteve Ferrer, Eva (2006) The Notion of Argument in PP Attachment. In *Computational Linguistics* **32**(3):341—378.

# Using a Treebank for Finding Opposites

Anna Lobanova

University of Groningen
Artificial Intelligence Dept.
E-mail: a.lobanova@ai.rug.nl

Gosse Bouma

University of Groningen
Information Science Dept.
E-mail: g.bouma@rug.nl

Erik Tjong Kim Sang

University of Groningen
Information Science Dept.
E-mail: erikt@xs4all.nl

**Abstract**

We present an automatic method for extraction of pairs of opposites (e.g. *hot-cold*, *top-bottom*, *buy-sell*) by means of dependency patterns that are learned from a 450 million word treebank containing texts from Dutch newspapers. Using small sets of seed pairs, we identify the best patterns for finding new pairs of opposites.

Treebanks are useful for generating dependency patterns expressing relations between words that occur far away from each other, something which is more difficult with textual patterns. Furthermore, textual patterns tend to find opposites expressed by the most frequent part-of-speech (PoS) category, viz. nouns ([17]). We examine whether dependency patterns can also be used for finding pairs of opposites of less frequent PoS classes: adjectives and verbs.

We successfully employ dependency patterns for extracting opposites but find that the best acquired patterns are too general and extract a lot of noise. We conclude that while syntactic information helps to identify opposites for less frequently co-occurring PoS categories, more data, e.g. available from the Web, should be used to improve the results.

## 1 Introduction

Recent years have produced increased efforts in research on automatic extraction of semantic relations like hyponymy, meronymy and synonymy. Yet, other relations, in particular, antonymy, have received little attention. In this paper, we present an automatic method for finding opposites by means of dependency patterns that are automatically acquired from a treebank of Dutch. We define opposites as a general class of antonyms that includes word pairs like *dead-alive*, *tall-short*, as well as incompatibles like *summer-winter*, *day-night*, *ask-answer*, etc. Our goal is to examine whether syntactic information is beneficial for identifying opposite words of different part-of-speech (PoS) categories.

Automatic extraction of opposites is useful for many NLP applications including sentiment analysis (e.g. by establishing the strength of antonymy of identified pairs [20]), automatic identification of contrastive relationships (see [19], [26]), and augmentation and verification of existing lexical resources, especially for languages other than English. A list of automatically found opposites can also be applied as a filter to improve the performance of automatic techniques for synonym, hyponym and meronym extraction ([18]), where antonym noise is a notorious problem ([16]).

Similarly to other lexically related words, opposites tend to co-occur with each other sententially and often they co-occur in so-called textual patterns like "*difference between X and Y*" or "*X as well as Y*" ([13]). However, opposites also occur outside of frequently used short textual patterns. For instance, in a Dutch example below (1), opposites *houden van - haten* ("to love" - "to hate") occur in parallel constructions outside of the scope of meaningful reoccurring textual patterns:

(1) Men houdt van Felicia, de Oprah Winfrey wannabe van Zuid-Afrika, of men haat haar.
*People love Felicia, the Oprah Winfrey wannabe from South-Africa, or people hate her.* (NRC, Dec 20, 2000)

Such cases are not rare. In fact, [13], who analysed 3000 newspaper sentences with well-established opposites, reports that in 38% of the sentences, opposites occurred outside of reasonable textual patterns. Because of this, many good instances can be missed, which in turn has a negative effect the recall of the pair extraction process. Dependency patterns can provide a plausible solution for this problem as they are acquired from treebank data, which contain syntactic relations between elements of a sentence and allow abstracting away from the surface structure. The dependency pattern $Verb1{:}conj \leftarrow of \rightarrow conj{:}Verb2$, for example, links the two verbs in (1), representing the shortest path between them in the dependency tree.

While an increasing number of available treebanks allows to use syntactic information in relation extraction, using dependency patterns for finding opposites has not yet been done. Overall, there is no consensus as to whether such methods outperform techniques based on textual patterns. For example, [27] compared two automatic methods for hyponym-hypernym extraction for Dutch. In one method they used dependency patterns, while the other method relied on textual patterns which contained PoS category information. They found that both methods performed equally well. Results in an earlier study of [25], however, showed that dependency patterns outperformed textual patterns with PoS information for hypernym-hyponym extraction in English.

An important difference between antonymy as opposed to meronymy and hyponymy is that only antonymy relation can occur between words of more than one PoS category, including nouns, adjectives, and verbs. Exploring whether dependency patterns can find opposites that belong to different PoS categories is useful for understanding the benefits of syntactic information for relation extraction, as well as the extent to which dependency patterns differ from textual patterns. Since verb candidates are more likely to co-occur in a sentence further away from each other than nouns and adjectives, a method based on dependency patterns might be more productive for extraction of antonymous verbs rather than nouns and adjectives.

Alternatively, the antonym detection process might not be affected by the PoS categories of the candidates. Previous pattern-based work on extraction of opposites used textual patterns identified by means of adjective-adjective seeds ([17]). Interestingly, the majority of pairs they found were noun-noun pairs. Thus, by using dependency patterns with seeds that belong to several PoS categories, we can study whether syntactic information is more useful for pairs and relations that belong to certain PoS categories.

**Outline** The remainder of this article is organized as follows. In Section 2 we give an overview of previous pattern-based studies on relation extraction as well as existing work on antonym extraction. Our method is discussed in Section 3. The results are presented in Section 4. Our main finding is that dependency patterns are rather general and find not only opposites but also other frequently co-occurring pairs. The system performed best with adjective-adjective seeds, followed by nouns and verbs. The results are discussed and summarized in Section 5 where we also discuss directions of future work.

## 2 Previous work

A pattern-based method for relation extraction was originally proposed by [10] who suggested that patterns, in which words co-occur, signal lexical semantic relationships between them and, therefore, can be used to identify those relations. Using six manually identified textual patterns like *such NP as NP*, she found phrases, e.g. '*such authors as Shakespeare*' and used them to successfully extract facts like e.g. *Shakespeare* is a kind of *author*. In the 8.6 million word corpus of encyclopedic texts, Hearst found 153 candidate hyponym pairs, of which only 61 were listed in a hyponym relationship in WordNet [8], suggesting that the method could easily add useful relations missing in WordNet. As future work, Hearst suggested that a similar approach can be used to identify other lexical relationships.

Testing Hearst's suggestion, [1] used patterns to find meronyms from a newspaper corpus of 100 million words. Starting with a set of selected meronym pairs as seeds, they extracted all sentences that contained them and manually identified plausible patterns. The best two patterns were then enlisted to extract new pairs. They report an accuracy of 55% for the top 50 meronyms derived for six seeds based on the majority vote of the evaluation of the pairs by five human annotators.

Neither [10], nor [1] identified patterns automatically. Using a minimally supervised bootstrapping algorithm *Espresso*, [22] identified generic patterns automatically and used them to extract a range of relations including meronymy and hyponymy. Also beginning with seed pairs, they extracted all sentences these pairs co-occurred in in a 6.3 million word newspaper corpus and used those sentences to generalize patterns. All patterns were automatically evaluated based on pointwise mutual information ([4]). Top-10 best patterns were used to find new pairs. Extracted pairs were also evaluated using an association score between a given pair and a highly reliable pattern. Since by nature generic patterns are frequent and contain a lot of noise, pattern recall was increased by using the Web to retrieve more instances. Their method had high precision and also high recall. The obtained precision scores for the sample of 50 extracted instances of hyponyms and 50 extracted instances of meronyms with their top algorithm were between 73% and 85% (based on evaluation by two human annotators). Our algorithm is based on Espresso, but instead of textual patterns, we apply dependency patterns. As Pantel and Pennacchiotti mention themselves (2006: 3), the way patterns are defined and extracted does not affect the algorithm.

[25] were the first to use syntactic information to automatically derive dependency patterns to find hyponym-hypernym pairs in English. In their approach, they compared performance of a number of classifiers that as their features used noun-noun pairs extracted from a fully parsed six million word corpus of newspaper texts and different types of patterns, including dependency patterns and textual patterns. Their best logistic regression classifier was based on dependency patterns. It outperformed a classifier based on manually crafted patterns from [10]. According to the authors, their results indicate that dependency patterns are not only useful for identification of hyponymy relation but that they are better at hypernym-hyponym extraction than methods based on textual patterns.

The extent to which syntactic information is beneficial, is still disputed. In particular, [27] replicated Snow et al.'s approach on Dutch and compared it with a method based on textual patterns with PoS information. No significant differences were found between these methods. The largest effect was found for Wikipedia texts, where dependency patterns found 23% more related pairs than textual patterns. The authors argue that this affect can be overcome by adding 43% extra data.

Studies described above dealt with noun-noun pairs only. In this study we aim at finding a relation expressed not only by noun-noun but also adjective-adjective and verb-verb pairs. Using Espresso-based algorithm for finding meronyms, [12] conducted a detailed evaluation of the role seed types can play in extracting the target relation. They found that the best results were achieved using seeds that belonged to the same PoS class rather than mixed types. By using seeds for each PoS category, we examine how grammatical category of seeds affects generation of patterns and, consequently, the range of opposites found. It might be that a pattern-based method performs better with seeds of a certain PoS category, e.g. the most frequent one expressed by nouns, something that is addressed in our study.

Existing work on automatic extraction of opposites is based on surface patterns that do not capture any syntactic information. Starting with a small set of adjective-adjective seeds, [17] extracted all sentences that contained any seed pair from a newspaper corpus of Dutch (72 million words). Textual patterns were automatically constructed, and top-50 most frequently occurring patterns that contained one of seed pairs at least twice were used to find new instances of antonyms. Patterns consisted of five or more tokens as shorter patterns extracted too much noise. Both patterns and found instances were automatically scored. The scoring of patterns was based on how often they contained seed pairs and their overall frequency. The scoring of pairs was based on the number of times a pairs occurred with each pattern and its score. The algorithm was repeated iteratively six times, using pairs with scoring above a set threshold as new seeds at each iteration. All found pairs with scoring above 0.6 were evaluated by five human annotators. The results showed that surface patterns can be used to find not only a small range of well-established opposites but a wider class of pairs known as incompatibles. Still, the precision scores were considerably lower than those found with automatic hyponym and meronym extraction. Based on the majority vote by five annotators, for the set of six seeds they report a precision of 28% for pairs with scoring $\geq$0.6 when separating opposites from incompatibles (54 pairs), and a precision of 67% when opposites and incompatibles were treated as one group (129 pairs). The authors suggest that one of the reasons for the lower scores is that although all seeds were adjectives, most of found pairs consisted of nouns. Antonymy as a relation is best understood for adjectives whereas with nouns there is a unclear boundary between incompatibles like *summer-winter* and correlates like *suspect-witness*. This made evaluation of the results more difficult. Importantly, all correlates they found indicated some kind of contrast (e.g. a found pair *suspect-witness* as opposed to correlates *table-chair*) suggesting that their results could be useful for automatic identification of contrast relations.

The study conducted by [17] is similar to the *Espresso* method ([12]), but the ranking of patterns and pairs is based on a different metric, making it difficult to compare results. In this study, we present an *Espresso*-like algorithm that is using the same metric as [12].

# 3 Current Study

## 3.1 Materials

**Corpus.** We used a 450 million word version of Twente Nieuws Corpus of Dutch (TwNC, [21]) that consisted of 26 million sentences. The corpus consists of newswire texts from five daily Dutch newspapers.[1] The corpus was syntactically parsed by Alpino, a parsing system for Dutch aimed at parsing unrestricted texts ([28]). The parsing accuracy of Alpino is over 90% (tested on a set of 2256 newspaper

[1] Namely, *Algemeen Dagblad*, *NRC Handelsblad*, *Parool*, *Trouw* and *Volkskrant*.

| Adjective-Adjective seeds | Noun-Noun seeds | Verb-Verb seeds |
|---|---|---|
| poor - rich | beginning - end | lose - win |
| open - closed | man - woman | give - take |
| large - small | day - night | buy - sell |
| fast - slow | question - answer | open - close |
| beautiful - ugly | advantage - disadvantage | find - lose |
| narrow - broad | peace - war | laugh - cry |
| dry - wet | top - bottom | end - begin |
| new - old | heaven - hell | increase - decrease |
| high - low | exit - entrance | save - spend |
| cold - hot | strength - weakness | confirm - deny |
| old - young | punishment - reward | succeed - fail |
| long - short | optimist - pessimist | ask - answer |
| happy - sad | husband - wife | attack - defend |
| active - passive | chaos - order | hate - love |
| right - wrong | predator - prey | fall - rise |
| dead - alive | employer - employee | exclude - include |
| heavy - light | fact - fiction | export - import |
| hard - soft | attack- defence | add - remove |

Table 1: List of (translated) seed pairs for each part-of-speech category.

sentences ([28]), which is comparable to the state-of-the-art parsers for English ([5], [3], [15]).

**Seeds.** Seed sets were manually compiled from available lists of well-established opposites studied in psycholinguistic experiments (e.g. word association tests [7]) and corpus-based experiments (e.g. in terms of *breadth of co-occurrence* in [14]) and discussed in theoretical classifications ([6]). A preliminary study showed that these seeds outperformed seed sets that consisted of morphologically-related pairs (e.g. *known* - ***un**known*) or top-50 most frequent antonyms presented in the Dutch lexical database CORNETTO ([11]). A complete list of adjective-adjective, noun-noun and verb-verb seeds used in this study is presented in Table 1.

## 3.2 The Algorithm

Our method is based on the well-known minimally-supervised bootstrapping algorithm, *Espresso* ([22]). First, using seed pairs as tuples, dependency patterns that contained both words of a pair, were extracted from the treebank. Patterns that were found once were discarded. Next, patterns were automatically scored. The reliability of a pattern $p$, $r_\pi(p)$, given a set of input pairs $I$ was calculated as its average strength of association across each input (seed) pair $i$ in $I$, weighted by the reliability of each input pair $i$, $r_\iota(i)$:

$$r_\pi(P) = \frac{\sum_{i \in I} \left( \frac{pmi(i,p)}{max_{pmi}} * r_\iota(i) \right)}{|I|}$$

where $pmi(i,p)$ is the pointwise mutual information score (Church and Hanks 1990) between a pattern and an input pair, and $\max_{pmi}$ is the maximum pointwise mutual information between all patterns and all pairs. The reliability of initializing seed pairs was set to 1. Next, the top-k most reliable patterns were used to find new candidate pairs. We set the number of initial set of top patterns to 10, adding one extra pattern at each iteration.[2] The reliability of found pairs, $r_\iota(i)$ was estimated as follows:

$$r_\iota(i) = \frac{\sum_{p \in P} \left( \frac{pmi(i,p)}{max_{pmi}} * r_\pi(p) \right)}{|P|}$$

where $P$ is the set of top-k found patterns.

The top-100 found pairs were used as new seeds in the next iteration. The process was repeated iteratively until at least 500 new pairs were acquired.

### 3.3 Evaluation

All found pairs were manually evaluated by three human annotators. They were asked to classify each pair as an opposite or a non-opposite. Opposites were described as words that belong to the same category but express the opposite of each other. We report a Fleiss's kappa score for inter-annotator's agreement (Randolph 2005). A score between 0.61 and 0.8 is considered to indicate a substantial agreement. In addition, we evaluated the results against CORNETTO, a newly available lexical resource for Dutch ([11]).[3] Finally, we calculated precision scores for each set of results, treating all pairs unanimously judged as opposites as *true positives*, pairs unanimously judged as non-opposites as *false positives* and discarding ambiguous pairs.

## 4 Results

### 4.1 Results for adjective-adjective pairs

Out of 519 pairs found with 18 adjective-adjective seeds, 34% (178 pairs) were judged as opposites by at least two annotators (82% of which received unanimous vote). They contained pairs like *automatisch - handmatig* ("automatic - manual"), *ziek - gezond* ("sick - healthy"), *leeg - vol* ("empty - full"). For 88% of those pairs (156) both words were listed in CORNETTO, but only 67 of them (43%) were linked

[2] Because we use a much bigger corpus than Pantel and Pennacchiotti [22], we do not retrieve additional instances of patterns from the web. We also do not use a discounting factor suggested in Pantel and Ravichandran (2004) and used in Pantel and Pennacchiotti [22] to control for the bias of *pmi* towards infrequent events. Instead we remove patterns and pairs that occur only once.

[3] This evaluation was done by means of a Python module PYCORNETTO developed by Erwin Marsi and available at http://code.google.com/p/pycornetto/.

| Nr of iteration | Adjective-Adjective pairs | Noun-Noun pairs | Verb-Verb pairs |
|---|---|---|---|
| 1 | 0.67 | 0.56 | 0.22 |
| 2 | 0.52 | 0.44 | 0.16 |
| 3 | 0.46 | 0.36 | 0.14 |
| 4 | 0.39 | 0.31 | 0.12 |
| 5 | 0.34 | 0.25 | 0.10 |

Table 2: Precision scores per iteration and PoS category (Adjective, Noun, Verb).

as opposites, indicating that for 57% of the valid pairs found by our method, the antonym relation was missing in the database. However, the majority of the candidate pairs, 66% (341 pairs), was unanimously judged as non-opposite. Among such pairs were e.g. *dood - zwaargewond* ("dead - heavily injured"), *politiek - zakelijk* ("political - bussinesslike"), *blij - tevreden* ("happy - contented") and others. Annotators achieved a Fleiss's kappa score of 0.73 indicating substantial agreement. Precision scores for each iteration, summarized in Table 2, show decreasing precision scores for later iterations. In particular, while precision score for the adjective seeds at the first iteration was 0.67, it decreased to 0.34 at the last iteration. One of the reasons for this can be that new pairs added at each following iteration make the results noisier. To investigate that we examined top-50 novel pairs extracted only at a given iteration. Among top-50 novel pairs found only at iteration one, 74% (37 pairs) were judged as opposites leading to a precision score of 0.86. At the last iteration only 26% of pairs (13) found only at that iteration were judged as opposites by the majority vote leading to a precision score of 0.26.

We also analysed the top patterns to see whether dependency patterns discovered by means of initial seeds were different from patterns discovered at later iterations with found seeds. The most frequent pattern at each iteration was *ANT1:conj ← or → conj:ANT2*, followed by patterns *ANT1:conj ← as well as → conj:ANT2* and *ANT1:conj ← neither nor → conj:ANT2*. Patterns found at first iteration were rather general and frequent, all ten of them were also found at each consequent iteration. Interestingly, our algorithm did not find an equivalent variant of one of the most frequent and productive textual patterns discovered with adjectival seeds by [17], namely *between X and Y*.

## 4.2 Results for noun-noun pairs

Out of 518 pairs found with 18 noun-noun seeds, 28% (143 pairs) were judged as opposites by at least two participants (72% of them received unanimous vote). Among pairs classified as opposites were pairs *kind - volwassene* ("child - grown up"), *tegenstander - vriend* ("adversary - friend"), *mislukking - succes* ("failure - success").

For 90% of pairs (128) that were judged as opposites, both words were listed in the CORNETTO database but only nine of them (7%) were linked as opposites. Thus, 93% of opposites are not captured by the lexical resource. Another 72%

(375 pairs) were judged by the majority vote as non-opposites. These pairs included many correlates, e.g. *politicus - sporter* ("politician - sportsmen"), *slip - top* ("underpants - top"), *fan - speler* ("fan - player"), as well as unrelated words like *rijkdom - vrede* ("wealth - peace") and *naam - talent* ("name - talent"). The annotators achieved a Fleiss's kappa score of 0.67 indicating sufficient agreement.

Again, as shown in Table 2, precision scores were higher at initial iterations (precision of 0.56 at iteration one), gradually decreasing 0.23 after iteration five. Analysis of top-50 novel pairs found at a given iteration showed that 58% of top-50 novel pairs at iteration one were judged as opposites leading to a precision score of 0.61. Only four novel pairs out of top-50 of the last iteration proved to be opposites. Thus the largest number of opposites were found at the first iteration.

Three most frequent patterns found with noun-noun pairs were general patterns *ANT1:conj* ← *as well as* → *conj:ANT2*, *ANT1:conj* ← *and* → *conj:ANT2* and *ANT1:conj* ← *but* → *conj:ANT2*. A variant of pattern with connective *but* was found only in the third iteration with the set of adjective-adjective seeds. Unlike patterns with adjective-adjective seeds, half of patterns found with noun-noun seeds were longer and contained dependencies between subjects and objects.

### 4.3 Results for verb-verb pairs

The annotators agreed least on the classification of 518 pairs found with 18 verb-verb seeds, achieving a Fleiss's kappa score of 0.56. Contrary to our expectations, this set had the lowest precision scores out of the three PoS category sets. Namely, only 15% (78 pairs) were opposites according to the majority vote. They contained pairs like like *trouwen - scheiden* ("to marry - to divorce"), *verhoog - verminder* ("to raise - to decrease"), ontvangen - verzenden ("to receive - to send"). For 77 of them (99%), both words were found in CORNETTO but only 20 were marked as opposites, missing 74% of good instances. Among the 440 pairs judged as non-opposites were correlates *trouwen - samenwonen* ("to marry - to live together"), near-synonyms *bekijk - bezoek* ("to see over - to visit") and frequently co-occurring words like *downloaden - spelen* ("to download - to play").

Looking at the top-50 novel pairs found at a each given iteration only showed that the precision scores were very low at all five iterations ranging from 0.12 at iteration one to 0.04 at iteration five. This suggests that dependency patterns were not able to find many reliable instances of opposites neither with original seeds nor with seeds acquired during iterations.

Among the top three iteration patterns for verb-verb seeds were *to ANT1 or to ANT2*, *ANT1 or ANT2* and *be ANT1 or ANT2*. We also found variants of the patterns *ANT1 as well as ANT2*, *to ANT1 or to ANT2*, *neither ANT1 nor ANT2* and *ANT1 more than ANT2*. Thus, patterns found with each seed set were equivalent.

## 5 Discussion and Future Work

We have studied the application of dependency patterns learned from a treebank for the automatic identification of pairs of opposite words. We presented results for three PoS categories: adjective-adjective, noun-noun and verb-verb pairs. We showed that the results depended on the target PoS category. The best results were achieved for adjective pairs, followed by noun and verb pairs. Analysis of novel pairs found only at a given iteration showed that the most reliable pairs were found at the initial iterations (precision scores of 0.67 for adjectives, 0.56 for nouns and 0.22 for verbs). While results for top-50 novel adjective and noun antonym pairs are comparable with the results from similar pattern-based methods for finding meronyms ([12]) and hyponyms ([25]), contrary to our expectations, dependency patterns were not productive for finding opposites expressed by verbs. One of the reasons for this is that the best patterns found at each iteration are too general. Opposites expressed by verbs are also the least frequent category of sententially co-occurring pairs suggesting that this result might reflect the behavioural preferences of antonymous verbs rather than limitations of a particular automatic method.

Preference for short and general patterns is one of the main shortcomings of the present method. As a result, our algorithm is not able to discover an equivalent of one of the most productive textual patterns for finding opposites "*between X and Y*". Instead coordination construction "*X and Y*" is treated as the shortest path, dismissing the preposition *between*.

The lexical semantic relation of antonymy is not present in the most up-to-date available lexical resource for Dutch CORNETTO ([11]) for 57% of the correct opposites found with adjective seeds, 74% of the opposites found with verb seeds and 93% of the opposites found with noun seeds. This suggests that this method can be used as a supplementary means for improving existing databases. One way to improve the method itself would be to extend the algorithm so that it finds more instances with a given pattern by e.g., using Web data. However, given that Web provides immense data repository, it has yet to be determined whether we need to use dependency patterns or whether PoS tagging as a preprocessing step would be sufficient for antonym harvesting.

## References

[1] Berland, M. and E. Charniak (1999) Finding Parts in Very Large Corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 57–64.

[2] Charles, W.G. and G. A. Miller (1989) Contexts of Antonymous Adjectives. *Applied Psycholinguistics* 10:3, pp. 357–375.

[3] Charniak, E. (2000) A Maximum-Entropy-Inspired Parser. In *Proceedings of*

*the First Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2000)*, pp. 132–139.

[4] Church, K. W. and P. Hanks (1990) Word Association Norms, Mutual Information and Lexicography. *Computational Linguistics* 16:1, pp. 22–29.

[5] Collins, M. (1996) A New Statistical Parser Based on Bigram Lexical Dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-1996)*, pp. 184–191.

[6] Cruse, A. (1986) *Lexical Semantics.* Cambridge: Cambridge University Press

[7] Deese J.E. (1964) The Associative Structure of Some Common English Adjectives. *Journal of Verbal Learning and verbal Behaviour* 3:5, pp. 347–357.

[8] Fellbaum Ch. (1998) *WordNet: An Electronic Lexical Database*. MIT Press.

[9] Randolph, J. J. (2005) *Free-marginal multirater kappa: An alternative to Fleiss' fixed-marginal multirater kappa.* Paper presented at the Joensuu University Learning and Instruction Symposium 2005, Joensuu, Finland.

[10] Hearst, M. (1992) Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pp. 539–545.

[11] Ales Horak and Piek Vossen and Adam Rambousek (2008) The Development of a Complex-structured Lexicon Based on WordNet. In *Proceedings of the 4th International Global WordNet Conference (GWC-2008)*, pp. 200–208.

[12] Ittoo, A. and G. Bouma (2010) On Learning Subtypes of the Part-Whole Relation: Do Not Mix Your Seeds. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-2010)*, pp. 1328–1336.

[13] Jones S. (2002) *Antonymy: A Corpus-based Perspective*. London: Routlegde.

[14] Jones, S., Paradis C., Murphy M.L. and C. Willners (2007) Googling for Opposites - a Web-based Study of Antonym Canonicity. *Corpora* 2:2, pp. 129–155.

[15] Lin D. and Pantel, P. (2001) Induction of Semantic Classes from Natural Language Text. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining (KDD-01)*, pp. 317-322, San Francisco, CA.

[16] Lin D., Zhao Sh., Qin L. and M. Zhou (2003) Identifying Synonyms among Distributionally Similar Words. In *Proceedings of IJCAI-2003*, pp. 1492–1493.

[17] Lobanova, A., van der Kleij, T. and J. Spenader (2010) Defining Antonymy: a Corpus-based Study of Opposites by Lexico-syntactic Patterns. *International Journal of Lexicography* 23:1, pp. 19–53.

[18] Lobanova, A., Spenader, J., van de Cruys, T., van der Kleij, T. and E. Tjong Kim Sang (2009) Automatic Relation Extraction - Can Synonym Extraction Benefit from Antonym Knowledge? In *Proceedings of WordNets and other Lexical Semantic Resources - between Lexical Semantics, Lexicography, Terminology and Formal Ontologies (NODALIDA2009 workshop)*, pp. 17–20.

[19] Marcu, D. and A. Echihabi (2002) An Unsupervised Approach to Recognizing Discourse Relations. in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)* pp. 7–12.

[20] Mohammad, S., Dorr, B. and G. Hirst (2008) Computing Word-Pair Antonymy. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 982–991.

[21] Ordelman R. J. F. (2002) *Twente Nieuws Corpus (TwNC)*. Parlevink Language Technology Group. University of Twente.

[22] Pantel, P. and M. Pennacchiotti (2006) Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *Proceedings of Conference on Computational Linguistics / Association for Computational Linguistics (COLING/ACL-06)*, pp. 113–120.

[23] Pantel, P. and D. Ravichandran. (2004) Automatically Labeling Semantic Classes. In *Proceedings of the HLT-NAACL conference*

[24] Ravichandran, D. and E. Hovy (2002) Learning Surface Text Patterns for a Question Answering System. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, pp. 41–47.

[25] Snow, R., Jurafsky D. and Ng A. (2005) Learning Syntactic Patterns for Automatic Hypernym Discovery. In *Proceedings of the NIPS 17*, pp. 1297–1304.

[26] Spenader, J. and G. Stulp (2007) Antonymy in Contrast Relations. In *Seventh International Workshop on Computational Semantics*

[27] Tjong Kim Sang, E. and K. Hofmann (2009) Lexical Patterns or Dependency Patterns: Which Is Better for Hypernym Extraction? In *Proceedings of CoNLL-2009*, pp. 174–182.

[28] van Noord, G. (2006) At Last Parsing is Now Operational. In Mertens, P., Fairon C., Dister A. and P. Watrin (eds.) *TALN 2006. Verbum Ex Machina. Actes de la 13e Conference sur le Traitement Automatique des Langues Naturelles*, pp. 20–42.

# Annotation of morphology and NP structure in the Copenhagen Dependency Treebanks (CDT)

Henrik Høeg Müller

Copenhagen Business School
Department of International Language Studies and Computational Linguistics
E-mail: hhm.isv@cbs.dk

### Abstract

This paper provides an overview of the annotation design for morphological structure in CDT. The structure of words and phrases is encoded as a dependency tree which can be specified in two different ways: either as an ordinary dependency tree or by means of an abstract operator specification. The dependency notation encodes the internal structure of phrasal compounds and regular NPs, while the operator notation encodes dependency structure within solid orthography compounds and derivationally constructed words. Finally, the paper discusses the semantic labeling system used in CDT and some specific issues related to the annotation of NPs.

## 1 Introduction

The Copenhagen Dependency Treebank (CDT) is an ongoing project which seeks to create a parallel treebank for Danish, English, German, Italian, and Spanish with 80,000 words in each language. The CDT treebanks are based on dependency, but the annotation includes not only syntax, but also analyses of morphological, discourse, and anaphoric structure. This multilevel annotation distinguishes CDT from other treebank projects which tend to focus on a single linguistic level[1], and it has the advantage of not obliging us to limit the kind of linguistic relations that can be annotated, and not having to draw precise, and often arbitrary, boundaries between morphology, syntax, and discourse (for an outline of discourse annotation, see, e.g., Webber [20] or Buch-Kromann et al. [3]). Our main claim is that by means of a primary tree structure supplemented by an inventory of secondary relations we will be able to give a unified account of morphology, syntax and discourse which is theoretically appealing while also providing a good basis for building automatic parsers and MT-systems. However, it is not possible here to

[1] For instance, the Penn Treebank (Marcus et al. [9]) and the Prague Dependency Treebank (Böhmová et al. [1]) mainly concentrate on syntax; the Penn Discourse Treebank (Prasad et al. [13], [14]) and the RST Treebank (Carlson et al. [4]) focus on discourse, and the GNOME project (Poesio [11]) on coreference annotation. The TuBa-D/Z treebank (Hinrichs et al. [6]), however, includes both morphology and coreference annotation and has thus multiple levels of annotation.

account for all the general design principles behind the CDT, and, therefore, as indicated in the title, the centre of attention will be morphology and NPs.

This paper is structured as follows. In Section 2, it is explained how morphological structure is annotated in CDT. In Section 3, focus is on the marking-up of NP structure, and, finally, the most central points are summed up in Section 4, which also includes a short comment on the annotators' evaluation of the system.

## 2 Morphological annotation

*2.1 Operator vs. dependency annotation*

The morphological annotation in the CDT treebanks is only concerned with derivation and composition, since inflectional morphology can be identified and analysed automatically with a high degree of accuracy for all the languages involved in the treebanks.

The complex internal structure of words, word-like phrases and regular NPs is encoded as a dependency tree which can be specified in two different ways: either as an ordinary dependency tree, i.e. similar to syntactic dependency annotation, cf., e.g., Buch-Kromann [2], Buch-Kromann et al. [3], Kromann [8], (the *dependency notation* in Figure 1), or by means of an abstract specification of how the dependency tree for a morphologically complex word is constructed from roots in combination with morphological operators (the *operator notation* in Figure 2).

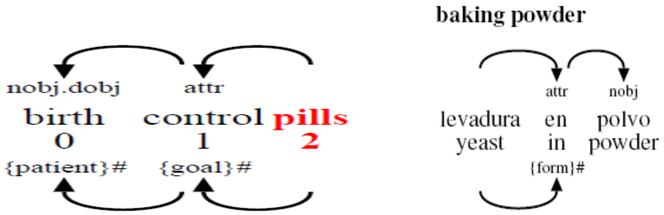


Figure 1. Dependency annotation of the phrasal compounds *birth control pills* (left) and *levadura en polvo* [baking powder] (right).[2]

*Krigsskib*: **skib –[krig]s/GOAL** [war ship]

*Træbord*: **bord –træ/CONST** [wooden table]

Figure 2. Operator annotation of the solid orthography compounds *krigsskib* [war ship] (left) and *træbord* [wooden table] (right).

In other words, the dependency notation specifies the tree directly, whereas the operator notation indicates how the tree can be constructed from a set of operators. The motivation for having these two annotation principles is that we use the dependency notation to encode dependency structure between

[2] The color code (red) and numbers (0, 1, 2, …) are tagging marks and not relevant in this context.

tokens (NPs and word-like phrases) in the automatically produced word tokenisation, while the operator notation is employed to encode dependency structure within tokens (derivations and compounds).

The analyses of the phrasal compounds in Figure 1 can be explained in the following way: The head of *birth control pills* is *pills*. The relation between the head and the non-head *birth control* is non-argumental, i.e. what we understand as one of attribution – basically because the head is non-predicative or non-relational. This relation is indicated by the arrow pointing from *pills* to *control* above the text, with the relation name written at the arrow tip. The other top arrow indicates that *control* functions as governor to the non-head *birth*, which is a noun object equivalent to a corresponding sentence level direct object.

The arrows below the text indicate semantic structure. The non-head activates the telic quale of the head – we refer to it as a "goal" relation – being the general assumption that the qualia of the head can be triggered by different modifiers, in this case a noun phrase (Pustejovsky [15], [16]). The head of *birth control* is predicative/deverbal, and *birth* fulfils the patient role of the head's argument structure (see, e.g., Grimshaw [5]).

In Figure 1 (right), the prepositional phrase headed by *en* functions attributively – being the head unable to project an argument structure – and the noun *polvo* [powder] is syntactically a noun object. The semantic relation established is "form", indicated again by the arrow at the bottom. The hash symbols following the semantic relation labels in both constructions indicate that the phrases in question show composite structure.

The operator annotations in Figure 2 show analyses of minimally complex Danish compounds. *Krigsskib* [war ship] is composed of the modifier *krig* [war], the head *skib* and the linking consonant or interfix -*s*. The annotation should be read in the following way: The minus sign indicates the pre-head position of the modifier, the lexical material of the modifier itself appears in square brackets, then comes the interfix which is a phonetically induced morpheme whose only function is to act as a glue between the head and the modifier, and finally, following the oblique slash, the meaning facet of the head noun selected by the non-head modifier, here a telic meaning relation.

The analysis of *træbord* [wooden table] follows the same principles, but here the meaning component prompted by the modifier is constitutive.

Figure 3 below shows a dependency and an operator annotation of the same Danish compound. The two types of annotation look very different, but they are merely two notational variants for the same underlying abstract dependency tree. So, you could say that the operator notation maps on to a dependency structure with equivalent principles to the ones governing syntactic expansions.

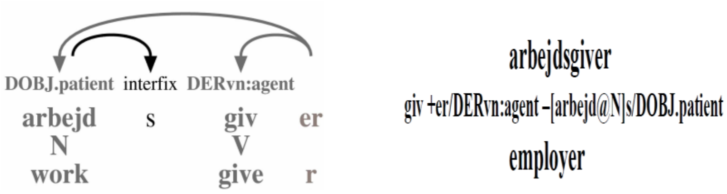


Figure 3. Morphological analysis of the compound *arbejdsgiver* [employer] annotated in dependency notation (left) and operator notation (right).

The example in Figure 3 is slightly more complex than the ones in Figure 1 and 2 because in this case the annotation of compounding is combined with that of derivational morphology. The analysis is as follows: The head of the compound is *giver* [giver], which is a derivationally complex lexeme. The operator "+er/DERvn:agent" indicates that the head is an agent nominalization of the verb *give* [give] triggered by the suffix *-er*. The annotation of the non-head, i.e. "–[arbejd@N]s/DOBJ.patient" indicates its pre-head position, that the lexical material is a noun with the interfix *-s*, cf. [arbejd@N]s, and that it corresponds syntactically to a direct object with the semantic function of Patient. The indication of word class with the specification "*@word-class*" is optional, but it should be indicated when the form is ambiguous, as in this case between a noun and a verb. The governor is the suffix which takes the root as dependent, and the non-head functions as dependent to the root. Generally, the root is governor (head) and the element activating the morphological operation functions as dependent. However, when the operator/affix is transformational or transcategorial, the operator functions as governing head and the root/stem as its dependent.

### *2.2 Operator annotation of different word-classes*

In CDT, the three word-classes nouns, adjectives and verbs are marked-up according to the operator annotation scheme.

As illustrated below, nouns can be morphologically expanded by pre-head modifiers and/or post-head modifiers. The position of the modifier is indicated simply as a minus sign for pre- and a plus sign for post-modification. The modifier itself can be a traditional prefix or suffix, or it can be a lexical root in the form of the non-head of a compound. The positional indication, i.e. plus/minus, says nothing about that.

**Prefixed noun:**

(1) *antihero*: hero –anti/NEG:contr

**Suffixed noun:**

(2) *payment*: pay +ment/DERvn:core

**Noun compound:**

(3) *brødproducent*: producer ! +nt/DERvn:agent –brød/DOBJ.patient
[bread producer]

The adjectives in (4)-(7) are annotated according to the same annotation principles as the nouns, but the semantic categories for adjectives differ from those of nouns with respect to the languages covered by CDT, cf. Table 1 below.

**Prefixed adjective:**

(4) *inactive*: active –in/NEG:contr

**Suffixed adjectives:**

(5) *folkelig*: folk +e[lig]/DERna:rel.norm
[folksy/popular]

(6) *historic*: history ! +ic/DERna:rel.norm

**Adjectival compound:**

(7) *good-sized*: size +d/DERna:rel.norm –good-/EVAL

The annotation of verbs is slightly different in the sense that they cannot carry derivational suffixes because the post-head position is restricted to inflectional endings, at least in the languages dealt with in CDT.

**Prefixed verbs:**

(8) *enjabonar*: jabón –+[en][ar]/DERnv –en/AGENT
[in-soap = do the lathering]

(9) *dislike*: like –dis/NEG:contr

**Verbal compound:**

(10) *lungeoperere*: operer –lunge/DOBJ.patient
[lung-operate]

Summarizing, an operator has the form "*pos affix*/*type*". The field *pos* specifies whether the abstract affix is attached to its base in prefix position ("–") or suffix position ("+"), or a combination of these (e.g., "–+"). The field *type* specifies the derivational orientation (e.g., "DERvn", {fig. 3}), either in the form of a categorial shift, i.e. a word-class transformation, or not. Moreover, the field *type* semantically and functionally identifies the type and, where relevant, the subtype, of the dependency relation that links the base with the abstract affix (e.g., "NEG:contr", {ex. 1}). The field *affix* specifies the abstract affix and its possibly complex internal structure. The abstract affix may be encoded either as a simple string representing a simple affix or a simple root (e.g., "*er*", "*arbejd*", {fig. 3}), or as a complex string of the form "[*stem*]" or "[*stem*]*interfix*", where "*stem*" encodes the internal structure of the abstract affix in operator notation (e.g.," −[arbejd@N]s/DOBJ.patient", {fig. 3}).

Finally, the number of exclamation marks used (e.g., "*historic*: history **!** +ic/DERna:rel", {ex. 6}) indicates how many letters have been removed from the derivational base in order to add the suffix, and the separation by square brackets (e.g., "*folkelig*: 'folksy/popular': folk +e[lig]/DERna:rel", {ex. 5}) indicates that the suffix "*-lig*" is connected to the base via the thematic vowel "*-e*". With this system of exclamation marks and brackets we are capable of separating linking elements such as thematic vowels, infixes and interfixes, on the one hand, from what is the suffix proper, on the other hand, and it allows

CDT to regenerate the word form in question on the basis of the operator instructions.

A sample of the most important relation types in the morphological annotation is listed in Table 1 below. The different relation types have taken inspiration from the works on morphological categories by Rainer [18] and Varela & Martín García [19]. All the relations can be annotated as either prefixes or suffixes or non-head roots in case of compounds; here they are just listed as they typically appear in the CDT languages. However, it is evident that some derivational meanings are typical for, or perhaps even restricted to, a specific word-class, but in principle any of the semantic relations can be used to describe derivation or compounding within all three word-classes. So, in that sense the system is flexible.

**Relations that typically appear with prefixes**
**SPACE:loc** (location: *intramural = mural −intra/SPACE:loc*)
**SPACE:dir** (direction/origin: *deverbal = verbal −de/SPACE:dir*)
**TIME:pre** (precedency: *prehistorical = historical −pre/TIME:pre*)
**TIME:post** (posteriority: *postmodernism = modernism −post/TIME:post*)
**NEG:contr** (contrast: *antihero = hero −anti/NEG:contr*)
**NEG:priv** (privation: *desalt = salt −de/NEG:priv*)
**AGENT** (causative: *acallar ‘silence’ = callar −a/Agent*)
**TELIC** (telic: *oplåse ‘open’ = låse −op/TELIC*)
**MOD:quant** (quantification: *multicultural = cultural −multi/MOD:quant*)
**MOD:eval** (evaluation: *maleducado* [mal-behaved] = *educado −mal/MOD:eval*)

**Relations that typically appear with suffixes**
**AUG** (augmentative: *perrazo ’big dog’ = perro +azo/AUG*)
**DIM** (diminutive: *viejecito ’little old man’ = viejo +ecito/DIM*)

***Verb derivation*:**
***DERnv*** (noun→verb derivation: *salar 'to salt' = sal +ar/DERnv*)
***DERav*** (adjective→verb derivation: *darken = dark +en/DERav*)
***DERvv*** (verb→verb derivation: *adormecer ’lull to sleep’ = dormir −+[a][ecer]/DERvv*)
***Noun derivation*:**
***DERvn:agent*** (verb→noun derivation: *singer = sing +er/DERvn:agent*)
***DERvn:core*** (verb→noun derivation: *exploitation = exploit@V +ation/DERvn:core*)
***DERan:qual*** (adjective→noun derivation: *bitterness = bitter +ness/DERan:qual*)
***Adjective derivation*:**
***DERva:pas.poten*** (deverbal adjective: *transportable = transport +able/DERva:pas.poten*)
***DERna:rel.norm*** (denominal adjective: *presidential = president +ial/DERna:rel.norm*)

**Relations that typically appear with compounds**
**CONST** (constitutive: *træbord ‘wooden table’ = bord −træ/CONST*)
**AGENT** (agent: *politikontrol ‘police control’ = kontrol −politi/AGENT*)
**SOURCE** (source: *rørsukker ‘cane sugar’ = sukker −rør/SOURCE*)
**FUNC** (function: *krigsskib ‘war ship’ = skib −[krig]s/FUNC*)
**LOC** (location: *loftlampe ‘ceiling lamp’ = lampe −loft/LOC*)

Table 1. Exemplification of relation types in the morphological annotation (relation types with head-switching are italicised).

## 3 Annotation of NP structure

This part of the paper discusses how NP structure compared with sentence level structure is annotated in CDT, concentrating on analogies and differences between these two linguistic levels (Grimshaw [5]). Figure 4 below is a simple example of a syntactic dependency annotation of a sentence. The complements *He*, *her* and *a kiss* are lexically licensed by the head *gave*, i.e. they function as arguments to the governor, while on the phrasal level *kiss* is a dependent of the indefinite article *a*. The arrows point from governor to dependent and the relation name is written at the arrow tip.

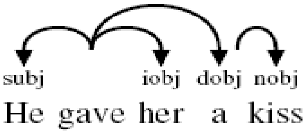


Figure 4. Basic CDT dependency annotation of sentence.

In general, on the sentence level semantic features are not annotated, i.e. a type system for verb-based annotation has not yet been introduced, the CDT does not make use of a semantic labeling system for arguments, and neither do we attempt to identify qualia-relations in a verb-argument context. However, all free adjuncts are labeled semantically according to which semantic relation they establish with the predicate, as illustrated in Figure 5 with a "manner" relation (left), and a relation of "contrast" (right), i.e. *instead of fruits*.

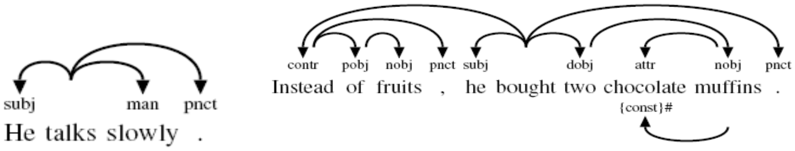


Figure 5. Annotation of sentence level free adjuncts expressing *manner* (left) and *contrast* (right).

With respect to NP-structure, we take our point of departure in the assumption that NPs with deverbal head noun project a dependency structure similar to the corresponding verb, as the top arrows of Figure 6 illustrate. In the dependency annotation above the text, we distinguish between "pobj" and "nobj", on the one hand, and "attr", on the other hand. The syntactic labels "pobj" and "nobj" indicate that the modifying noun or PP is lexically governed by the head, whereas the "attr"-label indicates that this is not the case. "nobj" is also used more widely when a noun is governed by an article or a preposition.

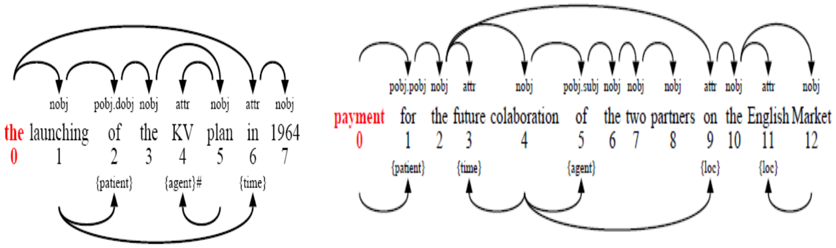
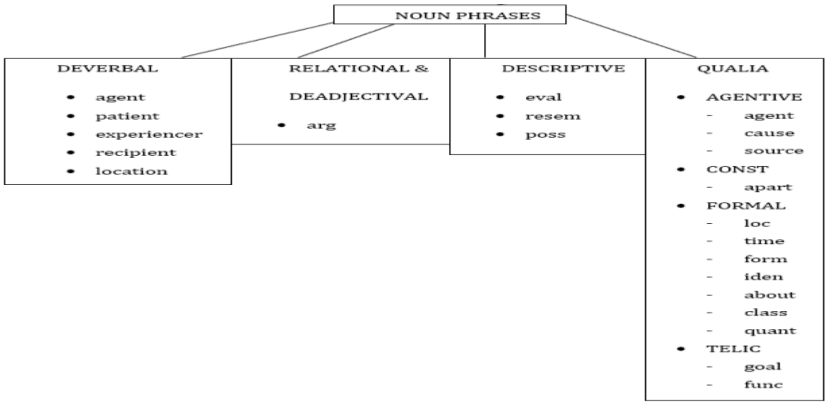


Figure 6. Full syntactic and semantic annotation of NPs.

The arrows at the bottom illustrate how we on the NP-level – contrary to the sentence level – use a system of semantic labeling for both lexically governed arguments (when the head noun is deverbal, relational or deadjectival, and, hence, projects an argument structure) and free adjuncts (when the head noun is non-predicative, and, hence, establishes a descriptive or qualia-type relation). The inventory for argument labels (deverbal, relational, deadjectival) and adjunct labels (descriptive, qualia) is listed in Table 2. There is a substantial overlap between sentence level and nominal level adjunct labels, but on the sentence level CDT makes use of a number of special semantic relations, such as certain pragmatic adverbials, and, e.g., the contrast adverbial in Figure 5 (right), which for various reasons do not seem to occur on the nominal level. Generally, i.e. both in the analysis of sentence level adjuncts, NP modification and with respect to derivational morphology, we have sought to let the qualia-structure be a guiding principle for the organization of the semantic inventory in CDT. This goes also for the anaphoric relations and discourse structure, whose annotation falls outside the scope of this paper. However, this strategy does not imply that it is possible to account for any semantic relation with point of departure in the qualia-structure, as also indicated in Table 2.



NOUN PHRASES

| DEVERBAL | RELATIONAL & DEADJECTIVAL | DESCRIPTIVE | QUALIA |
|---|---|---|---|
| • agent<br>• patient<br>• experiencer<br>• recipient<br>• location | • arg | • eval<br>• resem<br>• poss | • AGENTIVE<br>- agent<br>- cause<br>- source<br>• CONST<br>- apart<br>• FORMAL<br>- loc<br>- time<br>- form<br>- iden<br>- about<br>- class<br>- quant<br>• TELIC<br>- goal<br>- func |

Table 2. Semantic relations for annotating NPs.

Evidently, the head of an NP is not always derived from a predicate, and in CDT we calculate with two other types of head nouns, i.e. relational head nouns and absolute head nouns.

Relational nouns can be divided into, on the one hand, partitive and quantitative expressions which denote arbitrary parts of something and only exist due to the whole of which they form part (such as *top, piece, liter, centimeter*, etc.), and, on the other hand, role and kinship terms (such as *member, president, mother, brother*, etc.), which have independent existence and can be employed in an absolute, non-relational manner (such as *He saw a president on the street/I am a father*).

When the head is of the first type, i.e. denotes arbitrary parts of something, we use the label "apart"[3], cf. Figure 7 (left), and the semantic relation goes from the non-head to the head, which is a consequence of split headedness in the sense that the morpho-syntactic head, N1, functions as a specifier and N2, the second noun, is the semantic head. When the head is of the second type, in case of role terms for instance, we use the label "arg", cf. Figure 7 (right), – without further intents of semantic qualification – and the arrow goes the normal way from head to non-head. This label is also used when the head noun is deadjectival.

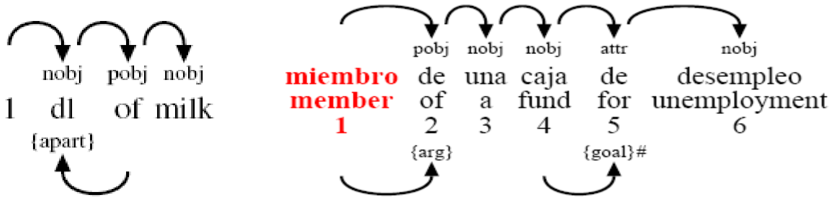


Figure 7. Annotation of NPs with relational head nouns.

When the head noun is absolute, i.e. it has no connection to relational or deverbal nouns in the sense that it does not select or imply reference to any other element, cf. Figure 8, its predicative force is identified through a slightly expanded set of qualia-like relations. Our assumption is that one of the qualia-roles listed in Table 2 is activated by a modifier, which has the form of a noun or a PP.

[3] The ”apart” relation is listed under the constitutive quale (CONST), cf. Table 2, which normally only applies when the head in non-relational. However, because of the special "partitive" nature established by nouns denoting arbitrary parts, the "apart" relation is categorized under the qualia-structure.

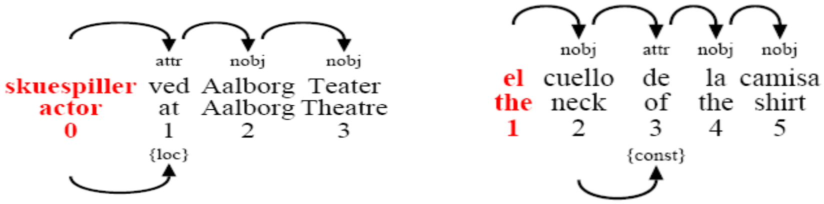


Figure 8. Annotation of NPs with non-predicative/relational head noun.

## 4 Conclusion

The main conclusions and perspectives of the design principles behind the CDT annotation of morphological and NP structure are the following.

The operator annotation and the dependency annotation build on the same underlying principles. They are merely two manifestations of the same system. We need the operator annotation system to account for the internal structure of tokens, in the form of derivations and solid orthography compounds, and the dependency system to tackle relations between tokens.

When building the morphological component of CDT, we sought to establish an intimate analogy between the original dependency based, sentence level framework and the morphological analysis principles. Both systems part from the basic assumption that coherent linguistic units, in the form of sentences or words, are determined by a dependency structure in which each word or morpheme is assumed to function as complement or adjunct to another word or morpheme, called the governor. By their lexical make-up or content, governors license the complements which function as arguments, whereas the adjuncts function as free modifiers, i.e. their presence is not lexically determined by the head. This distinction between arguments and modifiers, between lexically bound and unbound elements, applies at all levels of CDT.

On the sentence level, only the free adjuncts have been annotated with respect to semantics, i.e. every adverbial modifier has been tagged with a semantic label indicating its relation to the predicate. However, in the annotation of morphology and NPs, we have gone one step further, you could say, by introducing a semantic labeling system with which we seek to identify the relations triggered by different affixes when they are attached to their lexical bases, including, not least, argument roles inside NPs and the head noun qualia-values activated by noun and PP modifiers.

Both in the analyses of sentential adjuncts, NP modification and with respect to derivational morphology, the qualia-structure has been a guiding principle of how the semantic component of CDT is organized. Many relations we know from one linguistic level are reproduced or somehow imitated on other levels, and, therefore, it is theoretically appealing to try to unify the inventory. In that respect the qualia-structure is attractive because it provides a template which is sufficiently general for structuring the relations.

The combination of morphological annotation, in its broadest sense, and alignment of parallel texts – an aspect of CDT which has not been described in this paper – will provide a good basis for doing multilingual language processing in the form of building machine translation systems. Just to mention one aspect, it is crucial to know the nature of the semantic relations that hold between NP-constituents in the source language in order to construct an analogous and well-formed nominal concept in the target language, e.g., with respect to the use of prepositions, constituent order, linking vowels or consonants, etc. (see, e.g., Johnson & Busa [7]). It is also expected that the rule-based, non-automated, hand-annotation approach, which is the actual practice of CDT, over time, and on the basis of statistical models, can develop into a more or less semi-automatic annotation system, especially taking into consideration that we do annotation on all linguistic levels. Apart from providing a basis for building automatic parsers and MT-systems, the combination of morphological annotation and alignment of parallel texts will facilitate specific inquiries into morphological cross-linguistic contrasts.

Despite the semantic granularity and complexity of CDT, the annotators generally evaluate the morphological component positively in terms of functionality and user friendliness. They especially emphasize that the hierarchical organization of the system facilitates a relatively smooth narrowing down of options to a few of the best available. Also, the high degree of specificity of the labels is mentioned as a factor which eases the final, detailed assessment. On the more critical side, the annotators find that it has been complicated and time-consuming to learn the system. In comparison with, e.g., the annotation of anaphora and discourse, the marking-up of morphological structure seems to require a deeper understanding of the languages in question both in terms of morphological structure, etymology, and (non)-productivity of certain derivational patterns, again according to the annotators. In comparison with, e.g., GLML-annotation (Generative Lexicon Markup Language) of lexical semantic structure (Pustejovsky et al. [17]), which can be done by any (native) speaker of English without prior training or too much instruction, the annotation of derivations, compounds and NPs in CDT requires a certain level of linguistic and systemic expertise.

## References

[1] Böhmová, A., Hajič, J., Hajičová, E. & Hladká, B. (2003). The Prague Dependency Treebank: a three-level annotation scenario. In A. Abeillé (ed.). *Treebanks: Building and Using Parsed Corpora*. Dordrecht: Kluwer Academic Publishers.

[2] Buch-Kromann, M. (2006). Discontinuous Grammar. A dependency-based model of human parsing and language learning. Doctoral dissertation. Copenhagen: Copenhagen Business School.

[3] Buch-Kromann, M., Korzen, I. & Müller, H. H. (2009). Uncovering the 'lost' structure of translations with parallel treebanks. In I. M. Mees, F. Alves, & S. Göpferich (eds.). *Methodology, Technology and Innovation in Translation Process Research. Copenhagen Studies in Language* 38: 199-224.

[4] Carlson, L., Marcu, D. & Okurowski, M. E. (2001). Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue.*

[5] Grimshaw, J. (1990). *Argument structure*. Cambridge: MIT press.

[6] Hinrichs, E., Kubler, S., Naumann, K., Telljohann H. & Trushkina, J. (2004). Recent developments in linguistic annotations of the TuBa-D/Z treebank. In *Proceedings of the Third Workshop on Treebanks and Linguistic Theories*. Tübingen, Germany. 51-62.

[7] Johnston, M. & Busa, F. (1999). The compositional interpretation of compounds, In E. Viegas (ed.). Breadth and Depth of Semantics Lexicons. Dordrecht: Kluwer Academic. 167-87.

[8] Kromann, M. T. (2003). The Danish Dependency Treebank and the DTAG treebank tool. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT 2003), 14-15 November, Växjö*. 217–220.

[9] Marcus, M. P., Marcinkiewicz, M. A., Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19(2). 313–330.

[10] Mladová, L., Zikánová Š. & Hajičová, E. (2008). From sentence to discourse: building an annotation scheme for discourse based on Prague Dependency Treebank. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LCREC 2008).* 2564–2570.

[11] Poesio, M. (2004). Discourse annotation and semantic annotation in the GNOME corpus. In *Proceedings of the ACL Workshop on Discourse Annotation.*

[12] Prasad, R., Dinesh, N., Lee, A., Joshi, A. & Webber, B. (2006). Annotating attribution in the Penn Discourse TreeBank. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text.* 31–38.

[13] Prasad, R., Miltsakaki, E., Dinesh, A, Lee, A., Joshi, A., Robaldo L. & Webber, B. (2008a). *The Penn Discourse Treebank 2.0. Annotation Manual*. (IRCS Technical Report IRCS-08-01). University of Pennsylvania: Institute for Research in Cognitive Science.

[14] Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A. & Webber, B. (2008b). The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08).*

[15] Pustejovsky, J. (1991). The Generative Lexicon. *Computational Linguistics* 17: 409-441.

[16] Pustejovsky, J. (1995). *The Generative Lexicon*. Cambridge (Mass.). London, England: MIT Press.

[17] Pustejovsky, J., Rumshisky, A., Moszkowicz, J. L. & Batiukova, O. (2008). GLML: A Generative Lexicon Markup Language. https://sites.google.com/site/genlexml/papers

[18] Rainer, F. (1999). La derivación adjectival. In I. Bosque. & V. Demonte (eds). *Gramática Descriptiva de la Lengua Española*. Madrid: Espasa. Volume 3, chapter 70, 4595–4643.

[19] Varela, S. & Martín García, J. (1999). La prefijación. In I. Bosque. & V. Demonte (eds.). *Gramática Descriptiva de la Lengua Española*. Madrid: Espasa, volume 3, chapter 76, 4993–5040.

[20] Webber, B. (2004). D-LTAG: extending lexicalized TAG to discourse. *Cognitive Science* 28: 751–779.

# Using the linguistic knowledge in BulTreeBank for the selection of the correct parses

Petya Osenova and Kiril Simov
Linguistic Modelling Laboratory
Bulgarian Academy of Sciences
petya@bultreebank.org, kivs@bultreebank.org

### Abstract

In this paper, a method is presented for transferring of linguistic knowledge between two treebanks of Bulgarian, constructed within the same linguistic theory, but in its different versions and from different perspectives. BulTreeBank (BTB) follows HPSG94 and the sentences have been analyzed per se. The target Treebank BURGER is constructed by an HPSG grammar for Bulgarian. The linguistic information in BTB and BURGER is presented in the format of lexical categories and dependency relations. A set of transferring rules on the level of the categories (or list of categories) is defined to ensure the compatibility of the representations. Currently our goal is to provide a mechanism for the usage of the linguistic knowledge encoded in BTB as a set of discriminating properties for the selection of the correct analyses produced by BURGER.

## 1 Introduction

Any annotation effort over some language resource would take into account the usability of the annotated resource. The question of which treebank annotation is better has been discussed in many works – see for example Kübler *et al*. (2008). In the current project, we aim at a treebank which to support a Bulgarian-English HPSG-based statistical machine translation. For this task, a parallel treebank is needed, which to meet at least the following requirements: the analyses in both languages to be comparable; and the size to allow estimation of parameters for correspondences on different levels of linguistic analyses. In our case, the first requirement is ensured by sharing as many categories and principles in the analyses of both languages as possible. The second requirement imposes the usage of automatic methods in the creation of the treebank. Thus, we have to use two parsers – one for English and one for Bulgarian – which produce similar analyses with respect to common HPSG principles. For English we envisage to use the English Resource Grammar (ERG) (Flickinger 2000) and for Bulgarian we are developing a resource grammar based on the same principles.

This paper presents our first experiments on transferring of the linguistic knowledge between two HPSG-oriented resources of Bulgarian with the aim to disambiguate the analyses in one of them. The first resource is the

Bulgarian HPSG-based Treebank – BulTreeBank – (Simov et al. 2004), and the second one is another HPSG-based treebank under construction on the base of the Bulgarian Resource Grammar (BURGER). BulTreeBank (BTB) is based on an annotation schema, designed with respect to HPSG94 (Pollard and Sag 1994). It was semi-automatically constructed by using partial parsers. The full analyses had been completed manually in an XML format. The annotators had at disposal a reporting service, which prompted the places of errors when their decisions did not conform to the specified nodes and attributes in the DTD. The final analyses have been checked by two people. We consider this source reliable with respect to the following annotation levels: morpho-syntactic level (manually annotated), constituent level (partially automatically annotated, manually checked and completed), dependency relations – within each constituent the head-dependent relations have been annotated. Additionally, named entity annotation, co-reference annotation (relations – *equality*, *member-of*, *subset-of*), annotation of ellipses have been provided. BURGER (Osenova 2010) is an HPSG grammar under implementation by customizing the Matrix grammar. A Treebank to be constructed on the base of BURGER would be a treebank in which the correct analyses produced by BURGER are selected and stored. In this we follow the Redwood approach (Oepen et al. 2002a, 2002b). Here we investigate the way in which the two resources can be used in order to construct the BURGER Treebank via transfer of knowledge from BTB.

The result of the construction of BURGER Treebank will be used for the construction of a parallel Bulgarian-English treebank. The main usage of such a treebank is the implementation of machine translation system between the two languages. The steps of the BURGER Treebank annotation include:

- Selection of parallel sentences from a given aligned parallel corpus;
- HPSG analysis of corresponding sentences;
- Establishing of correspondences between the HPSG analyses.

The first step is relatively easy as much as already there are reasonably sized Bulgarian-English parallel corpora. The third step requires the analyses of Bulgarian sentences and the analyses of the English sentences to be comparable. In order to achieve this, the sentence analyses have to be modelled in the same way for both languages (Step 2). Our approach is based on the usage of the same grammar formalism – HPSG as implemented within Matrix grammar, thus, we will have similar grammars – ERG for English and BURGER for Bulgarian, implemented in the same grammar development environment – Linguistic Knowledge Builder (LKB). On the other hand, a construction of a grammar with wide coverage is a long term project which we cannot achieve within our current project. Thus, we need to reuse as much as possible from the already available resources for both languages. In this paper, we report a case study of the possibility to transfer the linguistic knowledge which is already incorporated within BulTreeBank in order support the creation of the BURGER treebank and its related grammar.

The structure of the paper is as follows: in the next section a brief overview on the related works is provided; then a comparison is presented between the annotation schemas behind BTB and BURGER; in Section 4 the linguistic analyses of both approaches are discussed; Section 5 comments on the transfer of the linguistic knowledge for selecting the good analyses; Section 6 presents the implementation of the transferring rules; the last section concludes the paper and gives some directions for future research.

## 2 Related Works

There are various approaches to transferring of knowledge from a treebank with respect to a specific task. Some works focused on converting existing constituent-based treebanks into dependency format (Daum et al. 2004), or from one linguistic theory into another theory (Hockenmaier 2006) and many others. The treebanks are used also for extracting lexical types and items for supporting a hand-crafted grammar (Cramer and Zhang 2009). Our task is to use the information in a treebank to select correct analyses produced by a parser. Having started the development of BURGER grammar and the related BURGER Treebank, we will gain from all the components of the developed infrastructure, such as grammar developing workbench (LKB), parsing environment (LKB and PET), profiling software ([incr tsdb()]), etc. The important idea to us is the mechanism behind the development of the Redwoods treebank. This treebank was compiled by coupling ERG and a tree selection module of [incr tsdb()] (Oepen et al. 2002b and Oepen and Callmeier 2000). ERG produces very detailed syntacto-semantic analyses of the input sentence. For many sentences, LKB overgenerates, producing analyses that are not acceptable. From the complete analyses different components can be extracted in order to highlight different views over the analyses: (1) derivation trees composed of identifiers of lexical items and constructions used to build the analysis; (2) phrase structure trees; and (3) underspecified MRS representations. From these types of information the most important with respect to the treebank construction is the first one, because it is good enough to support the reconstruction of the HPSG analysis by a parser. The steps of constructing the Redwood treebank are:

- LKB produces all possible analyses according to the current version of ERG;
- The tree comparison module provides a mechanism for selection of the correct analyses;
- The selection is done via basic properties (called also discriminating properties) which discriminate between the different analyses;
- The set of the selected basic properties are stored in the treebank database for later use in case of treebank update.

In our work, we take all the LKB analyses of the Bulgarian sentences produced by the current version of BURGER. Then we discriminate on the derivation trees because the information there is enough for the full analyses to be determined. Also the derivation trees are used in Redwood treebank setting to define the basic properties. As it was mentioned above, our idea is to use the analyses in BTB to extract the necessary discriminating properties.

The main difference from the manual selection of the correct analyses in our settings is that we can use the total linguistic knowledge, represented in BTB, instead of a predetermined list of discriminating properties. There exist two related tasks: (1) how to extract the discriminating properties from BTB, and (2) how to map them to BURGER analyses. Hence, some ideas have been used from the area of transformation of treebanks and transfer of linguistic knowledge. Our work is based on (Simov 2004), (Chanev et al. 2007) and others. A discriminating property has to be easy to determine within the BURGER analyses and easy to extract from BulTreeBank. In order to facilitate this, we use the ideas from the above mentioned works for transforming the treebank in a new format that would allow a better comparison. Ideally, the transformation has to be done on the knowledge level only, without references to actual implementation formats.

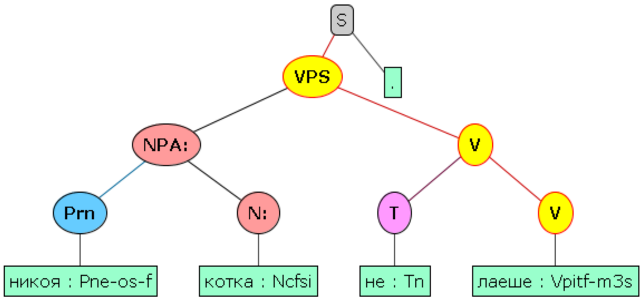
## 3 Annotation schemata: BTB vs. BURGER Treebank

The annotation schema behind BTB (Osenova and Simov 2007) generally follows the HPSG94 linguistic model. It incorporates the universal principles, such as Head Feature Principle, Valence principle, etc. In addition, it follows the hierarchical approach when attaching dependents to their heads. First, the complements are attached, then the subject being an external argument, and finally – the adjuncts. It should be noted that the complements are attached together, by one operation only. Additionally, in BTB the constituent structure is separated from the word order. It means that the topic-focus layer is not distinguished. In such a paradigm, crossing branches are allowed, and three types of discontinuity are envisaged (scrambling, topicalization and mixed). The implementation is in XML, where the XML tree structure is exploited to represent the constituent structure as much as possible with encoding of crossing branches via ID and IDREF attributes. The visualization takes the form of the XML tree and represent it as close as possible to the canonical syntactic trees. The dependency relations are encoded into the syntactic labels. For example, VPC means verbal phrase with a complement.

Apart from the phrase level, another level has been introduced – functional. It handles the various types of clauses (CLR, CLDA, CLQ, and CL), coordination, co-referenced pro-dropness, etc. BTB takes into account the types of named entities (*person*, *organization*, *location* and *other*), various co-references within the sentence as well as the ellipses.

The layers in BTB are modelled separately. Morphological analyses come first. The ambiguous ones have been disambiguated manually. Then chunks have been analyzed, and finally – full analyses with handling the specific attachments, discontinuities and cases of ellipsis. Non-local dependences are handled by the discontinuity markers only.

BTB introduces phrase structures and dependency relations, but lacks feature structures as well as a separate semantic layer of representation. The semantics can be derived as follows: the predicate structure via the dependency labels (arity) and co-references (control, pro-dropness); the relations – via the functional labels (nominalizations, subordinate clauses among others) and co-references (possession). The scope of quantification is present only in the selected interpretation by the annotator. Additionally, the analysis of names shows the semantically correct analysis with respect to subject and complement selection.

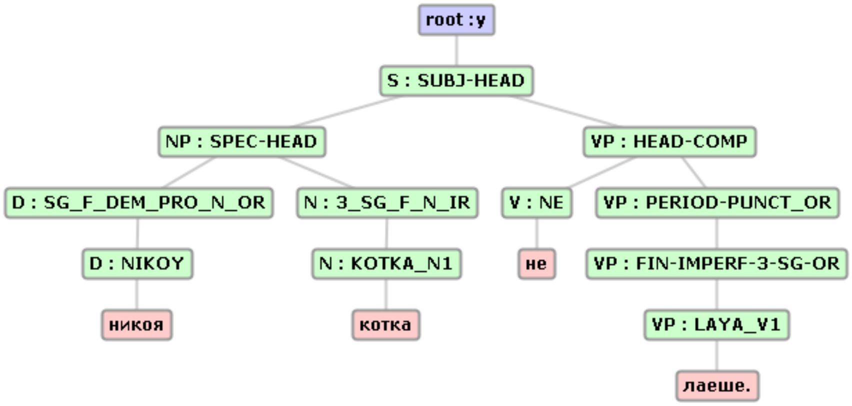


In the above picture the sentence (1) is presented:

(1) Никоя котка не лаеше.
Nobody cat no was-barking.
No cat was barking.

The determiner ‘nobody’ is viewed as an adjunct within the NPA. The phrase is also a subject to an intransitive verb.

The annotation schema of BURGER Treebank strictly follows the principles behind the BURGER grammar. Therefore, it is in accordance with Matrix grammar and other Matrix-based grammars viewed as best practices. In contrast to BTB, where the annotator had to decide on the correct analysis/analyses according to his/her knowledge using only partial analyses, in BURGER the most appropriate analysis/analyses have to be selected among the all produced by the grammar ones. I.e. the annotator is faced with multiple analyses before his/her selection. BURGER aims at combining all the linguistic levels – morphology, syntax and semantics. At the moment, the morphological module produces analyses which are not disambiguated. The syntactic one produces all the possible structures, including topicalizations where appropriate. The semantic module, which is encoded in MRS, gives information about the various relations, predicate structure, control, etc. Syntactically, BURGER introduces a more relaxed schema. It tolerates various types of attachments since it follows the assumption that all possible syntactic structures are allowed, and later on the best one will be chosen via some appropriate mechanism (statistical one or comparison against a gold standard or another). This presupposed freedom has two dimensions: – spurious-like ambiguity, such as adverb attachment to both – VP and S nodes; and non-fixed attachment of arguments. For example, subject might be attached to the head after the adjunct had been attached; or adjuncts might be attached to the head before the complements. In this way, no crossing branches are allowed. Also, each dependent is attached to its head one by one, irrespectively of being a complement or an adjunct. The next picture presents a tree of the BURGER counterpart of the sentence (1).



# 4 Comparing of the gold linguistic analyses

In order to make the comparison between the two types of annotation possible, a case study was performed. At the moment, BURGER covers the Matrix testset (Bender et al. 2002) with a slight extension. The set comprises 194 grammatical sentences with one or more analyses. First, the correct analysis/analyses was/were selected manually for each sentence. Then the same set was manually annotated with respect to the BTB annotation scheme.

The phenomena that are demonstrated by the testset are as follows: various predicate constructions (intransitive, transitive), control, modification, quantification, illocutionary force (questions, imperatives), clauses (relatives and reduced relatives, if-clauses, that-clauses), modals, negation, copula constructions, hybrid categories (deverbals, gerunds), light constructions, coordination, nominalization, quantification. The typical phenomena in Bulgarian include: clitic doubling, pro-dropness, double negation, some basic verb clusters (da-constructions or future tense without or with clitics), clitics in NPs.

In BURGER, from 654 analyses, 81 analyses are unique. For the rest, 277 have been chosen as good and 27 as possible, but rare. Altogether 348 analyses have been rejected, which makes more than 50 % of the produced ones. This result proves that a mechanism for disambiguation is needed (as expected). In the BTB version there are 207 analyses. From them only 13 cases have 2 analyses. They are mainly cases of topicalization readings. Only 3 cases give attachment varieties.

Concerning the syntactic modeling of the specific phenomena, there are several differing but comparable interpretations in both gold datasets. For example, in BTB phrases like ‘every cat’ or ‘some cat’ are analyzed as NPs, while in BURGER they are analyzed as head-specifier phrases. However, the head is still the noun. BTB distinguishes among pragmatic and other adjuncts. BURGER makes a distinction among intersective and scopal adjuncts. In BTB the subordinators and complementizers are viewed as markers. The projections are therefore functional labels. Then, they either are selected as complements, or they modify phrases. In BURGER both types of linking words take the introduced clauses as their complements. The coordination in BTB is analyzed in a flat way, i.e. as a non-headed phrase. In BURGER it is analyzed in levels (bottom, middle and top), and is considered a headed phrase. The question polar particles project lexical nodes in BTB, while phrasal ones of the type head-interesective modifier in BURGER. In BTB, the clitics project the lexical label of their heads, while in BURGER they undergo special head-clitic rules. The negative particle in BTB also projects the lexical label of its head. However, in BURGER it is treated as a verb, which takes a complement. To sum up, most of the analyses of both schemas are comparable, but there is also a need of formulating transferring rules, which to ensure the correct mapping.

## 5 Transfer of Linguistic Knowledge and Disambiguation

Although lacking a semantic layer, BTB analyses have semantically-oriented elements: dependency relations, named entities, co-references, hierarchical constituent structure. The syntactic structure transition seems more trivial as much as the analyses go into the same direction with only slight differences.

The main sources of ambiguity and multiple analyses are as follows: (1) morphological ambiguity, (2) various places of attachment, (3) neutral vs. focused ordering of constituents, (4) proliferation of several competing rules for the same item. They act separately or in various combinations. The more combinations among them, the more analyses appear as results.

Let us comment in more detail on the above sources of ambiguity. The first case produces all morphosyntactic possibilities. For example, in sentence (2) the verb is ambiguous between present and aorist tense of the perfective verb ‘give’. However, present tense is not grammatical:

(2) Абрамс **даде** цигара на Браун.
Abrahms **gives/gave** cigarette to Brown.
Abrahms is **giving/gave** a cigarette to Brown.

BTB analysis in this case would be only one – with the aorist tense. This information is used for 100 % of disambiguation during the transfer of linguistic knowledge to BURGER Treebank.

Case 2 from the above list produces all possible attachments irrespectively to the meaning of the sentence. For sentence (2) there are incorrect analyses, which attach the PP ‘to Brown’ to the noun ‘cigarette’ besides the correct ones, in which the PP ‘to Brown’ is attached to the verb as its second complement. Another example is sentence (3):

(3) Котката е **в градината**.
Cat-the is in garden-the
The cat is in the garden.

The PP is attached not only as a complement to the copula (as expected), but also as a modifier to a verb-complement phrase (as rejected in this case). In BTB there is only one analysis, namely the one with the complement. Thus, again – 100 % of such cases are disambiguated.

Source 3 introduces sentences with topicalized constituents. Let us consider sentence (4):

(4) **Онова куче** преследваше **Браун**.
That dog was-chasing Brown.
That dog was chasing Brown.

The canonical reading is the one in which the dog is the chaser and Brown is the chased one. In the topicalized version, it is vice versa. In BTB there are these both analyses presented with the preference to the first one.

The fourth case is triggered when the same item can undergo more than one rule. For example, for sentence (2) analyses are generated with the semantically empty preposition 'на' (dative) as well as with the modifying preposition 'на'. The latter analyses have to be rejected in this reading.

# 6 Implementation

As it was discussed earlier in the paper, our mechanism for transferring of linguistic knowledge from BulTreeBank to BURGER Treebank is based on the ideas of the treebank transformation. We decided to use a common target format to represent the important knowledge from both treebanks. The procedure is as follows:

- The analyses from BURGER are transformed into a new format;
- The corresponding analyses from BulTreeBank are also transformed into the new format;
- The knowledge within the new representations is unified on the basis of correspondences rules;
- The parse selection is done on the basis of comparing both unified representations.

In order to facilitate the comparison, we decided to use as a common format a dependency-like representation as follows. Each sentence is represented as a list of wordforms:

*w1 w2 w3 … wn*

This representation is necessary in order to keep track of word forms used in the lexical and head-dependent descriptions and their word order.

*Lexical elements:*

*wk:pos list-of-categories*

In this part of the representation, all the lexical categories that dominate the wordform in the representation of the corresponding treebank are stored. We assume that this list describes the lexical features of the wordform. Also, the position of the wordform in the sentence is stored.

*Head-dependent pair:*

<*wi:posi*, *wj:posj*> *list-of-categories*

For any two wordforms in the sentence where one of them is a lexical head of the other, the category of the minimal path between the two wordforms is stored. Sometimes we need to include additional information from the unary branches in order to have all the relevant information represented. Here is an example from the testset:

(5) Кога лаеше кучето
when barked dog-the
When did the dog bark

**BulTreeBank case:**

(**VPA** (**Adv** (**Pit** кога)) (**VPS** (**V** (**Vpitf-m3s** лаеше)) (**N** (**Ncnsd** кучето))))

*Lexical elements*:

| | |
|---|---|
| кога:1 | (Adv, Pit) |
| лаеше:2 | (V, Vpitf-m3s) |
| кучето:3 | (N, Ncnsd) |

*Head-dependent pair*:

| | |
|---|---|
| <кога:1 лаеше:2> | (VPA) |
| <лаеше:2 кучето:3> | (VPS) |

**BURGER case:**

There are two analyses, which attach the adverb either before the subject had been attached, or after it has been attached:

(**MOD-INT-OTHER-PHRASE**
(**ADV** кога)
(**HEAD-SUBJ**
(**FINITE-IMPERF-THIRD-SG00476-ORULE** лаеше)
(**THIRD_SG_NEUTER_NOUN_IRULE**,
**DEF-THIRD_SG_NEUTER_NOUN_ORULE**,
**BARE-NP** кучето)
)
)

and

(**HEAD-SUBJ**
(**MOD-INT-OTHER-PHRASE**
(**ADV** кога)
(**FINITE-IMPERF-THIRD-SG00476-ORULE** лаеше) )
(**THIRD_SG_NEUTER_NOUN_IRULE**,
**DEF-THIRD_SG_NEUTER_NOUN_ORULE**,
**BARE-NP** кучето)
)

They have the same representations[1] in our format:

*Lexical elements*:

| | |
|---|---|
| кога:1 | (ADV) |
| лаеше:2 | (FINITE-IMPERF-THIRD-SG00476-ORULE) |
| кучето:3 | (THIRD_SG_NEUTER_NOUN_IRULE,<br>DEF-THIRD_SG_NEUTER_NOUN_ORULE) |

*Head-dependent pair*:

| | |
|---|---|
| <кога:1 лаеше:2> | (MOD-INT-OTHER-PHRASE) |
| <лаеше:2 кучето:3> | (HEAD-SUBJ) |

Having these two representations, we need to use the rules for mapping of lists of categories between the two treebanks. Our rules are directed from

[1] This fact demonstrates one of the benefits of our representation – namely, that it is an indicator of spurious analyses.

BulTreeBank to BURGER Treebank, since our goal is to select the correct analyses in BURGER treebank. Thus, the rules have the form:

<list-of-BulTreeBank-categories> = <list-of-BURGER-categories>

Here are some examples

(Adv, Pit) = (ADV)
(V, Vpitf-m3s) = (FINITE-IMPERF-THIRD-SG00476-ORULE)
(N, Ncnsd) = (THIRD_SG_NEUTER_NOUN_IRULE,
DEF-THIRD_SG_NEUTER_NOUN_ORULE)
(VPA) = (MOD-INT-OTHER-PHRASE)
(**VPS**) = (HEAD-SUBJ)
(**VPS**) = (SUBJ-HEAD)

The result from the application of these rules is a new representation of the linguistic knowledge extracted from BTB, which is unified with the representation of the BURGER analyses. It can be used to select the correct BURGER analyses by comparing the two sets of descriptions. Note that the last two rules demonstrate the mapping with respect to word order which is explicitly encoded in the labels used by the BURGER grammar. Generally, such rules overgenerate over the BTB representations. As a result, we have more than one unified representation for one BTB analysis. In order to select a correct BURGER parse, we require an equality of the sets. Also, a procedure to go below the VPS label is needed in order to determine the word order between the head and the subject. This step is trivial in BTB, since the head is determined in most of the cases by its label[2].

The case study has shown that our idea of using BTB as a discriminator of the analyses is justified. 86 % of the correct analyses produced by BURGER were successfully selected by the discrimination properties extracted from BTB. The problematic cases refer to two linguistic presentations: coordination and complementation in NP. The first one needs a more elaborate set of rules, which to relax the BTB language model, since it rejects otherwise acceptable analyses. For example, BTB would accept the analysis in (6a) where there is a co-reference between the subject of the first conjunct and the pro-drop subject of the second, but would reject the analysis in (6b) where the subject is viewed as common to both predicates. The reason lies in the strong hierarchical mechanism of subcategorization:

(6a) [Кучето пристигна] и [залая].
[Dog-the came] and [started-to-bark].
[The dog came] and [started to bark].

(6b) Кучето [пристигна и залая].
Dog-the [came and started-to-bark].
The dog [came and started to bark].

The second problem arises from the fact that in BulTreeBank we accepted that all the dependents within and NP will be viewed as modifiers (see for the same decision in Butt et. al 1999: 46). However, in BURGER a hybrid approach has been taken – the relational nouns as well as the subject

[2] Only some phrases of type NP NP could need manual determination of the head.

counterpart in the frame of a deverbal noun are analyzed as complements. Thus, BulTreeBank contains analyses compatible with BURGER analyses which should be rejected by the ideology behind BURGER. This problem needs also smoothing of the BulTreeBank Schema for this particular task.

## 7 Conclusion and Future Work

In this paper, a method was presented for transferring of linguistic knowledge between two treebanks of Bulgarian, constructed within the same linguistic theory, but in its different versions and from different perspectives. BulTreeBank follows HPSG94 and the sentences have been analyzed per se. The target Treebank BURGER is more or less conformant to (Sag, Wasow and Bender 2003). It is being produced by an LKB-based Matrix grammar for Bulgarian, and has to discriminate among the overgenerated analyses. The linguistic information in BTB and BURGER is presented in the format of lexical categories and dependency relations. The actual categories and relations are HPSG generated on the basis of constituent labels in the corresponding analyses. A set of transferring rules on the level of the categories (or list of categories) is defined to ensure the compatibility of the representations. Currently our goal is to provide a mechanism for the usage of the linguistic knowledge encoded in BulTreeBank as a set of discriminating properties for the selection of the correct analyses produced by BURGER. Our current experiment proves the feasibility of this approach. We plan to extend the linguistic transfer with respect to higher coverage of data.

## 8 Acknowledgements

The work reported in the paper is supported by EuroMatrixPlus Project (http://www.euromatrixplus.eu/). We would like to thank Dan Flickinger for his support during the implementation of BURGER and to the reviewers for their comments on the paper.

## References

Emily Bender, Dan Flickinger and Stephan Oepen. 2002. *The Grammar Matrix: An Open-Source Starter-Kit for the Rapid Development of Cross-Linguistically Consistent Broad-Coverage Precision Grammars*. Procedings of the Workshop on Grammar Engineering and Evaluation. 8-14.

Miriam Butt, Tracy H. King, M.-E. Nino & F. Segond 1999: *A Grammar Writer's Cookbook.* CSLI.

Atanas Chanev, Kiril Simov, Petya Osenova and Svetoslav Marinov. 2009. *The BulTreeBank: Parsing and Conversion*. In: Nicolov, Angelova, and Mitkov (Eds.). Recent Advances in Natural Language Processing V: Selected papers from RANLP 2007. Vol. 309 in the series "Current Issues in Linguistic Theory", John Benjamins Publ., Amsterdam. 321-330.

Bart Cramer and Yi Zhang. 2009. *Construction of a German HPSG grammar from a detailed Treebank.* Proceedings of ACL-IJCNLP, pp. 37-45.

Michael Daum, Kilian Foth, and Wolfgang Menzel. 2004. *Automatic transformation of phrase treebanks to dependency trees*. In Proceedings of the 4th Int. Conf. on Language Resources and Evaluation, LREC-2004, Lisbon, Portugal. 99-106.

Dan Flickinger. 2000. *On building a more efficient grammar by exploiting types*. Natural Language Engineering, 6 (1) (Special Issue on Efficient Processing with HPSG), 15 – 28.

Julia Hockenmaier. 2006. *Creating a CCGbank and a Wide-Coverage CCG Lexicon for German*. Proceedings of ACL2006. pp. 505–512.

Sandra Kübler, Wolfgang Maier, Ines Rehbein, Yannick Versley. 2008. *How to Compare Treebanks*. Proceedings of LREC 2008.

Stephan Oepen and Ulrich Callmeier. 2000. *Measure for measure: Parser cross-fertilization. Towards increased component comparability and exchange*. In Proceedings of the 6th International Workshop on Parsing Technologies, Trento, Italy. pp. 183 – 194.

Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. 2002a. *The LinGO Redwoods Treebank: Motivation and Preliminary Applications*. Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002).

Stephan Oepen, Dan Flickinger, Kristina Toutanova, and Christoper D. Manning. 2002b. *LinGO Redwoods. A Rich and Dynamic Treebank for HPSG*. In Proceedings of The First Workshop on Treebanks and Linguistic Theories (TLT 2002), Sozopol, Bulgaria.

Petya Osenova and Kiril Simov. 2007. *Formal Grammar of Bulgarian*. IPP, BAS. Bulgaria. In Bulgarian.

Petya Osenova. 2010. *BURGER – Bulgarian Resource Grammar – Efficient and Robust.* Technical Report.

Carl Pollard and Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar*. Chicago University Press and CSLI Publications.

Ivan A. Sag, Thomas Wasow, Emily Bender. 2003. *Syntactic Theory: a formal introduction*. Second Edition. Chicago: University of Chicago Press.

Kiril Simov, Petya Osenova, Alexander Simov, and Milen Kouylekov. 2004. *Design and implementation of the Bulgarian HPSG-based treebank*. In: Journal of Research on Language and Computation, Vol. 2, Num. 4.

Kiril Simov. *HPSG-based annotation scheme for corpora development and parsing evaluation*. In: Nicolov, Botcheva, Angelova and Mitkov (eds), Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003, John Benjamins, Amsterdam/Philadelphia, Current Issues in Linguistic Theory (CILT), volume 260, 2004. 327-336.

# Evaluating Dependency Parsing Performance on German Learner Language

**Niels Ott and Ramon Ziai**

Collaborative Research Center 833, Project A4
University of Tübingen
E-mail: {nott,rziai}@sfs.uni-tuebingen.de

**Abstract**

We present an experiment on dependency parsing of German learner language. Ultimately aiming at evaluating the meaning of learner answers to German reading comprehension questions, we are interested in how reliable a parser trained on native language can identify the main argument relations. To that end, we manually annotated a small set of learner answers and parsed it using MaltParser (Nivre et al., 2007) trained on TüBa-D/Z (Telljohann et al., 2004). The evaluation of the results shows that semantically salient relations such as `SUBJ` and `OBJ` can generally be found reliably. Qualitative analysis indicates that the omission of syntactically central material, such as the finite verb, yields incorrect parses while other errors, e.g. in agreement or word order, can still be parsed robustly.

## 1 Introduction

In this paper, we present a pilot study about dependency-parsing German learner language with MaltParser (Nivre et al., 2007) trained on the TüBa-D/Z treebank (Telljohann et al., 2004). The context of this pilot study is an on-going research project on meaning comparison in realistic situations. Building on Bailey & Meurers (2008), we are exploring ways of evaluating student answers in reading comprehension tasks with respect to both the reading comprehension questions and the target answers given by language teachers.

Automatic dependency analysis of learner language is one component out of many in such a content assessment system. In this paper, we focus on this type of analysis as a separate subject of investigation, yet with an emphasis on our research intentions in the overall project. Unlike other projects involving learner language, we are not investigating form errors or L2 development. Our interest lies in the machinery required to perform robust analyses, supporting the creation of semantic representations, on several levels of complexity, given that the input often is not well-formed language.

For the pilot study, we worked on a snapshot of 106 learner answers from a task-based learner corpus which is currently being compiled within the project (Meurers, Ott & Ziai, 2010). This snapshot was annotated by three independent annotators using the dependency grammar scheme devised by Foth (2006). The procedure of selecting and annotating data as well as the peculiarities of annotating learner language are described in detail in section 3 after section 2 presents related work.

Since the corpus used in this study is not very large, we do not only look at quantitative evaluation measures: section 4 deals with the parsing procedure and also contains a qualitative analysis of selected issues arising in the automatic dependency analysis of learner language. We also take a look at automatic part-of-speech tagging and its influence on parser performance.

## 2 Related Work

Parsing learner language is not a novelty. Most notably, learner language is automatically analyzed in Intelligent Language Tutoring Systems, such as *e-tutor* (Heift & Nicholson, 2001), *Robo-Sensei* (Nagata, 2009) and *TAGARELA* (Amaral, 2007; Amaral & Meurers, 2011). Here, parsing is employed with the ultimate aim of giving feedback to students, mainly on form errors. Therefore, the parsing strategy needs to account for and explicitly model learner errors in a way that allows error detection and feedback (see e.g. Menzel & Schröder 1999).

More recently, there is some research on devising syntactic annotation schemes specific to language acquisition in progress, so-called *interlanguage*. Dickinson & Ragheb (2009) present such an approach, analyzing and annotating the interlanguage of English language learners. Most closely related to our work is the experiment presented by Dickinson & Lee (2009), where corpus annotation has been altered to support training of a dependency parser capable of handling a limited range of learner errors, namely postpositional particles in Korean.

However, in this paper, we are not concerned with giving feedback to learners or accurately describing interlanguage. Instead, we want to robustly parse learner language in order to access its content, and subsequently compare such content against that of reference answers. Therefore, due to our different motivation, we instead investigate how a parser trained on native language behaves when confronted with learner language. To our knowledge, such an experiment has not been done for German with a hand-annotated gold standard.

## 3 Annotation of Learner Answers

In this section, we describe the creation of a small corpus of learner answers to reading comprehension questions and its manual annotation with dependency grammar analyses. This small corpus serves as the gold standard for parser evaluation in the experiment described in the presented paper.

### 3.1 Data Source

The data used in the presented experiment and in our research project is collected in large German programs in the US, at Kansas University (Prof. Nina Vyatkina) and The Ohio State University (Prof. Kathryn Corl). Using the WEb-based Learner COrpus MachinE (WELCOME), a tool that has been developed especially for this purpose, German teachers in the two programs are collecting reading comprehension exercises consisting of texts, questions, target answers, and corresponding student answers (Meurers, Ott & Ziai, 2010). Each student answer is transcribed from the hand-written submission by two independent annotators. These two annotators also assess the contents of the answers purely on the basis of meaning: Did the student answer convey the meaning that was required by the question?

The corpus emerging from our on-going four-year collection phase is steadily growing. For this paper, we took a snapshot and selected learner answers according to the following criteria: a) full agreement in meaning assessment, b) edit distance between the two transcriptions of handwriting at most 1, c) minimum length of five tokens. Inevitably, answers by different students to one and the same reading comprehension question are quite similar. In order to ensure variety in our data, we randomly selected only one student answer for each question after applying the constraints a)–c).

The resulting subcorpus consists of 106 learner answers containing 109 sentences from the beginner and intermediate levels of the respective German programs. The average sentence length is 8.26 tokens with a standard deviation of 3.11. The shortest sentence contains 2 tokens, the longest 17. Tokenization and sentence segmentation were performed automatically using the OpenNLP[1] components and their default statistical models.

### 3.2 The Annotation Process

As far as the choice of dependency annotation scheme is concerned, we first looked at the ones employed at the CoNLL-X shared task on dependency parsing (Buchholz & Marsi, 2006). This task used a version of the TIGER treebank (Brants et al., 2002) converted to dependency grammar. However, we quickly found it difficult to annotate our data using this scheme because it is based on the phrase structure backbone of the TIGER annotation scheme (Albert et al., 2003). Thus, constructing a dependency analysis with the TIGER scheme manually would have required us to produce a phrase structure-based analysis first, which was not our intention. In contrast, Foth (2006) provides a readily available scheme and manual for German dependency grammar that we found convenient to use for the task of manual annotation.

Consequently, we used a dependency grammar version of the Tüba-D/Z treebank (Telljohann et al., 2004), because using Versley (2005)'s approach, it can be converted to the scheme devised by Foth (2006). However, relying only on

[1] http://opennlp.sourceforge.net

Foth's dependency grammar model in the annotation process would have mixed two variables in the parser evaluation step: the difference between Foth (2006) and the auto-converted Tüba-D/Z annotation (Versley, 2005), and the actual performance of the parser trained on Tüba-D/Z but parsing learner language. To minimize this problematic effect, annotators were advised to use the auto-converted Tüba-D/Z as the definite resource in case Foth (2006) was unclear or incomplete.[2]

The 109 sentences of our subcorpus were annotated by three independent annotators using DgAnnotator[3]. The percentage agreement between the three annotators computed to 88.1% for labeled attachment. However, we found only 67 sentences for which at least two annotators had fully agreed on the dependency annotation (both relation labeling and attachment). For the remaining 42 sentences, two of the annotators independently examined the differences in the annotations, writing down comments on the most salient issues having caused the deviation in the annotations. The third annotator then served as a judge fixing the annotations of these 42 sentences according to the comments.

Part-of-speech (POS) annotation was conducted automatically using Tree Tagger (Schmid, 1994) equipped with the standard model for German. This standard model uses the Stuttgart Tübingen Tagset (STTS, Thielen et al. 1999). Since the STTS variant used in Tüba-D/Z is slightly different, we converted the tagger output to fit the POS annotation that the parser had seen during training. Annotators were advised to manually correct the annotation on the POS layer in the case of tagging errors. The POS layer was double-checked by the abovementioned third annotator for correctness and consistency. As a result of this step, 7.2% of all POS labels in our subcorpus were manually post-corrected.

The context of the reading comprehension exercises consisting of reading texts, questions, and target answers was not taken into account in the annotation process. Similarly to the parser, the annotators did not have any additional sources of context knowledge for interpreting the learner answers.

### 3.3 Specific Issues in Annotating Learner Language

A popular theme in second language acquisition (SLA) research is error tagging of learner corpora (see e.g. Granger et al. 2009). Error tagging refers to the process of annotating defects in learner language with regard to well-formed versions of utterances, also called target hypotheses (Lüdeling et al., 2005). Target hypothesis are highly subjective: Lüdeling (2008) found that in an essay written by an advanced learner of German, there was not a single one out of 17 sentences for which all of her five professionally trained German teachers agreed on one single target hypothesis.

Dickinson & Ragheb (2009) propose a scheme for describing the learner's interlanguage instead of the L2 that the learner is about to acquire. In contrast, we admittedly use native language categories for annotating learner language. There are several reasons for following this route: first, a practical reason is that annotated

[2] More precisely speaking, annotators used only the test set as a resource. See also section 4.

[3] `http://medialab.di.unipi.it/Project/QA/Parser/DgAnnotator/`

*Learner sentence:*

‘Sie dachte es, welche das schönste Puppenwagen, den sie je gesehen hatte.’

*Target hypothesis 1:*

Sie dachte es, welcher der schönste Puppenwagen [ist], den sie je gesehen hatte
She thought it, which the finest doll’s pram [is] that she ever seen had

*Target hypothesis 2:*

Sie dachte, es wäre der schönste Puppenwagen, den sie je gesehen hatte
She thought, it was the finest doll’s pram, that she ever seen had

Figure 1: Differing target hypotheses due to learner errors.

corpora for training a parser need to be large and they are only available for native language at the moment. Second, for the purposes of our project we need to compare learner answers with target answers, questions, and reading texts written by native or near-native speakers. However, an interlanguage category system would not necessarily be applicable to native text. And third, interlanguages are specific to both the learner’s background and his stage of acquisition, and thus inherently difficult to capture in a single annotation scheme or parsing model.

For a number of sentences in our small corpus, we found it impossible to annotate them without constructing a target hypothesis. Among the 42 problematic sentences mentioned earlier, there were six cases in which the two commenters agreeingly attributed the deviation in the dependency annotation to differing target hypotheses. One such case is presented in Figure 1, where several learner errors make interpretation difficult.

For further research we therefore propose to follow the route suggested by Lüdeling (2008): target hypotheses should be explicitly annotated in the corpus. However, since we do not make use of them in our automatic dependency analysis, other users of the data would be the ones to benefit more from such an annotation.

# 4 Parsing Results

## 4.1 Setup

We used MaltParser (Nivre et al., 2007) for our parsing experiments. As previously mentioned, we applied the conversion procedure presented by Versley (2005) to release 5 of the TüBa-D/Z treebank (Telljohann et al., 2004) in order to get a training data basis that uses the Foth (2006) annotation scheme. We used 90% (40676 sentences) of the TüBa-D/Z sentences for training the parser, leaving out 10% (4520 sentences) to be used as a native language evaluation set, a point of reference for our experiments with learner language. The parameters used for training MaltParser where taken from Gómez-Rodríguez & Nivre (2010), using MaltParser’s new 2-planar algorithm which is capable of handling some non-projectivity. The resulting

|  | Learner LAS | TüBa LAS | Learner UAS | TüBa UAS |
|---|---|---|---|---|
| Automatic POS | 79.15% | 83.12% | 84.81% | 86.38% |
| Manual POS | **85.71%** | 88.07% | **90.22%** | 90.50% |

Table 1: Overall attachment scores for learner data set and TüBa-D/Z test set (LAS: labeled attachment score, UAS: unlabeled attachment score).

model achieves 88.07% labeled attachment score (LAS) on our TüBa-D/Z test set, which is state-of-the art.

Using this model, we parsed the hand-annotated learner data set presented in section 3. In order to be able to assess the contribution of proper POS annotation, we ran the parser on both automatically tagged and manually corrected versions of the data set. However, since a performance difference could also be due to the fact that automatic POS tagging is simply always inferior in quality to manual inspection, and not a result of learner language specifics, we made the same distinction with the TüBa-D/Z test set.

### 4.2 Quantitative Evaluation

The quantitative results are summarized in Table 1. All figures were obtained using the *eval.pl* script from the CoNLL-X shared task on dependency parsing (Buchholz & Marsi, 2006), which does not depend on any particular dependency scheme. Naturally, quantitative results on a small data set like the one we annotated should be interpreted with some caution as far as representativeness is concerned. However, it seems that the parsing results on our learner data set are within acceptable range ($\approx$ 3%) of what is currently considered to be state-of-the art in dependency parsing. Interestingly, automatic POS annotation causes a similar performance drop in both data sets. This indicates that there is no clear difference in the parsing contribution of higher quality POS between learner and native language. Note also the relatively high difference between labeled and unlabeled attachment scores for the learner data set, suggesting that the gap between knowing where to attach a word and how to label the attachment is wider for learner language, which seems plausible given the ungrammatical constructions learners sometimes use.

We also looked at parsing performance on individual dependency relation types. Recall that our ultimate goal is to obtain a representation of learner utterances suitable for semantic comparison with a reference answer. For such a representation, functor-argument relations such as `SUBJ` and `OBJA` thus seem particularly important, because they specify the main verb's arguments. Following Foth (2006, p. 6)'s distinction of argument from modifier relations, we summarized the ones which were interesting to us and had more than 10 gold standard instances in Table 2. The figures represent combined scores for both correct dependency labels and correct attachments. (An overview of the dependency labels used in this paper is given in Table 3 in the appendix.)

|  | Learner Data Set |  |  | TüBa-D/Z Test Set |  |  |
|---|---|---|---|---|---|---|
|  | Recall | Precision | Count | Recall | Precision | Count |
|  | Argument relations |  |  |  |  |  |
| DET | 99.02% | 97.12% | 102 | 99.15% | 99.51% | 9272 |
| SUBJ | 90.53% | 88.66% | 95 | 88.10% | 91.37% | 6043 |
| OBJA | 84.31% | 84.31% | 51 | 81.77% | 77.70% | 2979 |
| CJ | 87.18% | 89.47% | 39 | 93.91% | 94.03% | 2514 |
| PRED | 65.38% | 85.00% | 26 | 75.80% | 80.38% | 1157 |
| AUX | 100.00% | 95.83% | 23 | 93.93% | 97.03% | 2503 |
| OBJP | 36.36% | 80.00% | 11 | 55.92% | 69.13% | 769 |
|  | Modifier relations |  |  |  |  |  |
| KON | 63.27% | 65.96% | 49 | 81.22% | 79.09% | 3003 |
| ADV | 68.18% | 73.17% | 44 | 83.76% | 84.38% | 5770 |
| PP | 80.00% | 55.81% | 30 | 75.25% | 75.07% | 6302 |

Table 2: Precision and recall for selected dependency types including attachment. The figures are based on the parser output based on gold POS tags and the *Count* columns represent the number of occurrences in the respective gold standards.

According to the figures we obtained, subject and (accusative) object relations can be found approximately as reliably as in native language text, with precision and recall around 90% for subjects and 84% for accusative objects. Predicates are missed more often, with a recall of just 65.38%, but tend to be reliable if identified. Prepositional complements are often analyzed as adjuncts, indicated by the low recall for `OBJP` (prepositional object) and relatively low precision for `PP` (prepositional adjunct). However, this distinction was also difficult for human annotators, since it is not always obvious whether a prepositional element is an obligatory verb argument or not.

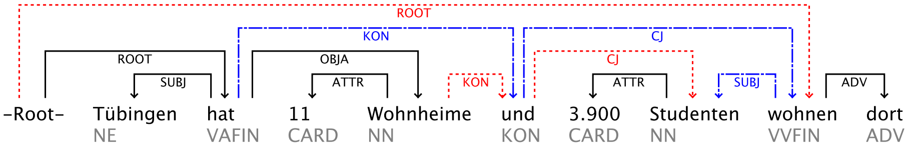
### 4.3 Qualitative Evaluation

We also qualitatively evaluated our parsing results, partly because our data set is small and thus quantitative means of evaluation are of limited usefulness, and also because it is instructive to closely inspect the problems a parser encounters when confronted with learner data. In the following figures[4], dash-dotted arcs represent the gold standard whereas dotted red arcs represent the parser's decisions. Solid lined arcs represent correctly analyzed structures. Where necessary for explanation, we included target hypotheses although as mentioned in section 3, we did not explicitly construct these during annotation.

[4] We used *What's wrong with my NLP?* for generating the figures, see `http://code.google.com/p/whatswrong`

### Common Dependency Parsing Problems

Unsurprisingly, some of the issues are already well-known from parsing native language. Most notably, coordination is problematic and occurs quite frequently which, given the prompts for lists of items or propositions students are faced with, is to be expected. Consider the sentence in Figure 2.



‘Tübingen has 11 dormitories and 3.900 students live there’

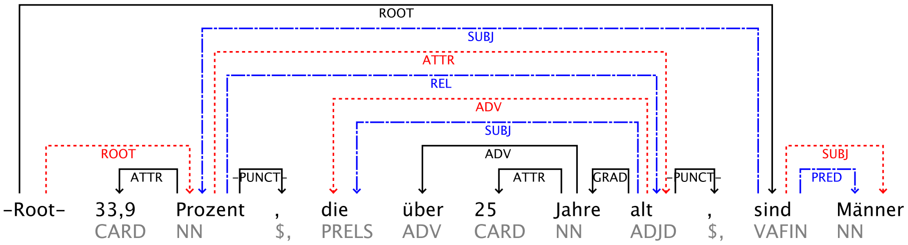
Figure 2: Sentence-level coordination wrongly analyzed.

The sentence is a perfectly well-formed example of sentence-level coordination, however the parser seems to have problems attaching conjuncts over longer distances. This might be due to the fact that the feature models underlying most parsers, including the one used here, only look a few tokens behind or ahead.

### Learner Errors Not Handled Well

Of course there are also examples where the very nature of learner language made a correct dependency analysis in terms of L2 impossible. Usually these were the cases where also the human annotators were unsure of how to annotate a given sentence. Consider the sentence in Figure 3, where the learner left out the finite verb of a relative clause.

In the given target hypothesis of this sentence, the finite verb in the relative clause would have been the link to the main clause. However, since that verb is missing, the parser used the adjective *alt* (‘old’) for this purpose. Incidentally, so did the human annotators, albeit with different labels for the relations involving *alt*.



*Target hypothesis:*

33,9 Prozent, die über 25 Jahre alt [sind], sind Männer.
33.9 percent, who over 25 years old [are], are men.

Figure 3: Dependency errors due to missing verb in relative clause.

### Learner Errors Handled Well

On the bright side, some learner errors also seem to be handled well. Specifically, local problems such as wrong verb forms or word order are usually handled correctly. The sentence in Figure 4 is such an example.

Here, the learner apparently overused participles, namely *gekramt* instead of preterite *kramte* ('rummaged') and *aufgefunden* instead of infinitive *finden* ('to find'). Moreover, *wollte* ('wanted') ought to be at the end of the subordinate clause, not next to the subject *sie* ('she'). However, all of these errors are analyzed robustly and the parser comes up with the analysis proposed by the human annotators.

## 5 Conclusion

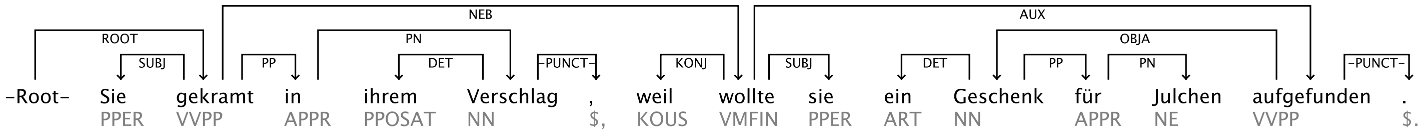
We presented a parsing experiment on German learner language using a state-of-the art dependency parser trained on native language. In order to evaluate the result, we hand-annotated a sample of the German learner corpus we are currently collecting. The results are generally promising, showing that the main functor-argument relation types we are most interested in can generally be identified reliably, with precision and recall in the area of 80–90%. Furthermore, qualitative evaluation shows that while learner errors that result in the omission of syntactically central material lead to serious problems in dependency analysis, other errors are often handled well. The latter include agreement errors, verb tense errors and word order errors.

Future work should include the annotation of more data from several levels of difficulty in order to obtain a wider range of linguistic phenomena, making a quantitative evaluation more meaningful. An interesting addition to explore would be to also annotate the learner sentences with well-formedness judgments, enabling us to find out whether there is a significant correlation between ill-formedness and parsing errors. The annotation of target hypotheses could make the implicit assumptions underlying the dependency analysis explicit in the future, which would facilitate the interpretation of the annotation.

Concerning our intent of evaluating the meaning of learner answers, we are now more confident that a dependency analysis of German learner language based on an L2 parser model is a viable method, provided one is "careful" with the interpretation of the parse and knows what kind of input will potentially result in bad analyses such as the one in Figure 3. Going forward, we are planning to investigate methods of transforming dependency trees into semantically more useful representations, possibly in the spirit of the 'compositional facets' employed by Nielsen et al. (2009) in the context of Intelligent Tutoring Systems.

### Acknowledgements

We wish to thank Detmar Meurers and Holger Wunsch for their valuable comments. We are also thankful to Janina Kopp who was of great assistance in the annotation process. Finally, we also benefited from three anonymous workshop reviews.



Target hypothesis:

Sie kramte in ihrem Verschlag, weil sie ein Geschenk für Julchen finden wollte.
She rummaged in her shed, because she a gift for Julchen to-find wanted.

'She rummaged in her shed because she wanted to find a gift for Julchen.'

Figure 4: Robust analysis of ill-formed sentence.

# References

Albert, S., J. Anderssen et al. (2003). *TIGER Annotationsschema*. Universität des Saarlandes and Universität Stuttgart and Universität Potsdam.

Amaral, L. (2007). Designing Intelligent Language Tutoring Systems: integrating Natural Language Processing technology into foreign language teaching. Ph.D. thesis, The Ohio State University.

Amaral, L. & D. Meurers (2011). On Using Intelligent Computer-Assisted Language Learning in Real-Life Foreign Language Teaching and Learning. *ReCALL* 23(1).

Bailey, S. & D. Meurers (2008). Diagnosing meaning errors in short answers to reading comprehension questions. In J. Tetreault, J. Burstein & R. D. Felice (eds.), *Proceedings of the 3rd Workshop on Innovative Use of NLP for Building Educational Applications (BEA-3) at ACL'08*. Columbus, Ohio, pp. 107–115.

Brants, S., S. Dipper, S. Hansen, W. Lezius & G. Smith (2002). The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*. Sozopol.

Buchholz, S. & E. Marsi (2006). CoNLL-X shared task on multilingual dependency parsing. In *CoNLL-X '06: Proceedings of the Tenth Conference on Computational Natural Language Learning*. Morristown, NJ, USA: Association for Computational Linguistics, pp. 149–164.

Dickinson, M. & C. M. Lee (2009). Modifying Corpus Annotation to Support the Analysis of Learner Language. *CALICO Journal* 26(3), 545–561.

Dickinson, M. & M. Ragheb (2009). Dependency Annotation for Learner Corpora. In *Proceedings of the Eighth Workshop on Treebanks and Linguistic Theories (TLT-8)*. Milan, Italy.

Foth, K. (2006). *Eine umfassende Constraint-Dependenz-Grammatik des Deutschen*. Tech. rep., Universität Hamburg.

Gómez-Rodríguez, C. & J. Nivre (2010). A Transition-Based Parser for 2-Planar Dependency Structures. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. pp. 1492–1501.

Granger, S., E. Dagneaux, F. Meunier & M. Paquot (2009). *International Corpus of Learner English Version 2*. Presses Universitaires de Louvain.

Heift, T. & D. Nicholson (2001). Web Delivery of Adaptive and Interactive Language Tutoring. *International Journal of Artificial Intelligence in Education* 12(4), 310–325.

Lüdeling, A. (2008). Mehrdeutigkeiten und Kategorisierung: Probleme bei der Annotation von Lernerkorpora. In P. Grommes & M. Walter (eds.), *Fortgeschrittene Lernervarietäten*, Tübingen: Niemeyer, pp. 119–140.

Lüdeling, A., M. Walter, E. Kroymann & P. Adolphs (2005). Multi-level error annotation in learner corpora. In *Proceedings of Corpus Linguistics*. Birmingham.

Menzel, W. & I. Schröder (1999). Error diagnosis for language learning systems.

*ReCALL* Special ed., 20–30.

Meurers, D., N. Ott & R. Ziai (2010). Compiling a Task-Based Corpus for the Analysis of Learner Language in Context. In *Proceedings of Linguistic Evidence*. Tübingen, pp. 214–217.

Nagata, N. (2009). Robo-Sensei's NLP-Based Error Detection and Feedback Generation. *CALICO Journal* 26(3), 562–579.

Nielsen, R. D., W. Ward & J. H. Martin (2009). Recognizing entailment in intelligent tutoring systems. *Natural Language Engineering* 15(4), 479–501.

Nivre, J., J. Nilsson, J. Hall, A. Chanev, G. Eryigit, S. Kübler, S. Marinov & E. Marsi (2007). MaltParser: A Language-Independent System for Data-Driven Dependency Parsing. *Natural Language Engineering* 13(1), 1–41.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*. Manchester, UK, pp. 44–49.

Telljohann, H., E. Hinrichs & S. Kübler (2004). The TüBa-D/Z Treebank: Annotating German with a Context-Free Backbone. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*. Lissabon.

Thielen, C., A. Schiller, S. Teufel & C. Stöckert (1999). *Guidelines für das Tagging deutscher Textkorpora mit STTS*. Tech. rep., Institut für Maschinelle Sprachverarbeitung Stuttgart and Seminar für Sprachwissenschaft Tübingen.

Versley, Y. (2005). Parser Evaluation across Text Types. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT)*. Barcelona, Spain.

## Dependency Labels

| Label | Short description | Label | Short description |
|---|---|---|---|
| ADV | adverbial modifier | OBJA | accusative object |
| ATTR | nominal attribute | OBJP | prepositional object |
| AUX | auxiliary verb | PN | preposition complement |
| CJ | complement of a conjunction | PP | prepositional adjunct |
| DET | determiner of a noun | PRED | predicate |
| GRAD | accusative NP as measuring unit | -PUNCT- | punctuation |
| KON | non-final coordination conjunct | REL | relative clause |
| KONJ | subordinating conjunction | ROOT | root of the sentence |
| NEB | subordinate clause | SUBJ | subject |

Table 3: Dependency labels used in this paper, taken from Foth (2006).

# Semantic Annotation of Deverbal Nominalizations in the Spanish AnCora corpus

| Aina Peris | Mariona Taulé | Horaci Rodríguez |
|---|---|---|
| CLiC- UB | CLiC- UB | TALP -UPC |
| Gran Via,585 | Gran Via,585 | Jordi Girona Salgado 1-3 |
| 08007 Barcelona | 08007 Barcelona | 08034 Barcelona |
| aina.peris@ub.edu | mtaule@ub.edu | horacio@lsi.upc.edu |

## Abstract

This paper presents the methodology and the linguistic criteria followed to enrich the AnCora-Es corpus with the semantic annotation of deverbal nominalizations. The first step was to run two independent automated processes: one for the annotation of denotation types and another one for the annotation of argument structure. Secondly, we manually checked both types of information and measured inter-annotator agreement. The result is the Spanish AnCora-Es corpus enriched with the semantic annotation of deverbal nominalizations. As far as we know, this is the first Spanish corpus annotated with this type of information.

## 1. Introduction

In recent years semantically annotated corpora have been made available to the research community: PropBank (Palmer et al., 2005) for English, the TIGER corpus (Brants et al., 2002) for German, the PragueTreebank (Bohmova et al., 2001) for Czech, and the AnCora corpora (Taulé et al., 2008) for Catalan and Spanish. Most corpora focus on the argument structure of verbs, but some also annotate deverbal nominalizations because, like verbs, they also contain rich semantic information. However, they have until now been represented only in English corpora such as NomBank (Meyers, 2007)[1]. This paper presents the methodology and the linguistic criteria followed to annotate deverbal nominalizations in the Spanish AnCora-Es corpus[2]. The main goal achieved is the enrichment of AnCora-Es with the annotation of denotation types (i.e., result, event, and underspecified) and the argument structure of deverbal nominalizations. Identifying this information can be very useful for many tasks and applications of Natural Language Processing (NLP). In addition, such a corpus can provide real evidence for the linguistic study of nominalizations.

[1] Nominalizations are also represented in the different FrameNet projects, which are lexical databases supported by corpus evidence (Boas, 2009).

[2] AnCora-Es is a 500,000-word corpus annotated at different linguistic levels: from morphology to pragmatics (coreference): http://clic.ub.edu/corpus/ancora.

The remainder of the paper is organized as follows. Sections 2 and 3 describe the annotation of denotation types and of argument structures, respectively. Both sections include a description of the inter-annotator agreement tests. Finally, main conclusions are drawn in Section 4.

## 2. Denotation Types

A relevant feature of deverbal nominalizations is their denotation, i.e., the semantic interpretation they are associated with. We focused on this information because it is one of the most controversial and studied topics in the literature, and it is not a straightforward distinction. In the linguistic literature (Grimshaw, 1990; Picallo, 1999; Alexiadou, 2001), nominalizations are said to have basically two semantic denotations: nominalizations denoting an event (1a)—i.e., they refer to an action,—or a result (1b)—i.e., they refer to the result of an action. However, in a previous study (Peris and Taulé, 2009), we observed that these two denotations do not allow us to account for the data in the corpus. First, it is not always possible to distinguish between event and result, since the linguistic context is sometimes not informative enough. We label such cases as underspecified types (1c), resulting finally in three possible denotation values. Second, we noticed that nominalizations can take part in a lexicalized construction, thus, we added the attribute <lexicalized>. One of the three above-mentioned denotation values is assigned to the whole lexicalized construction only in the case of nominal lexicalizations (1d)[3].

(1a) [La **reconstrucción**$_{\text{<event>}}$ de la ciudad por los chinos]$_{\text{NP}}$ tiene lugar en estos momentos.[4]
‘[The **reconstruction** of the city by the Chinese] is being carried out at the moment.’
(1b) No espere [una **definición**$_{\text{< result>}}$ de la palabra cultura de María]$_{\text{NP}}$.
‘Do not expect [a **definition** of the word culture from María].’
(1c) Se espera [la **llegada**$_{\text{< underspecified>}}$ de 450 observadores extranjeros]$_{\text{NP}}$.
‘[The **arrival** of 450 foreign observers] is expected.’
(1d) Se habla de [un **golpe de Estado]** $_{\text{< lexicalized=“yes”> < result>}}$ de manera irresponsible.
‘[A **coup d’état**] is being talked about in an irresponsible way.’

### 2.1 Methodology

The annotation of denotation types consists in associating a type to each deverbal noun. By *deverbal noun* we mean a noun morphologically derived

[3] We distinguish six types of lexicalization containing nominalizations according to their similarity to different word classes: nominal, prepositional, verbal, adjectival, adverbial and conjunctive lexicalizations.

[4] All the examples are extracted from the AnCora-Es corpus.

from a verb or a so-called *cousin* noun (Meyers, 2007)[5]. Cousin nouns are nouns that give rise to a verb (e.g., *revolución* 'revolution'> *revolucionar* 'to revolutionize'), or nouns semantically related to a verb (e.g., *genocidio* 'genocide' can be related to *exterminar* 'to exterminate'). Deverbal nouns were previously selected from a list automatically obtained using a predefined set of ten suffixes[6] that take verbal stems and have an action-result meaning (Santiago and Bustos, 1999).

The annotation of denotation types was carried out in two steps. First, the annotation was automated by means of the ADN classifier (Peris et al., 2010), which uses a machine-learning approach taking as features most of the information that appears in the current guidelines (see Section 2.3), obtaining 80, 6% F-measure. In a second step, the results were manually validated in order to ensure the accuracy of the annotation. Precisely, we annotated a total of 23,000 tokens belonging to 1,655 types of deverbal nominalizations.

## 2.2 Annotation Scheme

The attribute to represent the denotation value in the corpus is <denotationtype> and its possible values are: event, result and underspecified. This information is assigned to deverbal nouns together with the attribute <originlexicalid>, whose value is the base verb; thus, ensuring the connection with the corresponding verbal lexical entry of AnCora-Verb-Es (Aparicio et al., 2008). Lexicalized constructions are marked with the attribute <lexicalized> and the value 'yes'.

## 2.3 Linguistic Criteria for the Classification of Denotation Types

This section details the morphological, syntactic and semantic criteria for the classification of deverbal nominalizations into denotation types. Before applying these criteria, the annotators check whether the nominalization is part of a lexicalized construction. If so, they choose a lexicalization type. In (1d), *golpe de estado* 'coup d'état' is considered to be a lexicalized construction for three reasons. First, its reference changes in relation to the simple nominalization (*golpe* 'hit'); second, the second element (*estado* 'état') cannot take its own complements (e.g., *democrático* 'democratic') (1d'); and finally, the insertion of an element into the lexicalized construction is infelicitous (1d'').

[5] We understand that a deverbal nominalization is semantically related to a verb, regardless of whether it is morphologically derived from a verb (marked with the <cousin> attribute with a negative value) or not (marked with the <cousin> attribute with a positive value).

[6] The suffixes are: -a, -aje, -ión/-ción/-sión/-ón, -da/-do, -dura/-ura, -e, -ido, -miento/-mento, -ncia/-nza, -o/-eo.

(1d’) *Se habla de [un **golpe de Estado** democrático] de manera irresponsable
‘[A democratic **coup d’état**] is being talked about in an irresponsible way.’
(1d’’)*Se habla de [un **golpe de** gran **Estado**] de manera irresponsable
‘[A **coup de** large **état**] is being talked about in an irresponsible way.’

Once it has been decided whether the nominalization is part of a lexicalized construction, the denotation value is assigned according to the criteria that we present next. These criteria are not deterministic, they must be understood as indicators, the combination of which help us to decide the nominalization denotation.

**a) Incorporated Argument:** In Spanish the denotation type is result when the deverbal nominalization incorporates an internal argument from the corresponding base verb. For instance, *invento* ‘invention’ denotes the object resulting from the verbal action as well as the verbal action (2).

(2) [El **invento**Arg1<denotationtype= “result”> de Juan] tuvo mucho éxito.
‘ **[**John’s **invention**] had a lot of success.’

**b) Plurality:** An identifying criterion for result nominalizations proposed in the literature is their ability to take the plural inflection (3a), unlike event nominalizations (3b)[7].

(3a) Para compensar [las **pérdidas**<denotationtype=“result”> ante sus depredadores], los tities traen al mundo gemelos.
‘To compensate [the **losses** before their predators], monkeys bring twins to the world.’
(3b) […] aunque [la **pérdida**<denotationtype=“event”> del pívot Rodney_Dent] puede condenar a los de Rick_Pitino.
‘[…] although [the **loss** of the pivot Rodney_Dent] can condemn those of Rick_Pitino.’

**c) Determiners:** It is widely accepted that event nominalizations can only be introduced by the definite article and the possessive, and they can also appear without a determiner (4a); whereas, result nominalizations can also be introduced by other types of determiners such as demonstratives, indefinite articles and numerals (4b).

(4a) No fue un hecho aislado, sino [*la* **culminación**<event> de [una dinámica de deterioro de las instituciones por_parte_del PP]].

[7] It is also stated that some events can be pluralized. We are aware of this possibility, but our annotation experience has revealed that most plural nominalizations denote results.

‘It was not an isolated fact, but [*the* **culmination** of [a process of deterioration of the institutions on the part of the PP]].
(4b) Las exportaciones totales pasaron de 12,3 millones de dólares a 14,8 millones, lo que supone [*una* **subida**$_{\text{<result>}}$ del 20,47_por_ciento].
‘The total exports increased from 12.3 million dollars to 14.8 millions, which means [*an* **increase** of 20.47 per cent].’

**d) Complementation:** From the literature we learned that there are two types of nominalization complements that characterize a specific denotation type. First, relational adjectives are arguments of result nominalizations (5a), but not of event nominalizations (5b). Indeed, (5b) is ungrammatical because *producción* ‘production’ cannot be understood as an event: interpreting the relational adjective *quesera* ‘cheese’ as an Arg1 blocks this reading.

(5a) El tema de conversación era [la **actuación**$_{\text{<result>}}$ *policial*$_{\text{AP-arg0-agt}}$].
‘The topic of discussion was [the police **acting**].’
(5b) *[La **producción**$_{\text{<event>}}$ quesera$_{\text{AP-arg1-pat}}$ por los holandeses][8].
‘[The cheese **production** by the Dutch].’

Second, temporal adjuncts of result nominalizations must be introduced by the preposition *de* ‘of’ (6a), whereas this preposition is not needed for temporal adjuncts of event nominalizations (6b).

(6a) Hoy, tras [una **negociación**$_{\text{< result>}}$ *de trece horas* $_{\text{PP-argM-tmp}}$], se ha aprobado un nuevo texto sobre la reforma del seguro de desempleo.’
‘Today, after [a **negotiation** of thirteen hours], a new text has been approved on the reform of the unemployment insurance.’
(6b) La compañía presentó una auditoría por primera vez desde [su **constitución** $_{\text{< event>}}$ *en 1989* $_{\text{PP-argM-tmp}}$]
‘The company submitted a clean audit for the first time since [its **constitution** in 1989].’

**e) Verbal class:** Following Alexiadou (2001) and Picallo (1999) we have taken into account the semantic class of the verb from which the nominalization derives. This is very useful in order to decide the denotation type. Nominalizations are annotated with the verbal classes declared in the verbal lexicon AnCora-Verb. There are four general classes that are defined according to Vendler’s (1967) event classes: accomplishments, achievements, states and activities. Next, we briefly detail how they influence the annotation of denotation types.

[8] This example is not from the AnCora-Es corpus.

**Accomplishments:** Verbs belonging to this class can give rise to result, event and underspecified nominalizations. The reading of the nominalization depends on which verbal arguments are syntactically realized in the nominalized NP and by which constituents (Table 1).

| **Arg0** | **Arg1** | **Arg2** | **Denotation** |
|---|---|---|---|
| Not realized | Not realized | Not realized | Result |
| Not realized | Not realized | PP/GRP | Underspecified |
| Not realized | PP/GRP/Poss[9] | PP/GRP/Poss | Event |
| PP-*por*-Agent | PP/GRP/Poss | Not realized or PP/GRP/Poss | Event |
| PP-*de* Agent | PP/GRP/Poss | Not realized or PP/GRP/Poss | Result |
| PP-*de/por* Agent | PP/GRP/Poss | Not realized | Underspecified |
| PP-*de/por* Agent | PP/GRP/Poss | PP/GRP/Poss | Event |
| PP-*de/por* Cause | PP/GRP/Poss | Not realized | Underspecified |
| PP-*de/por* Cause | PP/GRP/Poss | PP/GRP/Poss | Underspecified |
| Any constituent | patient possessive | Any constituent | Event |

Table 1: Denotation types according to the argument realization for nominalizations derived from an accomplishment.

**Achievements**: Verbs belonging to the achievement class are realized in unaccusative structures (i.e., with no Arg0). Therefore, the denotation types of the corresponding nominalizations depend on the syntactic realization of Arg1 and Arg2 (see Table 2).

| **Arg1** | **Arg2** | **Denotation** |
|---|---|---|
| Not realized | Not realized | Result |
| Realized | Not realized | Underspecified |
| Not realized | Realized | Underspecified |
| Realized | Realized | Event |

Table 2: Denotation types according to the argument realization for nominalizations derived from an achievement.

**States**: Verbs included in this semantic class denote states and their corresponding nominalizations are always result nominalizations.

**Activities**: Predicates belonging to this class are unergative and their corresponding nominalizations can only have a result interpretation. Their subject can be explicit, but it is always introduced by the preposition *de* 'of'.

**e) Selectors:** When the above criteria do not lead to a clear denotation type, we found other indicators that can help select one, the so-called

[9] PP stands for prepositional phrase, GRP stands for genitive relative pronoun, and Poss stands for possessive determiner.

*selectors*. We distinguish two types of selectors: (i) external selectors, i.e., those elements that point to a specific denotation from outside the nominalized NP (7a); and (ii) internal selectors, i.e., prefixes within the nominalized NP that indicate a specific denotation type (7b).

(7a) <u>*Durante*</u> [la **presentación**$_{\text{< event>}}$ del libro $_{\text{CN-arg1-pat}}$], él abogó por la formación de los investigadores en innovación tecnológica.
'<u>During</u> [the **presentation** of the book], he advocated for the training of researchers in technological innovation.'
(7b) Hoy [la **reubicación**$_{\text{< event>}}$ del ex ministro $_{\text{CN-arg1-tem}}$] no resulta fácil.
'Today, [the **relocation** of the ex minister] does not seem easy.'

The preposition *durante* 'during' in (7a) gives a clue to interpret *presentación* 'presentation' as an event. In (7b), the nominalization *reubicación* 'reubication' with the prefix *re-* having a reiterative meaning must be of event type. This is due to the fact that the repetitive meaning only applies to bases that denote actions.

### 2.4 Inter-Annotator Agreement

In order to ensure the quality and reliability of the manual validation of denotation types, an inter-annotator agreement test was carried out. Five Linguistic graduate students participated in the test. Since none of them had experience in distinguishing between denotation types and this is not an easy semantic distinction, we built a training data set consisting of one hundred sentences. Each sentence contained a deverbal nominalization. For the real test we used a data set of two hundred sentences (i.e., two hundred deverbal nominalizations). For the purpose of annotation, the five annotators took into account the criteria presented in the previous section. They were required to work individually. We measured agreement using observed agreement (Scott, 1955) and the Kappa coefficient (Siegel and Castellan, 1988). Table 3 shows the average results of the inter-annotator agreement test.

| Average Pairwise Results | Training Set | Test Set |
|---|---|---|
| Observed agreement | 68% | 75% |
| Kappa | 44% | 60% |

Table 3: Inter-annotator agreement results for denotation types.

As expected, there is an improvement between the training and test data sets that is even more noticeable in the kappa measure (16% improvement). With regard to the result for the test data set, it can be said that the agreement level is not bad (60% Kappa, 75% observed agreement) given that the semantic distinction we are dealing with is very difficult[10]. Furthermore, this agreement score makes it possible to ensure the quality of the annotation since the annotators were allowed to seek advice from each other during the manual annotation process.

## 3. Argument Structure and Thematic Roles

The annotation of argument structures was based on the initial assumption that deverbal nominalizations inherit the argument structure of the base verb. Thus, the assignment of arguments and thematic roles was conducted taking into account the argument structures specified in the AnCora-Verb-Es lexicon. As we explained above, each noun is linked to the corresponding verbal lexical entry via the attribute <originlexicalid>. The annotation is limited to constituents inside the nominalized NP (8a) and also to arguments incorporated into the noun (8b)[11]. In the latter case, the noun was annotated with the attributes <arg> and <tem> and their values (see Section 3.2) during the manual validation process.

(8a) [El **impulso** *de la investigación*] es un punto clave para España.
'[The **promotion** *of research*] is a key point for Spain.'
(8b) [La ***propuesta*** de María] es muy buena.
'[María's ***proposal***] is very good.'

In Spanish, the constituents that can be arguments inside the NP are: PPs, relational adjective phrases (APs), GRPs, and possessive specifiers (spec-dp). Other types of constituents such as the rest of APs, NPs, adverbial phrases (ADVPs), or sentences (S) cannot be arguments inside the NP and receive the semantic label "RefMod" ("Reference Modifier"), which indicates that they modify the reference of the noun they are complementing.

### 3.1 Methodology

The annotation of argument structures consists in assigning the argument position (attribute <arg>) and the corresponding thematic role (attribute <tem>) to each argument in a nominalized NP. As in the case of the denotation type, an automated process was first applied, followed by a manual validation. A set of heuristic rules in a decision-list format was used for the automated annotation, obtaining 74% F-measure (Peris and Taulé, forthcoming).

[10] Standard guidelines describe a kappa over 75% as excellent, 40% to 75% as fair to good, and below 40% as poor (Fleiss, 1981). According to this, a 60% is a quite good agreement.

[11] The *proposal* is the thing proposed, so the Arg1-patient is incorporated into the noun.

### 3.2 Annotation Scheme

We use the same annotation scheme as the one followed to annotate the argument structure of verbs in AnCora-Es, which was in turn based on PropBank (Palmer et al., 2005) and VerbNet (Kingsbury, et al., 2002). In this way, we ensure the consistency of the annotation of arguments between the different predicates (nouns and verbs). The selected arguments are incrementally numbered expressing their degree of proximity in relation to their predicate ('arg0', 'arg1', 'arg2', 'arg3', 'arg4') and the adjuncts are labeled as ArgM. The list of thematic roles includes 19 different labels widely used in linguistics[12]. In a nutshell, we assign the attribute <arg> and <tem> to each nominal argument. The combination of the six arguments tags and the thematic roles tags results in a final tagset of 36 semantic tags.

### 3.3 Criteria

The annotators look up the verbal entries of the AnCora-Verb lexicon to assign arguments and thematic roles to the constituents of nominalized NPs. If a constituent can be interpreted according to an argument declared in the verbal entry, then it is annotated with this argument position <arg> and thematic role <tem>. If not, then there are two possibilities: (a) the constituent is interpreted as an ArgM that is not represented in the verbal entry, or (b) it has no argumental interpretation but is a noun modifier, which is labelled as "RefMod".

### 3.4 Inter-Annotator Agreement

We also wanted to ensure the quality and reliability of the manual annotation of arguments and thematic roles. To this end, the inter-annotator agreement test was conducted on a data set of one hundred sentences, each sentence containing a deverbal nominalization with at least one candidate to be an argumental constituent. A total of 131 constituents were included. Three Linguistic graduate students with previous experience in annotating verbal argument structure participated in the test. Due to their experience no training was needed. The test consisted in deciding, for each constituent, (a) whether it was an argument, and, if so, (b) which argument position and thematic role it should be assigned (out of a total of 36 possible tags). To this end, they had to take into account the information specified in the AnCora-Verb lexicon about the verbal sense from which they decided the deverbal nominalization

[12] These labels are: 'agt' (agent), 'cau' (cause), 'exp' (experiencer), 'scr' (source), 'pat' (patient), 'tem' (theme), 'atr' (attribute), 'ben' (beneficiary), 'ext' (extension), 'ins' (instrument), 'loc' (locative), 'tmp' (time), 'mnr' (manner), 'ori' (origin), 'des' (goal), 'fin' (purpose), 'ein' (initial state), 'efi' (final state) and 'adv' (adverbial).

came. This was important since we measured inter-annotator agreement taking into account whether the annotators agreed on the verbal class from which the nominalization derived. We measured inter-annotator agreement using observed agreement (Scott, 1955) and the Kappa coefficient (Siegel and Castellan, 1988). Disagreements on the argument and thematic role are expected when the verbal class taken as starting point is not the same since the arguments and thematic roles to be mapped vary. For this reason, we calculate observed and kappa measures increasing the penalization rate for disagreements when the annotators picked the same verbal class, and reducing the penalization rate for disagreements when the annotators picked different verbal classes. The weighting schema for measuring agreement was empirically set to 40% for the former, and to 60% for the latter.

Total agreement = (0.4*same VerbalClass) + (0.6*different VerbalClass)

Table 4 presents the results of the inter-annotator agreement test. The columns show the results for each pair of annotators and the average result. The rows present observed agreement (OA) and kappa coefficients according to the above formula.

| **ANNOTATOR PAIRS** | **A and B** | **A and C** | **B and C** | **AVERAGE RESULT** |
|---|---|---|---|---|
| Same verbal class | 119 | 125 | 125 | |
| OA | 86% | 96% | 90% | 90.6% |
| Kappa | 84% | 94% | 88% | 88.6% |
| Different verbal class | 12 | 6 | 6 | |
| OA | 66% | 66% | 83% | 71.6% |
| Kappa | 60% | 58% | 80% | 66% |
| **Total** | 131 | 131 | 131 | |
| OA | 74% | 78% | 85.8 % | 79.2% |
| Kappa | 69.6% | 72.4% | 83.2 % | 75% |

Table 4: Inter-annotator agreement results for argument structure and thematic roles.

We focus on the average result (last column). As expected, when the annotators did not agree on the verbal class, the agreement decreases approximately 20% both in observed (71.6%) and kappa (66%) agreement scores with respect to when the annotators agreed on the verbal class (90.6% and 88.6%, respectively). According to the above-presented measure, the mean of inter-annotator agreement reaches 75% kappa, which translates to 79.2% observed agreement. This is a satisfactory level of agreement given that there are 36 possible tag combinations, which largely increases the opportunities for disagreement. Furthermore, this agreement score allows us to provide for a reliable manual annotation process.

## 4. Conclusions and Future work

In this paper, we presented the methodology followed to annotate deverbal nominalizations in the AnCora-Es corpus. The first step was to run two independent automated processes: one for the annotation of denotation types and another one for the annotation of argument structures. Secondly, and the focus of this paper, we manually checked both types of information and measured inter-annotator agreement. The result is the Spanish AnCora-Es corpus enriched with the semantic annotation of deverbal nominalizations. As far as we know, this is the first Spanish corpus annotated with this type of information. Future work will focus on applying the presented methodology to annotate the nominalizations in the Catalan AnCora-Ca corpus.

## Acknowledgments

This research has received support from the projects Text-Knowledge 2.0 (TIN2009-13391-C04-04), AnCora-Net (FFI2009-06497-E/FILO) and KNOW2 (TIN2009-14715-C04-04) from the Spanish Ministry of Science and Innovation, and a FPU grant (AP2007-01028) from the Spanish Ministry of Education. We also thank three anonymous reviewers for their comments and suggestions.

## References

Alexiadou, A. (2001). *The Functional Structure in Nominals. Nominalization and Ergativity*. Amsterdam / Philadelphia: John Benjamins.

Aparicio, J. et al. (2008). 'AnCora-Verb: A Lexical Resource for the Semantic Annotation of Corpora'. In *Proceedings of Language, Resources and Evaluation*, LREC 2008. Marrakech, Morocco.

Boas, H (2009). *Multilingual FrameNets in Computational Lexicography. Methods and Applications*. Mouton de Gruyter.

Bohmova, A. et al. (2001). 'The Prague Dependency Treebank: Three-Level Annotation Scenario' In *Treebanks: Building and Using Syntactically Annotated Corpora, ed. Anne Abeille. Kluwer Academic Publishers.*

Brants, S et al., (2002). 'The TIGER Treebank'. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*. Sozopol.

Fleiss, J.L. (1981). Statistical methods for rates and proportions. New York: John Wiley.

Grimshaw, J. (1990). *Argument Structure*. Cambridge, MA.: The MIT Press.

Kingsbury, P. et al. (2002). 'Adding semantic annotation to PennTreeBank'. In *Proceedings of the 2002 Conference on Human Language Technology*. San Diego, CA.

Meyers, A. (2007). 'Anotation Guidelines for NomBank-Noun Argument Structure for PropBank'. Online Publication:
http://nlp.cs.nyu.edu/meyers/nombank/nombank-specs-2007.pdf

Palmer, M. et al. (2005). 'The Proposition Bank: An Annotated Corpus of Semantic Roles', *Computational Linguistics*, 21 (1). MIT Press.

Peris, A. and M.Taulé (2009). 'Evaluación de los criterios lingüísticos para la distinción evento y resultado en los sustantivos deverbales'. In *Proceedings of the 1st International Conference on Corpus Linguistics.* Murcia (Spain).

Peris et al. (2010). 'ADN-Classifier: Automatically assigning denotation types to nominalizations'. *In Proceedings of the 7th International Conference on Language Resources and Evaluation* (LREC-2010). Valleta. (Malta).

Peris, A. and M.Taulé (forthcoming paper). 'Annotating the Argument Structure of Deverbal Nominalizations in Spanish '.

Picallo, C. (1999). 'La estructura del sintagma nominal: las nominalizaciones y otros sustantivos con complementos argumentales'. In Bosque & Demonte (Ed.) *Gramática descriptiva de la lengua española*. Madrid: Real Academia Española / Espasa Calpe.

Santiago, R. and E. Bustos (1999). 'La Derivación Nominal'. In Bosque & Demonte (Ed.) *Gramática descriptiva de la lengua española.* Madrid: Real Academia Española / Espasa Calpe.

Scott, W. A. (1955). 'Reliability of content analysis: The case of nominal scale coding'. *Public Opinion* Quarterly, 19(3): 321–325.

Siegel, S. and N. J. Castellan. (1988). *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, New York.

Taulé, M. et al. (2008). ' Ancora: Multilevel Annotated Corpora for Catalan and Spanish'. In *Proceedings of 6th International Conference on Language Resources and Evaluation*. Marrakesh (Morocco).

Vendler, Z. (1967). *Linguistics in Philosophy*. Cornell University Press, Ithaca, N.Y.

# The Scope and the Sources of Variation in Verbal Predicates in English and French

Tanja Samardžić, Lonneke van der Plas, Goljihan Kashaeva, Paola Merlo
University of Geneva
Linguistics Department, Rue de Candolle 2, 1204 Geneva
E-mail: {Tanja.Samardzic, Lonneke.vanderPlas,
Goljihan.Kashaeva, Paola.Merlo}@unige.ch

**Abstract**

In this article, we examine the variation of the predicate-argument structure between English and French in an experimental and data-driven approach. We annotate a corpus of 1000 French sentences with predicate-argument structure using a framework originally developed for English. We identify a number of non-matching predicates and examine them in a qualitative analysis. We show that these two languages do not differ substantially in the inventory and the nature of verbal predicates, even though certain general grammatical properties may result in some variation at this level of representation. We argue that the proposed comparative study can provide a basis for identifying the level of specificity needed for developing a framework for multilingual annotation of predicate-argument structure.

## 1 Introduction

The analysis of the predicate-argument structure of a sentence results in a linguistic representation that defines relations between the constituents in the sentence that cannot be defined by the rules of syntax only.

Cross-linguistically, the predicate-argument structure of a sentence is considered to be more stable (less varying) than its syntactic form. The English sentence in (1a) can be considered as equivalent to the French sentence in (1b), despite the fact that the positions of their syntactic subjects are occupied by different kinds of lexical elements and that the complements of the verbs differ both syntactically and semantically. The predicate-argument structure of (1a) and (1b) is equivalent since verbs (*liked*, *a plu*), which are the predicates of the sentences, take the same kind of arguments (EXPERIENCER, CONTENT) in both languages.

(1) a. [EXPERIENCER Mary] liked [CONTENT the idea]. (English)
b. [CONTENT L'idée] a plu [EXPERIENCER à Marie]. (French)

Not all predicates behave in the same way in a cross-linguistic context. Some of them are known to give rise to more parallel or cross-linguistically stable syntactic structures, while others result in more divergent syntactic realizations. For instance, the predicates expressed by verbs meaning “creation”, such as English *create, build, construct* are likely to be realized as transitive verbs with direct objects in many different languages, while predicates denoting some state of mind, such as the verbs in (1), can be involved in different syntactic structures in different languages [13].

Linguistic investigations into predicate-argument structure have been mostly concerned with the kinds of labels (semantic or thematic roles) that can be assigned to the arguments. In addressing cross-linguistic variation, these approaches assume that the same predicates are associated with the same argument structure in different languages. Their focus is on the differences in syntactic realisation of the arguments and the semantic features that give rise to the variation. Little attention has been given to the cross-linguistic variation in the argument structure itself. However, a corpus-based analysis of the variation between English and Chinese [5] suggests that cross-linguistic parallelism in the predicate argument structure cannot be assumed. This study shows that 17% of arguments of English predicates do not map to the arguments of the corresponding Chinese predicates.

The aim of our study is to examine the variation of the predicate-argument structure between English and French predicates in an experimental and data-driven approach. We perform a qualitative analysis of divergent predicates to identify general principles underlying the variation. We argue that the proposed comparative study can provide a basis for establishing the level of specificity needed to develop a framework for multilingual annotation of predicate-argument structure.

We identify the varying predicates by annotating the predicate-argument structure in a corpus of one-thousand naturally occurring French sentences using an annotation framework originally developed for English. Cases in which the English annotation framework fails to be applicable to French predicate-argument structure are selected as strong examples of cross-lingual variation. Such an approach rests on two important methodological decisions.

First, we compare two closely related and well-documented languages. This choice is based on the assumption that English and French constitute a minimal pair suitable for micro-comparisons [9]. Since the variation in the predicate-argument structure is related to lexical items rather than to broader syntactic structures, micro-comparison seems to be the adequate approach, providing a good setting for identifying the elements of grammar that potentially underlie the variation.

Second, we take a corpus-based approach analysing a big sample of naturally occurring instances of predicates using an annotation scheme as a framework. This decision is motivated by the fact that studying linguistic variation requires analysing large amounts of data. Furthermore, the results of such an analysis can be directly incorporated in developing or improving tools for automatic natural language processing, especially automatic analysis of the predicate-argument structure.

## 2 Choosing the annotation framework

There are three current frameworks proposed for annotating corpora with predicate-argument structure: FrameNet [2], VerbNet [11], and PropBank [16]. All of them have been developed on the basis of English data and have been used to annotate English corpora. However, these frameworks are implementations of linguistic theories of the predicate argument structure that have been developed to account for universal phenomena, which is why they can be expected to apply to other languages as well.

All three resources describe the same linguistic phenomena covering approximately the same range of lexical items. Nevertheless, they are conceived with different purposes and with different theoretical backgrounds. As a result, the choice of the framework can have important consequences for the outcome of the annotation, especially in a cross-linguistic setting.

Predicate labels used in FrameNet (frames) and VerbNet (verb classes), are intended to capture the level of lexical semantics which is common to a group of lexical items. Different words can bear the same label. For instance, the verbs *accomplish, achieve, bring about* all bear the label *Accomplishment* in FrameNet. This is not the case in the PropBank framework, where predicate labels capture the specific meaning of verb senses. This is why the style of annotation used in FrameNet and VerbNet can be considered as more abstract and, thus, more portable across languages than the annotation in PropBank. Indeed, FrameNet has often been used as a basis for developing similar resources for other languages [4].[1] Furthermore, argument labels in FrameNet are assigned without taking into account the syntactic function of the constituents that bear them, while the PropBank argument labels can depend on the syntactic function. This feature makes the FrameNet annotation less tied to the syntactic representation and thus to a particular language than it is the case with the PropBank annotation.

In our study, however, we use the PropBank framework to annotate the predicate-argument structure of French sentences. We take this decision for two reasons.

First, the lexicon in this resource is built by extracting and describing all the predicates that occur in a predefined sample of naturally occurring sentences. Since our aim is to annotate exhaustively a corpus of naturally occurring sentences, we expect that such a lexicon provides a better coverage than the lexicon in FrameNet, which is not developed with a corpus-driven methodology.

Second, the labels used in PropBank both for predicates and arguments involve fewer theoretical assumptions than the labels in FrameNet. While FrameNet labels capture mostly linguistic intuition about the targeted level of lexical semantics and the relations between the lexical items, the PropBank labels rely strongly on the observable behaviour of words. The distinctions between verb senses, for instance, are made taking into account the differences in subcategorisation frames of each

[1] VerbNet is a relatively new resource and it is rarely used for automatic labelling of semantic roles.

sense. This approach can be expected to provide more tangible criteria for annotators in deciding how to annotate each instance of the predicate-argument structure found in the corpus, ensuring a more reliable and more consistent annotation.

## 3 Materials and methods

In identifying the points of variation in the predicate-argument structure between English and French, we rely on corpus data. We manually annotate a corpus of one-thousand French sentences using the PropBank annotation framework originally developed for English. Those French predicates for which no appropriate PropBank label (English verb sense) can be found are considered as varying. These sentences are automatically extracted from the corpus and manually analysed. We normalised the instances of predicates extracted to their citation form and looked up a comprehensive French-English dictionary to identify the English translation of the French predicate that is closest to the literal translation. If needed, the translations are verified with a native speaker. For example, we find the English expression *shed light* as the translation for the French expression *faire la lumière*. Finally, we classify and analyse the types of varying predicates, discussed in Section 4.

We envisage to detect cross-linguistic variation by looking at those cases where the manual annotation using a resource developed for a source language (English) fails to be applicable to the target language (French). It must therefore be ensured that failure is due to a genuine inability to provide a common analysis for the two languages and not to other causes. To ensure good quality and reliability of the annotation, we provided the annotators with detailed guidelines, that consist of an extensive description of the PropBank annotation framework (adapted from the original PropBank guidelines [1] to the French language), parallel English and French examples of predicate-argument analysis, and a manual on how to use the annotation tool[2]. Furtermore, we offered the annotators intensive training. The main annotation is preceded by a two-stage training phase, and a calibration phase, following [14]. More details on the annotation procedure can be found in [17]. For our current investigations into divergences in predicate argument structure, it is important to stress that the task of using an English resource to annotate French predicates is well-defined. This is reflected in high inter-annotator agreement (an average F-score of 95%) after discussion and individual re-annotation. In the calibration phase, when annotators were not able to discuss the annotations, an average F-score of 81% is reached. This score is calculated after reducing the very fine-grained PropBank verb sense labels to more general verb class labels with the predicate-argument structure preserved. We can assume that failure to annotate French predicates using the English resource is due to a genuine inability to provide a common analysis for the two languages and not to other causes.

The French corpus that we annotated consists of 1040 sentences drawn from

[2] The annotation tool we use is an adaptation of the user-friendly, freely available Tree Editor (TrEd, [15]).

the French portion of the Europarl corpus [12]. This corpus contains translations of the proceedings of the European Parliament in 11 languages parallelised at the sentence level.[3] We use 40 sentences for the training, so that the final French corpus annotated with predicates and arguments contains 1000 sentences.

## 4 Non-matching predicates

We find 90 instances of predicates that could not be annotated for a total of 1985 predicates in the 1000 sentences of the annotated corpus. This ratio indicates that a great majority of French predicates found in our corpus directly corresponds to an English verb sense with the same predicate-argument structure. The PropBank lexicon provides a better coverage for our corpus than the FrameNet lexicon for annotating a German corpus of a similar size [4], where 30% of instances of German predicates required adding a label to the existing FrameNet set.

The mismatching predicates are mostly conventionalised expressions with different degrees of semantic compositionality, including light verb constructions, collocations, and fully uncompositional idioms.[4]

In this section, we group and analyse the non-matching predicates in an attempt to identify the categories that are potential sources of variation. We classify the predicates according to the degree to which they differ from the closest English counterparts.

The degree of parallelism between the corresponding expressions in the two languages is defined in terms of the cross-linguistic stability of the lexical categories and syntactic properties of the lexical items involved in the expressions. On the basis of the assumption that the lexical category is not an inherent feature of a lexical item ([7]), we consider items such as the English verb *need* and the French noun *besoin* as matching, even though they are associated with different lexical categories in the two languages. By syntactic properties we understand the subcategorisation frame of a lexical item, specifying the number and form of its complements.

The French structures that are most parallel with English are those where the corresponding lexical item preserves its category across languages, and the arguments of the predicates preserve their syntactic form (Group 1). Expressions that include predicates that require different syntactic forms of their arguments in the two languages are considered less parallel (Group 2). Group 3 includes the expressions where both the category of corresponding lexical items differs and syntactic

[3] Even though we use the French side as a monolingual corpus in our study (the annotators do not have access to the actual translations), we take the sample from a parallel corpus because this gives us a possibility to look up the actual translations of the expressions we are interested in while analysing the variation. In addition, manual annotation of one side of a parallel corpus can be a useful resource for experiments in automatic annotation transfer.

[4] Our analysis does not address idioms (3 out of the 90 instances). We also exclude one mismatch which is clearly not due to linguistic reasons (the PropBank lexicon did not contain the English verb *dramatise* for the French verb *dramatiser*). This leaves us with 86 instances that are discussed.

divergences take place. Lastly, for a number of French predicates and their English counterparts it was not possible to find any lexical correspondence. These predicates are classified as Group 4.

### 4.1 Characteristics of the non-matching predicates

The first property which can be noticed about the non-parallel transitive constructions is that they tend to be headed by very frequent verbs with rather general meaning. The most characteristic verb for these constructions is the verb *faire* which occurs in all four types of mismatches. It can have two kinds of meaning. One of its meanings can be described as vaguely causative, denoting that its argument which is syntactic subject (ARG0 in the PropBank annotation) brings about the entity denoted by its object (ARG1), e.g. *faire pression* : *put pressure*. The other meaning is copulative, denoting that the argument which is its syntactic object (ARG1) denotes some property of the other argument (ARG0), e.g. *faire l'objet de* : *be the objective*. The meaning of the other verbs used in these constructions can be described in these terms as well.

The arguments considered as ARG1 in these constructions are headed by abstract nouns, including deverbal nouns, such as *pression* and *objectif* in the examples above. The meaning of these nouns is generalised with no specific reference, which is often reflected in the fact that they are not preceded by an article.

There are two dominant ways in which French transitive constructions are transformed into the corresponding English expressions. The transitive verbs of the groups 1 and 2, where the lexical categories are preserved, correspond either to an English verb with the same properties (general meaning of the same type), or to the copulative *be*. Transitive expressions of the group 3 are typically transformed into a single English verb, if they involve a verb with vaguely causative meaning. Otherwise, they are transformed into a copulative construction. Expressions of group 4 are mainly transformed into a single English verb.

The mismatching expressions with intransitive verbs as predicates are less numerous than those with transitive verbs. The examples found in our corpus provide a basis for identifying two factors that can make intransitive verbs in French hard to match with English verbs. We note that most of the examples are pronominal verbs (*s' exprimer, se mettre d'accord, se réjouir, se féliciter, se prononcer*), where the reflexive particle does not bear a semantic role. Another group of predicates are impersonal verbs (*il suffit, il convient*), which do not assign a semantic role to their syntactic subject and whose use is limited to a single morphological form (3rd person singular). Some predicates are characterised by both features (*il s'agit de, il se peut que*). We note that these predicates are most often transformed into a copulative construction in English.

There are two expressions with intransitive verbs that are neither pronominal nor impersonal (*tenir de, tomber bien*). Their transformation to English forms follows the patterns identified with other predicates. The expression *tenir de* corresponds to English copulative constructions, while the English form that corresponds to *tomber bien* is another verb (*come*), but with the same kind of general meaning. Both French *tomber* and English *come* are unaccusative verbs of motion.

## 4.2 Overcoming the cross-linguistic divergence

The structures associated with French predicates identified as not matching with English predicates during the annotation still retain a certain level of parallelism with the corresponding English structures. In most of the cases, it is precisely the number and the meaning of the predicates' arguments that remain unchanged. Variation can be identified at other levels of the structure. It is limited to two domains: the choice of the lexemes which encode the impoverished meaning of the predicates and the choice of lexical category of the predicating lexical unit.

For cross-linguistic mapping of the predicate-argument structures which involve variation in one of the two domains, the representation of the structure needs to be slightly more abstract than the one that is currently used in PropBank. In this section, we propose the representations of the structures which are valid cross-linguistically introducing only minimal generalisations needed to address the observed variation.

### 4.2.1 Parallel structures: Group 1

In the case where the heading verbs are not translations of each other, while the rest of the structure is preserved across languages, we propose assigning a special label to the verbs. Since the meaning of these verbs is impoverished in these usages, no specific verb sense label can account for it. The label on these verbs needs to express their general meaning and it needs to be applicable to multiple verbs. Such a representation is given in (2).

(2) a. [ARG0 Nous] [REL-CAUS **tirons**] [ARG1 les leçons du passé]. (French)
b. [ARG0 We] [REL-CAUS **learn**] [ARG1 the lessons of the past]. (English)

The following predicates belong to this group:

avoir lieu : take place;
avoir rien à voir : have nothing to do;
attirer l'attention : draw attention;
céder la place : make room;
faire la lumière : shed light;
faire alliance : form an alliance;
faire pression : put pressure;
faire appel : make an appeal;
tirer la leçon: learn the lesson

The semantically impoverished verb in French is assigned a more abstract label CAUSE. This label marks two characteristics of the predicate at the same time: the fact that it is not an ordinary predicate and its general meaning. All the impoverished verbs in our sample have the same general meaning, vague cause, but other labels could be used for other general meanings that can potentially occur.

This approach would require a set of abstract labels to be defined within the framework of PropBank in addition to the verb sense labels.[5] These labels would resemble the FrameNet labels, since they would express an abstract layer of the meaning of the verbs and since they would be applied to multiple lexical units. The difference is that these labels would encode a more abstract meaning than it is the case with the FrameNet labels. Also, they would be used only for a very limited set of lexical items in special contexts where their meaning is impoverished.

### 4.2.2 Almost parallel structures: Group 2

A case of almost parallel structures is represented in (3), where the French pronominal verb corresponds to the English transitive verb. A parallel representation of this sentence requires treating the pronominal clitic in French (*m* in (3a)) as one of the verb's arguments and assigning it the same label as to the corresponding argument in English (ARG1 in (3b)).

(3) a. [ARG0 Je] [ARG1 **m'**] [REL-EXPRESS.01 **exprime**] [ARG2 sur le texte de M. L]. (French)

b. [ARG0 I] [REL-EXPRESS.01 **express**] [ARG1 **my opinion**] [ARG2 on the Mr L.' s document]. (English)

The meaning of the ARG1 in the English sentence can be interpreted as non-specific or general, but this has no consequences for the annotation framework, since the argument labels are already general.

Apart from the intransitive verb given in (3), this group includes the following transitive verbs:

faire l'objet de : be the objective;
laisser qqn. sceptiqe : be sceptical;
tenir compte de : take into account;
avoir qqch comme objectif : be the objective;
donner à qqn. à penser : make sbd. think;

### 4.2.3 Category changing structures: Group 3

The non-parallel structures that involve a change in the lexical category of the predicating word require a stronger deviation from the standard PropBank representation. Most of these cases involve a predicate that is expressed synthetically in one language (as a single verb), while it is expressed analytically in the other language (as a combination of a verb and a predicating complement). The corresponding lexical units in these cases are the synthetic verb in one language and the verb's complement in the analytical expression in the other language. The following predicates belong to this group:

[5] Efforts in this directions have already been made [8]

Transitive:
avoir besoin de : need;
avoir trait de : have to do;
avoir retard : be late;
donner suite à : follow up;
y faire face : face;
faire état : state;
faire défaut : lack;
mettre qqch en exergue : highlight;
porter remède : remedy;
prendre conscience de : be/become aware;
rendre compte : be accountable

Intransitive:
se mettre d'accord : agree;
se réjouir : be happy;
il suffit : be enough

A possible way of representing the cross-linguistic parallelism in the predicate-argument structure of these expressions is to label the corresponding lexical items with a predicate label, leaving the heading verb of the analytical expression without a label, as illustrated in (4).

(4) a. [ARG0 Le marché] **a** [REL-NEED.01 **besoin**] [ARG1 de règles]. (French)
b. [ARG0 The market] [REL-NEED.01 **needs**] [ARG1 rules]. (English)

The verbal head of the French analytical predicate (*a*) is not assigned a predicate label. Instead, the predicate is the noun heading its complement (*besoin*), which is, at the same time, lexical counterpart of the verb in English (*need*). The unlabelled verb in this case would be treated as a functional word which has no semantic arguments — a lexicalised light verb ( [7], [6], [10]).

Such representation is not possible in the current PropBank setting due to two limitations. First, all the verbs except the copulative *be* are considered as predicates. Leaving a non-copulative verb without a label would go against this principle. Moreover, strictly functional treatment of light verbs is subject of debate in the literature, with some authors arguing in favour of such approach for certain verbs ([6], [10]) and others providing evidence of semantic content of light verbs [3]. Second, only verbs are considered as predicates in PropBank which is why the frames are not specified for other lexical categories. This means that an existing verbal frame in the English resource would need to be adapted to be used with a French nominal predicate. The limitation is even more important in predicates which are expressed in an analytical form in English, such as *be enough* in (5b), corresponding to the French *suffit* in (5a). The argument structure of the English adverb enough is not specified at all in PropBank's frame files.

(5) a. Il [REL-ENOUGH **suffit**] [ARG0 de lire le programme]. (French)
b. It **is** [REL-ENOUGH **enough**] [ARG0 to read the manifesto]. (English)

### 4.2.4 Expressions with no direct translations: Group 4

Finally, the cases identified as the strongest mismatches in our classification can be resolved applying one of the described approaches too, as illustrated in (6).

The only difference is that the English label will not be a direct translation of the French word in question. This, however, should not pose a problem for the result of annotation. The label does not have to be a direct translation. It only needs to be consistent (and consistently found in a corpus) with the word.

(6) a. [ARG0 Les dirigeants politiques] doivent **faire** [REL-SHOW.01 **preuve**] [ARG1 d' un réel courage]. (French)

b. [ARG0 Political leaders] must [REL-SHOW.01 **show**] [ARG1 real courage]. (English)

This group includes the following predicates:

Transitive:
faire preuve de : show;
se faire un plaisir : be happy;
mettre qqch en oeuvre : implement;
remettre qqch en état : repair;
porter atteinte : undermine;
prendre la parole : speak;
prendre la peine : bother;
assigner qqn. à résidence : put sbd. under house arrest;
metre qqch. en cause : call stg. into question

Intransitive:
il s'agit de : be about;
il convient : should;
se féliciter : be pleased;
il se peut que : it is possible that;
se prononcer : give an opinion;
tenir de : be;
tomber bien : come in a good moment

The analysis of the mismatching expressions with lexical categories preserved across languages (groups 1 and 2) suggests that using a special label for verbs with impoverished meaning would ensure a more adequate and cross-linguistically valid annotation framework.

The analysis of the category changing predicates suggests that the lexical category of the predicating lexical items can change across languages, while the argument structure is still preserved. Therefore, specifying predicate-argument structures for all predicating lexical items, including verbs, nouns, and adjectives is important for making an annotation framework directly portable across languages.

# 5 Conclusions

A translation is an implicit representation of the meaning of a sentence. Applying the same annotation scheme for different languages is a way of making explicit the components of meaning that are expressed in a predicate.

The general linguistic conclusion of our investigation is that most mismatches concern the lexical realisations of the predicates and not the argument structure itself. Even for the cases identified as mismatches in our study, the number and the kind of arguments are unchanged across languages. The mismatches are due to the different choices the two languages make with respect to lexical items that realise the meaning of semantically impoverished verbs, and to the lexical category of the predicating word. Our analysis also suggests that the structural correspondence fails more when a clause contains a verb with less specific meaning. This correlation between specificity of meaning and structural realisation is a confirmation of the cross-linguistic validity of those approaches, such as [13], that hypothesize a direct correspondence between the components of meaning of a predicate and the syntactic realisation of the arguments.

Applying a specific annotation framework such as PropBank in a cross-linguistic study proved plausible. A great majority of French predicates found in our corpus (95 %) corresponded directly to an English verb sense specified in PropBank. Our findings about the sources of the mismatches in the remaining predicates indicate that using a small set of special labels for verbs with impoverished meaning and providing annotation for all categories of predicating words (verbs, nouns, and adjectives) would improve the empirical adequacy and cross-linguistic validity of this framework.

## Acknowledgements

The work described in this paper was financed by the EU FP7 programme under grant agreement 216594 (CLASSIC project: www.classic-project.org) and by SNSF (project No. 122643).

## References

[1] Olga Babko-Malaya. Propbank annotation guidelines. 2005.

[2] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 86–90, Montreal, Canada, 1998. ACL / Morgan Kaufmann Publishers.

[3] Claudia Brugman. Light verbs and polysemy. *Language Science*, 23:551–578, 2001.

[4] Aljoscha Burchardt, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Padó, and Manfred Pinkal. FrameNet for the semantic analysis of German: Annotation, representation and automation. In Hans Boas, editor, *Multilingual FrameNets in Computational Lexicography: Methods and applications*, pages 209–244. Mouton de Gruyter, 2009.

[5] Pascale Fung, Zhaojun Wu, Yongsheng Yang, and Dekai Wu. Learning bilingual semantic frames: Shallow semantic parsing vs. semantic role projection.

In *11th Conference on Theoretical and Methodological Issues in Machine Translation (TMI 2007)*, pages 75–84, Skovde, Sweden, 2007.

[6] Jane Grimshaw and Armin Mester. Light verbs and theta-marking. *Linguistic Inquiry*, 19:205–232, 1988.

[7] Kenneth Hale and Samuel Jay Keyser. On argument structure and the lexical representation of syntactic relations. In Kenneth Hale and Samuel Jay Keyser, editors, *The View from Building 20*, pages 53–110. MIT Press, 1993.

[8] Jena Hwang, Archna Bhatia, Claire Bonial, Aous Mansouri, Ashwini Vaidya, Nianwen Xue, and Martha Palmer. Propbank annotation of multilingual light verb constructions. In *In Proceedings of the 4th Linguistic Annotation Workshop (The LAW IV)*, pages 82–90, Uppsala, Sweden, 2010.

[9] Richard Kayne. *The Oxford Handbook of Comparative Syntax*, chapter Some notes on comparative syntax, with special reference to english and french. Oxford University Press, 2005.

[10] Kate Kearns. Light verbs in English. Manuscript, 2002.

[11] Karin Kipper Schuler. *VerbNet: A broad-coverage, comprehensive verb lexicon*. PhD thesis, University of Pennsylvania, 2005.

[12] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit 2005*, Phuket, Thailand, 2005.

[13] Beth Levin and Malka Rappaport Hovav. *Argument realization*. Cambridge University Press, Cambridge, 2005.

[14] Sebastian Padó. *Cross-Lingual Annotation Projection Models for Role-Semantic Information*. PhD thesis, Saarland University, 2007.

[15] Petr Pajas and Jan Štěpánek. Recent advances in a feature-rich framework for treebank annotation. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 673–680, Manchester, UK, 2008.

[16] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105, 2005.

[17] Lonneke van der Plas, Tanja Samardžić, and Paola Merlo. Cross-lingual validity of propbank in the manual annotation of french. In *In Proceedings of the 4th Linguistic Annotation Workshop (The LAW IV)*, pages 113–117, Uppsala, Sweden, 2010.

# Grammatical number of nouns in Czech: linguistic theory and treebank annotation

Magda Ševčíková, Jarmila Panevová, Zdeněk Žabokrtský

Charles University in Prague, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
E-mail: {sevcikova,panevova,zabokrtsky}@ufal.mff.cuni.cz

**Abstract**

The paper deals with the grammatical category of number in Czech. The basic semantic opposition of singularity and plurality is proposed to be enriched with a (recently introduced) distinction between a simple quantitative meaning and a pair/group meaning. After presenting the current representation of the category of number in the multi-layered annotation scenario of the Prague Dependency Treebank 2.0, the introduction of the new distinction in the annotation is discussed. Finally, we study an empirical distribution of preferences of Czech nouns for plural forms in a larger corpus.

## 1 Introduction

Morphological categories are described from formal as well as semantic aspects in grammar books of Czech (and of other languages; for Czech see e.g. [7]). Both these aspects are reflected in the annotation of Prague Dependency Treebank version 2.0 (PDT 2.0; see [4] and http://ufal.mff.cuni.cz/pdt2.0). In the present paper, the grammatical category of number of nouns is put under scrutiny.

In Section 2, we explain the basic semantic opposition of the category of number (singularity vs. plurality, prototypically expressed by singular and plural forms, respectively) and focus on nouns that are used predominantly, or even exclusively, either in singular or in plural. Our analysis is based on large corpus data and confronted with the traditional linguistic terms (such as singularia tantum, pluralia tantum). We propose to enrich the basic singular-plural opposition with the opposition of simple quantitative meaning (concerning number of entities) vs. pair/group meaning (number of pairs/groups).

Current representation of formal and semantic features of the category of number within the multi-layered annotation scenario of PDT 2.0 is briefly introduced in Section 3. The PDT 2.0 annotation scenario has been designed on the theoretical basis of Praguian Functional Generative Description (FGD; [14]). As the linguistic theory of FGD has been still elaborated and refined in particular aspects, the question of the introduction of the recent theoretical results is to be asked. Section 4 brings novel observations related to the distribution of preferences of individual nouns for plural forms derived from the Czech National Corpus data.

# 2 The category of number in Czech

## 2.1 Meaning and form of the category of number

Number is a grammatical category of nouns and other parts of speech in Czech. With nouns, the category of number reflects whether the noun refers to a single entity (singularity meaning) or to more than one entity (plurality). With other parts of speech (adjectives, adjectival numerals, verbs), the number is imposed by agreement. In this paper we deal with the number of nouns only.[1]

In Czech, the number is expressed by noun endings. Czech nouns have mostly two sets of forms directly reflecting the opposition of singularity and plurality: singular forms and plural forms. Nouns *ruka* ‘arm’, *noha* ‘leg’, *oko* ‘eye’, *ucho* ‘ear’, *rameno* ‘shoulder’, *koleno* ‘knee’ and diminutives of these nouns have an incomplete set of (historical residues of) dual forms as well, which are used instead of the plural forms when referring to body parts.

According to the data from the SYN2005 subcorpus of the Czech National Corpus,[2] singular and plural forms of nouns occur roughly in the ratio 3:1 in Czech texts (22,705,247 singular forms:7,440,382 plural forms). Concerning the ratio of singular and plural forms for single nouns, a detailed analysis of the SYN2005 data was carried out. The SYN2005 corpus contains 452,015 distinct noun lemmas, only those with more than 20 occurrences were involved in the analysis (48,806 lemmas). The majority of Czech nouns (42,550 lemmas out of 48,806) is used in singular more often than in plural (see Fig. 1 (a); for further details see Sect. 4).[3] At both ends of the scale there are nouns that clearly prefer either singular or plural forms, or are even limited to the singular on the one hand or to the plural on the other.

In our opinion, the predominance of singular or plural forms can be traced back, roughly speaking, to two factors. The first of them is the relation of the noun to the extra-linguistic reality: some nouns refer to objects that occur in the reality mostly separately or in large amounts, respectively. The other one lies in the language itself, more specifically, in the process of structuring the described reality by the language: for instance, groups of some entities are considered as a

[1] According to Mathesius ([8]), the number of nouns belongs to functional onomatology, with other parts of speech it is a part of functional syntax.

[2] SYN2005 is a representative corpus of Czech written texts, containing 100 million both lemmatized and morphologically tagged tokens ([2]).

[3] The ratio between singular and plural forms corresponds to the known language principle considering the singular to be the unmarked member of the basic number opposition, it can be used in some contexts instead of its marked counterpart (plural).

single whole and as such referred to by the singular form (e.g. so-called collective nouns); on the contrary, some single objects are, often due to their compoundness or open-endness, referred to by the plural form (so-called pluralia tantum).

Nouns preferring singular or plural forms are discussed in Sect. 2.2 and 2.3, in Sect. 2.4 a new semantic distinction is introduced.

## 2.2 Nouns preferring singular forms

More than a third of the analyzed noun lemmas (16,473 lemmas out of 48,806) was used exclusively in singular in the SYN2005 data. Most of them are proper names (12,286 lemmas; see Fig. 1 (a) and (b)).[4] Only singular forms were found also for nouns such as *dostatek* 'sufficiency', *údiv* 'astonishment', *sluch* 'hearing', *kapitalismus* 'capitalism', *zámoří* 'overseas', *severovýchod* 'northeast', *potlačování* 'repression', *pohřbívání* 'burying', *arabština* 'Arabic'. A strong preference, though not exclusivity, of singular forms is characteristic for nouns denoting a person, an object, an institution, en event etc. that is unique or fulfils a unique function in the given context or segment of reality, e.g. *svět* 'world', *republika* 'republic', *prezident* 'president', *začátek* 'beginning', *centrum* 'center' (see e.g. uniqueness of *Česká republika* 'Czech Republic', *prezident USA* 'President of the U.S.').

In case of proper names, the predominance of singular forms can be seen as anchored in the extra-linguistic reality (see Sect. 2.1): as they refer to a person, an object, a place etc. and identify them as individuals, they often occur in singular. However, for the absolute majority of Czech proper names plural forms can be formed, they are used to refer to several persons, objects etc. named with the same proper name (see the plural of the first name *František* in ex. (1)) or metaphorically (ex. (2)) etc. The other (i.e. intralinguistic) "reason" for the preference of singular concerns collective nouns (e.g. *dělnictvo* 'labour', *hmyz* 'insects', *listí* 'leaves').

Proper names, collectives as well as mass nouns (*voda* 'water', *měď* 'copper'), names of processes and qualities (*kvašení* 'fermentation' and *sladkost* 'sweetness') and possibly other are subsumed under the term of singularia tantum in grammar books of Czech (e.g. [7]). Since we have found out that plural forms of proper names, collectives etc. do occur in the corpus data (though with a lower, but not insignificant frequency) we consider the term singularia tantum rather inappropriate. Beside this, in grammars of Czech the scope of this class is defined vaguely. The potentiality of the grammatical system to form a full paradigm with both singular and plural forms opens the possibility to use these plural forms for meaning shifts, metaphors, occasionalisms etc. (see ex. (3) and (4)).[5]

[4] There were nearly 20,000 proper name lemmas with 20 or more occurrences found in the SYN20005 data. More than 12,000 of them occurred in singular only, more than 900 had only plural forms (Sect. 2.3), more than 7,000 proper name lemmas were used both in singular and plural.

[5] The other way round, when considering the nouns that are truly limited to singular in the corpus data, they have no noticeable (semantic, derivational etc.) feature in common and to cover them with the term singularia tantum seems not to be profitable.

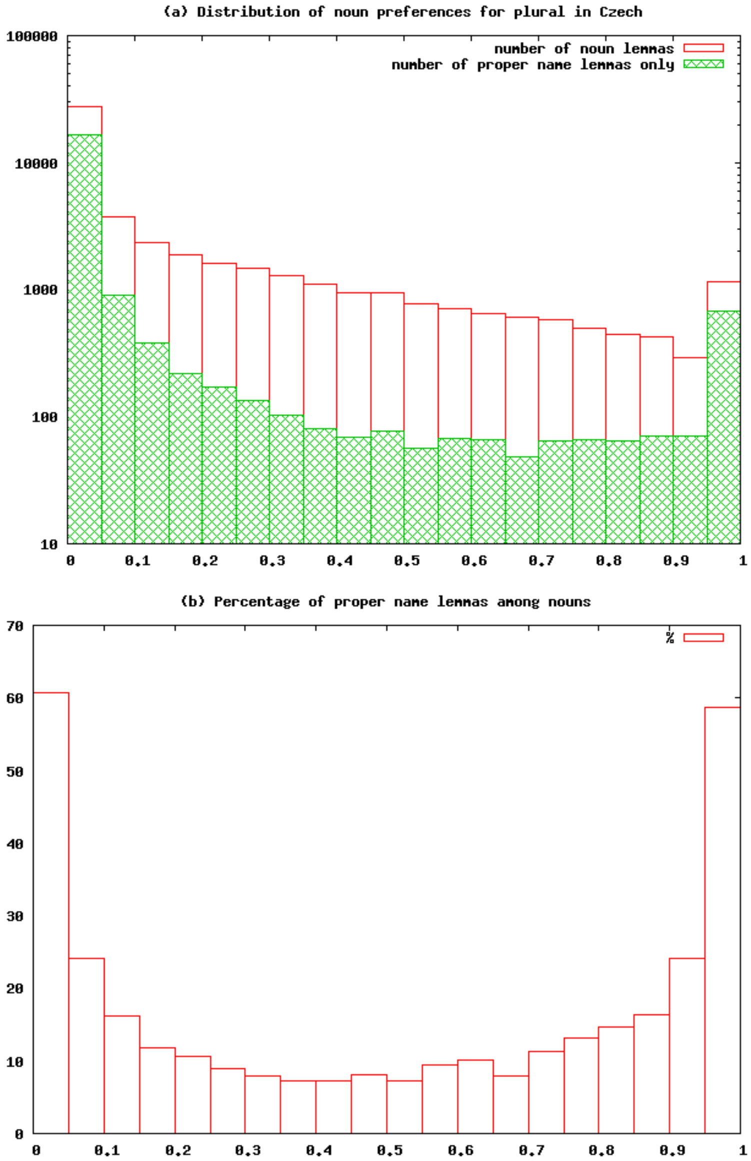


Figure 1: (a) Histogram of noun preferences for plural according to the SYN2005 corpus. The horizontal axis represents the ratio of plural forms (among all occurrences of a noun lemma), the vertical axis represents the number of distinct lemmas having its preference for plural in a given interval. (b) Percentage of proper names among noun lemmas with respect to their preferences for plural.

(1) *Třeba na Tomáše Kaberleho čekali dva* ***Františkové****, otec i bratr.* 'For instance, two **Františeks**, father and brother, were waiting for Tomáš Kaberle.' (SYN2005)

(2) *Vymizí* ***Goethové****, také* ***Beethovenové*** *se ztratí či jsou umlčeni.* '**Goethes** disappear, **Beethovens** get lost or will be silenced as well.' (SYN2005)

(3) *Čerstvé listy špenátu očistíme a propláchneme alespoň ve třech* ***vodách***. 'Fresh spinach leaves are to be cleaned and washed in at least three **waters**.' (SYN2005)

(4) *Nikdy neodolá* ***sladkostem***. 'He never resists **sweets**.' (SYN2005)

## 2.3 Nouns used predominantly in plural

The preference of plural is typical for much fewer nouns than the preference of singular; only 941 lemmas (out of 48,806 noun lemmas analyzed) occurred more often in plural than in singular in the SYN2005 corpus data. Most of these nouns are proper names (607 lemmas; in particular toponyms) and nouns such as *záda* 'back/backs', *noviny* 'newspaper/newspapers', *vrata* 'gate/gates', which are referred to as pluralia tantum in linguistic terminology. The set of forms of most pluralia tantum is truly limited to plural forms, the plural is used to refer to a single object as well as to a number of them (the current meaning is to be resolved on the basis of context, knowledge of situation etc.; cf. the noun *dveře* 'door/doors' in ex. (5)). A singular form of a plurale tantum is used only exceptionally (e.g. with *kalhoty* 'trousers', *brýle 'glasses'*), nevertheless, it has usually a shifted meaning (*kalhota* as *nohavice* 'trouser leg') and the plural preserves the ambiguity. The term of pluralia tantum proved to be adequate according to the performed data analysis (it is used in the present paper).

Besides proper names and pluralia tantum, among nouns with only plural forms there were only few nouns that have both singular and plural forms, e.g. *arašíd* 'peanut', *monocyt* 'monocyte', *autodíl* 'spare part', *johanita* 'Knight of Malta'. In our opinion, the singular of these nouns, though commonly available, did not occur in the data since the nouns refer to persons, objects etc. that usually occur in groups or large numbers in the reality. The same reason applies to nouns such as *hasič* 'fireman', *potravina* 'food', *živina* 'nutrient', *dohad* 'guess' that occurred in singular in less than 10 % of their corpus occurrences.

(5) *Z chodby byly otevřené jen* ***dveře*** *do kuchyně, ostatní* ***dveře*** *byly zavřené.* 'In the corridor, only the **door** to the kitchen was open, the other **doors** were closed.'

## 2.4 Pair nouns and group nouns

It is worth noting that some nouns with the plural preference do not refer just to a larger amount of entities but prototypically to a pair or to a usual group or set of them – we speak about pair/group nouns.[6] For instance, the plural form *ruce* 'hands' means usually the pair of the upper limbs, not just a larger amount of

[6] Vossen and Copestake [16] use the term "group noun" for nouns such as *band* that refer to a group of people etc.

them, *rodiče* 'parents' are usually a pair of persons (mother and father), not several mothers and/or fathers, *vlasy* 'hair' is a (mostly not precisely quantifiable) set of hairs on one's head, *klíče* 'keys' refer to a bunch of keys. Besides the pair/group meaning the plural form of these nouns can express, without any formal change, a larger number of pairs/groups of the objects in question or simply a larger number of the objects (common plural meaning) as well.[7]

The difference between the meanings of a pair/group and several pairs/groups on the one hand and the common plural on the other becomes evident when counting the objects: in Czech the noun that refers to a pair/group (or to several pairs/ groups) is compatible only with a set numeral, while when used in the common plural meaning, the amount is expressed by a cardinal; see ex. (6) and (7). Set numerals ("souborové číslovky" in Czech terminology) are a special subtype of numerals expressing the number of pairs and other groups.[8] Set numerals are available, for instance, in Serbian and Croatian as well whereas in English or German they have no counterpart within numerals, the number of sets is then indicated by expressions such as *a pair/two pairs*. In Czech the form of numerals is one of the means for resolving the ambiguity between the common plural meaning and the pair/group meaning.

For the already mentioned nouns *ruce*, *vlasy*, *klíče*, *rodiče*, *boty*, *rukavice* and many others, the pair/group meaning is frequent, though not limited to them. It could be, according to the recent linguistic analysis based on large corpus data [12], expressed by most Czech concrete nouns. The hypothesis that the pair/group meaning is not bound up with nouns as lexical units is supported, for instance, by the unlimited co-occurrence of nouns with set numerals (ex. (8)) or by the fact that each noun which expresses the pair/group meaning in a particular context can be used in the common plural meaning (or in singular expressing singularity) in other contexts.

Therefore, we propose to consider the pair/group meaning as a semantic feature opposed to the simple quantitative meaning. Combining this distinction (set vs. simple) with the basic singular-plural opposition, four combinations are to be considered: sg.simple (singularity meaning), sg.set (meaning of a pair/group), pl.simple (common plural meaning), and pl.set (meaning of several pairs/groups). With nouns having both singular and plural forms, the meaning sg.simple is expressed by the singular, the other three meanings by the plural form. With pluralia tantum, a plural form is used for all four meanings.

(6) *Během pár týdnů jsem protančila **troje boty***. 'During a few weeks I wore through **three pairs of**

[7] Dual forms (available for the above listed nouns) are not distinguished from plural forms here since they just discern the body part meaning from the other meanings of the particular noun (e.g. the instrumental dual form *očima* for human eyes vs. instrumental plural form *oky* for loops in mesh) but have no distinctive function concerning the pair/group meaning in Czech: dual forms (just as plural forms with other nouns) refer to a pair or several pairs of the particular body part as well as to a large amount of them.

[8] In connection with a plurale tantum, a set numeral expresses the number of pieces of entities.

**shoes** by dancing.' (SYN2005)

(7) *Až doma zjistil, že mu prodali* ***dvě*** *levé* ***rukavice***. 'Only after his arrival home he found out that they had sold **two** left **gloves** to him.'

(8) *Najdeme-li* ***dvoje*** *velké* ***stopy*** *a mezi nimi* ***jedny*** *menší, řekneme si: "rodina na výletě".* 'If we find **two sets of** big **tracks** and **one set of** smaller **ones** between them, we say: "a family on a trip".' (SYN2005)

# 3 The category of number in the multi-layered annotation scenario of PDT 2.0

## 3.1 Annotation of number at the morphological layer of PDT 2.0

Formal morphological characteristics of words are described at the morphological layer of PDT 2.0. At this layer, a positional tag specifying part of speech and particular morphological categories was assigned to each token. The fourth position of the tag is reserved for the category of number. With noun forms and forms of other parts of speech that are marked for number (adjectives etc.), one of five values is to be assigned: basic values S and P with singular and plural forms, respectively, the value D with dual forms, values W and X for ambiguous cases; see [5].[9]

The preference of nouns either for singular or for plural is reflected in the morphological annotation by the assignment of plural lemmas to pluralia tantum. The set of nouns with plural lemmas roughly corresponds with lemmas marked as pluralia tantum in representative dictionaries of Czech ([6], resp. [3]); a singular form of a plurale tantum (such as *kalhota*, see Sect. 2.3) is assigned a plural lemma (*kalhoty* in the respective case) and the value S at the fourth tag position. However, neither pluralia tantum nor nouns limited to singular are marked explicitly in the morphological annotation.

## 3.2 Annotation of number at the tectogrammatical layer of PDT 2.0

The meaning of morphological categories is involved in the so-called tectogrammatical annotation of PDT 2.0, at which the (linguistic) meaning of the sentence is described as a dependency tree structure consisting of labeled nodes and edges.[10] At the tectogrammatical layer, the meaning of the category of number is encoded in the grammateme number. Grammatemes are node attributes capturing the meaning of semantically relevant morphological categories such as number and gender for nouns, degree of comparison for adjectives and adverbs, tense and aspect for verbs.[11] The grammateme number was assigned to nouns and substantival pro-

[9] With parts of speech that do not express number (e.g. prepositions), a dash (-) was filled in.

[10] Besides the morphological and tectogrammatical annotation, PDT 2.0 data were assigned also at the so-called analytical layer. At this layer, a dependency tree describing the surface-syntactic structure was assigned to each sentence.

[11] On the contrary, e.g. neither the category of case for nouns nor that of gender for adjectives were captured within the tectogrammatical annotation as they are only imposed by government or agree-

nouns and numerals (for details see [10], [13]).

Two values of the grammateme number were defined: sg and pl. Since the majority of Czech nouns express the semantic opposition of singularity and plurality directly, the values of the number grammateme could be assigned mostly automatically, mapping the number value involved in the morphological tag onto the grammateme value: the tag value S corresponds to the grammateme value sg, tag values P and D to the grammateme value pl.[12]

From the nouns described in Sect. 2.2 and 2.3, the grammateme value was assigned manually only to nouns that strongly prefer plural and use this form to refer to a single entity as well as to a larger amount of them (pluralia tantum). Since pluralia tantum were not marked explicitly in the PDT 2.0 data nor e.g. in the morphological dictionary used for tagging,[13] the manual annotation concerned nodes whose lemma was used in plural in more than 95 % lemma occurences in the PDT 2.0 data and nodes with lemmas that were marked as pluralia tantum in the Dictionary of Standard Czech Language ([6]).[14]

### 3.3 Annotating the pair/group meaning

The pair/group meaning explained in Sect. 2.4 has been introduced in the theoretical background only recently ([12])[15] and was not involved in PDT 2.0. Nevertheless, since the annotation scenario of PDT has been built on the theoretical basis of FGD, reflecting the state-of-the-art of this framework, and currently a new, both revised and extended, version of PDT (PDT 3.0) is being prepared (the revision concerns annotation of grammatical categories as well, see [11]), we are facing the question whether the pair/group meaning should be incorporated in the PDT 3.0 annotation scenario.

Before any large-scale annotation can start, it should be checked whether the phenomenon to be annotated (the pair/group meaning of plural) is reasonably frequent and practically distinguishable in the data. We performed the following pilot annotation experiment. Within 1,000 plural forms randomly selected from the SYN2005 corpus, the pair/group meaning was identified in 55 cases. If we project the same ratio on the tectogrammatically annotated sections of PDT 2.0 (which contain around 60,000 occurrences of denotative nouns in plural forms), we could

ment, respectively. Treatment of grammatical categories in FGD is closely related to the approach of Meaning–Text Theory (cf. [9]; correspondences between FGD and MTT are analyzed in [18]).

[12] Nouns, substantival pronouns and numerals with tag values W and X were assigned manually with sg or pl according to their meaning or with the value nr defined for semantically ambiguous cases. A special value inher was assigned to reflexive and relative pronouns that "inherit" the number from the coreferred node (in cases of grammatical coreference).

[13] Unlike the PDT 2.0 data and tools, information on pluralia (as well as singularia) tantum is involved, for instance, in the Croatian Morphological Lexicon [15].

[14] The lemma list obtained from PDT 2.0 data overlapped with that extracted from [6] to a large extent.

[15] In spite of the fact that this semantic feature was mentioned already in [7]. Some remarks concerning the way of expressing the pair/group meaning in Hungarian, Brazilian Portuguese, Syrian Arabic or Dutch can be found in [1].

expect roughly three thousand occurrences of the pair/group meaning. This seems to be a sufficiently high frequency: if we compare it to the frequency of functor values (i.e. dependency relations, semantic roles) annotated at the tectogrammatical layer, more than half of them does not reach this number (e.g. the functor HER for modifications with the meaning of heritage or TFRHW for modifications with the temporal meaning "from when").

Before starting the large-scale annotation, it will be further necessary to measure the inter-annotator agreement and to find ways how to automatically exclude plural forms that are not likely to have the pair/group meanings, so that the set of annotation instances is maximally reduced.

## 4 Empirical distribution of preference for plural

As we have already mentioned above, the average ratio of occurrences of singular and plural noun forms in Czech texts is 3:1. Obviously, nouns largely differ in their preferences for singular and plural. This section investigates the distribution of such preferences over the vocabulary of Czech nouns. For the purpose of this experiment, we ignore the fact that singular/plural preferences may vary across different senses of a single noun.

Let us have a function $pl(l)$ which expresses the preference of a noun lemma $l$ for plural forms simply as a relative proportion of occurrences of plural word forms among all tokens with the lemma $l$ in a given corpus (in other words, it estimates the probability of plural given the lemma).

We would like to estimate the distribution of values of $pl(l)$ across the noun vocabulary. Instead of PDT 2.0, which is too small for such estimates, we used SYN2005, which contains 100 million tokens. There are around 450,000 distinct noun lemmas in SYN2005, with around 30 million occurrences in total, out of which 7.4 million are plural forms. We divided the range of values of $pl(l)$ uniformly into 20 subintervals. We disregarded lemmas with less than 20 occurrences. Fig. 1 (a) shows the resulting histogram with the vertical axis representing the number of distinct noun lemmas having $pl(l)$ within a given subinterval.

One can immediately see two peaks in the leftmost and rightmost subintervals. The inner part of the histogram with $pl(l)$ between 0.1 to 0.9 resembles an exponential distribution. This is visually emphasized by using the logarithmic scale on the vertical axis, as the curve becomes close to linear (note that the same pattern can be seen in the distribution of English noun preferences for plural derived from the British National Corpus, see Fig. 2).

This is a striking observation. First, the distribution for values between 0.1 and 0.9 seems to be monotonous. If there were no assumptions about the $pl(l)$'s distribution across the noun vocabulary, one would expect rather a bell-shaped curve with the peak close to the average value of $pl(l)$, which is 0.16.[16] Second, the fact

[16] Recall that the histogram shows how the vocabulary of noun lemmas is partitioned with respect to their preference for plural, regardless of their total frequency in the corpus (the frequency was

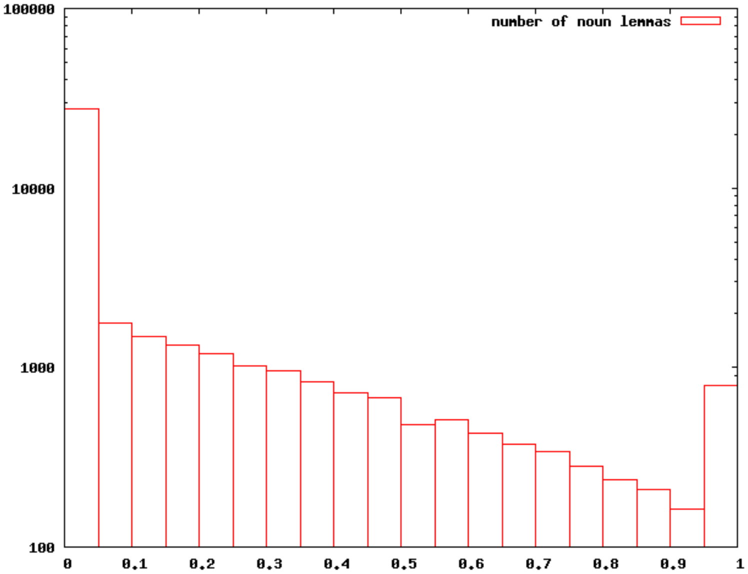


Figure 2: Histograms of noun preferences for plural according to British National Corpus.

that the distribution seems to be so regular and quite close to a perfect exponential shape suggests that there must be some relatively simple control mechanism in the language and that the distribution reflects a (dynamic) equilibrium to which this mechanism has led. The mechanism could be related to the process of lexical diversification during the language evolution.[17]

In our opinion, the key to the distribution lies in the language economy: the language tends to minimize the "energy" needed for expressing a meaning to be conveyed wherever possible. Expressing plural forms is usually more demanding than expressing singular forms. So always when a plural form of a certain noun lemma is used, the speaker might be "tempted" (not on the conscious level and not very intensively, though) to introduce a new word which could express a similar meaning by a singular form (e.g. *forest* instead of *trees*). If the new word gets spread over the population of language users, it will partially substitute the original noun and thus the original noun's $pl(l)$ will decrease. A dynamic equilibrium between this force and forces in the opposite direction (increasing the vocabulary is also costly) is reached. The fact that the distribution has an exponential form

used only for pruning infrequent nouns for which the estimate would be too unreliable). That is why the average $pl(l)$ does not correspond to the proportion of plural forms in the corpus.

[17] We do not expect the distribution to be predominantly influenced by extra-linguistic factors. It is difficult to imagine any language-independent prior distribution of singularity versus repetitiveness in the physical world around us; the distinction depends rather on how we structure our perception of the world by our language.

suggests that the equilibrium can be described by a first-order linear differential equation. However, this is only a preliminary hypothesis that should be further elaborated using laws of quantitative linguistics ([17]) and verified on other grammatical oppositions.

## 5 Conclusion

The paper is focused on the grammatical category of number of nouns within the theoretical linguistic description and the annotation of PDT 2.0. Based on large corpus data analysis, special attention has been paid to nouns with strong preferences either for singular or plural forms.

Besides the quantitative observations, the semantic opposition of singularity and plurality, which constitutes the category of number, has been refined with the distinction of the simple quantitative meaning and the pair/group meaning. The inclusion of the established opposition in the annotation scenario is not surprising and needs not to be justified whereas the involvement of the newly proposed distinction of the simple quantitative meaning and the pair/group meaning is to be carefully discussed.

## Acknowledgements

This work has been supported by the grant GA CR P406/2010/0875 of the Grant Agency of the Czech Republic and by the grants LC536 and MSM0021620838 of the Ministry of Education, Youth and Sports of the Czech Republic.

## References

[1] Corbett, G. G. (2000) *Number*. Cambridge University Press, Cambridge.

[2] Czech National Corpus - SYN2005. Institute of Czech National Corpus, Faculty of Philosophy and Arts, Charles University in Prague, Prague 2005. <http://www.korpus.cz>

[3] Filipec, J. et al. (1998) *Slovník spisovné češtiny pro školu a veřejnost*. Academia, Praha.

[4] Hajič, J. et al. (2006) *Prague Dependency Treebank 2.0*. CD-ROM, Cat. Nr. LDC2006T01. Linguistic Data Consortium, Philadelphia.

[5] Hana, J. et al. (2005) *Manual for Morphological Annotation. Revision for the Prague Dependency Treebank 2.0*. ÚFAL Technical Report No. 2005/27. ÚFAL MFF UK, Prague.

[6] Havránek, B. et al. (1989) *Slovník spisovného jazyka českého*. 2. vydání. Academia, Praha.

[7] Komárek, M. et al. (1986) *Mluvnice češtiny 2*. Academia, Praha.

[8] Mathesius, V. (1929) Funkční lingvistika. In *Sborník přednášek pronesených na Prvém sjezdu československých profesorů filosofie, filologie a historie v Praze 3.–7. dubna 1929*, pp. 118–130. Praha.

[9] Mel'čuk, I. A. (1988) *Dependency Syntax: Theory and Practice*. State University of New York Press, New York.

[10] Mikulová, M. et al. (2006) *Annotation on the tectogrammatical level in the Prague Dependency Treebank. Annotation manual*. ÚFAL Technical Report No. 2006/30. ÚFAL MFF UK, Prague.

[11] Panevová, J. and Ševčíková, M. (2010) Annotation of Morphological Meanings of Verbs Revisited. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pp. 1491–1498. ELRA, Paris.

[12] Panevová, J. and Ševčíková, M. (in prep.) Počítání substantiv v češtině (Poznámky ke kategorii čísla). To appear in *Slovo a slovesnost*.

[13] Razímová, M. and Žabokrtský, Z. (2006) Annotation of Grammatemes in the Prague Dependency Treebank 2.0. In *Proceedings of the LREC 2006 Workshop on Annotation Science*, pp. 12–19. ELRA, Paris.

[14] Sgall, P., Hajičová, E. and Panevová, J. (1986) *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. D. Reidel Publishing Company, Dordrecht.

[15] Tadić, M. and Fulgosi, S. (2003) Building the Croatian Morphological Lexicon. In *Proceedings of the EACL2003 Workshop on Morphological Processing of Slavic Languages*, pp. 41–46. ACL, Budapest.

[16] Vossen, P. and Copestake, A. (1993) Untagling Definition Structure into Knowledge Representation. In Briscoe, T., de Paira, V and Copestake, A. (eds.) *Inheritance, Defaults and the Lexicon*, pp. 246–274. Cambridge University Press, Cambridge.

[17] Wimmer, G. and Altmann, G. (2005) Unified derivation of some linguistic laws. In *Quantitative Linguistics*, pp. 791–807. De Gruyter, Berlin & New York.

[18] Žabokrtský, Z. (2005) Resemblances between Meaning–Text Theory and Functional Generative Description. In *Proceedings of the 2nd International Conference of Meaning–Text Theory*, pp. 549–557. Slavic Culture Languages Publishers House, Moscow.

# On the Derivation Perplexity of Treebanks

Anders Søgaard

Center for Language Technology
University of Copenhagen
Njalsgade 142
DK-2300 Copenhagen S
Email: soegaard@hum.ku.dk

Martin Haulrich

ISV Computational Linguistics Group
Copenhagen Business School
Dalgas Have 6
DK-2000 Frederiksberg
Email: mwh.isv@cbs.dk

**Abstract**

Parsing performance is typically assumed to correlate with treebank size and morphological complexity [6, 13]. This paper shows that there is a strong correlation between derivation perplexity and performance across morphologically rich and poor languages. Since perplexity is orthogonal to morphological complexity, this questions the importance of morphological complexity. We also show that derivation perplexity can be used to evaluate parsers. The main advantage of derivation perplexity as an evaluation metric is that it measures global aspects of parsers (like counting exact matches), but is still fine-grained enough to derive significant results on small standard test sets (like attachment scores).

## 1 Introduction

State-of-the-art accuracy on a particular parsing dataset is typically assumed to correlate with treebank size and morphological complexity of the language in question [6, 13]. Out-of-domain evaluation also shows that parsers are typically very domain-sensitive. For example, as shown in the CoNLL 2007 shared task, parsers trained on the Penn Treebank are much better at parsing the Wall Street Journal than at parsing biomedical articles or transcribed speech. This of course relates to perplexity, since out-of-domain text is much less predictable for language models.

However, no one has to the best of our knowledge used perplexity as a metric to better evaluate parsers and treebanks. This paper suggests some ways to do so and presents some preliminary experiments.

The easiest way to explain perplexity is perhaps by considering the case of a fair $k$-sided die. The perplexity of such a die is $k$, meaning that we are $k$-ways perplexed about the outcome of the die. If the die is unfair and always end with the same side up, the perplexity is 1, meaning that we are certain about the outcome of the die.

Given a language model **lm** trained on a corpus (of sentences or derivation orders), we are interested in how well it predicts a sample of test instances $x_1, \ldots, x_N$. The perplexity is defined as:

$$2^{-\Sigma_{i=1}^{N} \frac{1}{N} \log_2 \mathbf{lm}(x_i)}$$

Better language models will tend to assign higher probabilities to the instances in our sample and will thus have lower perplexity. In our experiments in this paper, we use standard trigram language models with modified Kneser-Ney smoothing and interpolation.

Dependency treebanks are collections of dependency trees. A dependency tree is a tree that represents a syntactic analysis such that words are vertices with various labels and grammatical functions label the directed edges (dependencies). Each word thus has a single incoming edge, except one called the root of the tree. Dependency parsing is thus a structured prediction problem with trees as structured variables. Each sentence has exponentially many possible dependency trees. The observed variables are typically sentences with words labeled with part-of-speech tags. The parsing task for each sentence is to find the dependency tree that maximizes an objective function which is typically learned from a dependency treebank.

The standard metrics used in dependency parsing are labeled attachment score (LAS), i.e. the ratio of words with correct syntactic heads and grammatical functions, unlabeled attachment score (UAS), i.e. the ratio of words with correct syntactic heads, and exact matches (EM), i.e. the ratio of sentences in which all words are assigned correct syntactic heads. The disadvantage of LAS and UAS is that attachment scores do not reflect global properties of the predicted syntactic analyses, while EM has the disadvantage that differences are seldom statistically significant on small evaluation data sets. This paper suggests that perplexity of derivation order may also be a useful metric for parser evaluation. It is not a stand-alone metric, but used in conjunction with LAS or UAS it may supply the global information that EM is supposed to reflect.

Perplexity of derivation order may also be used to predict state-of-the-art accuracy on treebanks and thereby indirectly for treebank evaluation. State-of-the-art parsing performance is typically assumed to correlate with treebank size and morphological complexity, but in this paper we show that there is a strong correlation between perplexity of derivation order and parsing performance across morphologically rich and poor languages.

## 1.1 Related Work

Nivre [6] presents an analysis of the CoNLL 2007 shared task, building on [9], and draws the conclusion that state-of-the-art parsing accuracy primarily depends on morphological complexity.

> The ten languages involved in the multilingual track can be grouped into three classes with respect to the best parsing accuracy achieved:
>
> - Low (LAS = 76.3-76.9): Arabic, Basque, Greek
> - Medium (LAS = 79.2-80.2): Czech, Hungarian, Turkish
> - High (LAS = 84.4-89.6): Catalan, Chinese, English, Italian
>
> To a large extent, these classes appear to be definable from typological properties. The class with the highest top scores contains languages with a rather impoverished morphology. Medium scores are reached by the two agglutinative languages, Hungarian and Turkish, as well as by Czech. The most difficult languages are those that combine a relatively free word order with a high degree of inflection. Based on these characteristics, one would expect to find Czech in the last class. However, the Czech training set is four times the size of the training set for Arabic, which is the language with the largest training set of the difficult languages. On the whole, however, training set size alone is a poor predictor of parsing accuracy, which can be seen from the fact that the Italian training set is only about half the size of the Arabic one and only one sixth of Czech one. Thus, there seems to be a need for parsing methods that can cope better with richly inflected languages.

The same conclusion was the motivation for a workshop in Statistical Parsing of Morphologically Rich Languages at NAACL'10 in Los Angeles, California [13]. In this paper, we show that, not surprisingly, there is a strong correlation between treebank size and state-of-the-art accuracy, but also that a stronger correlation exists between relative derivation perplexity and state-of-the-art accuracy, even across morphologically rich and poor languages.

Evaluation of parsers has been widely debated in recent years. Caroll et al. [2] review the parsing evaluation metrics that were available at the time and propose a new one. They first discuss a number of metrics that can be used with unannotated corpora, incl. coverage, average ambiguity and perplexity. The problem with coverage and average ambiguity is that the metrics do not say anything about accuracy. The perplexity of a parsing model on a corpus may under certain assumptions tell us about the accuracy of the model or about the "degree to which a model captures regularities in the corpus by minimising unpredictability and ambiguity". The advantages of this metric are that it has a clear probabilistic interpretation, allows meaningful comparison and can be used "as a method for scaling results obtained using other corpus-dependent measures to allow for some degree of cross-corpus comparison and evaluation." The disadvantages are that the metric is "expensive to compute", "only applicable to probabilistic models" and that it "only provides a weak measure of accuracy." The notion of derivation perplexity introduced here differs from perplexity of a probabilistic parsing model in that it can be read off structures immediately. We can therefore talk about the derivation perplexity of parsers as well as treebanks. Consequently, it is *not* expensive to compute, it is applicable to non-probabilistic models (such as transition-based dependency parsers), and we also show that it correlates strongly with stronger measures of accuracy.

Rimell et al. [11] suggest a new evaluation scenario for dependency parsing of English text. They construct a corpus of 700 English unbounded dependency constructions. They argue:

> These are interesting for parser evaluation for the following reasons: one, they provide a strong test of the parser's knowledge of the grammar of the language, since many instances of unbounded dependencies are difficult to recover using shallow techniques in which the grammar is only superficially represented; and two, recovering these dependencies is necessary to completely represent the underlying predicate-argument structure of a sentence, useful for applications such as Question Answering and Information Extraction.

One problem with this approach, noted already by [2], is that the usefulness of such a corpus depends heavily on what constructions are included. It is certainly not trivial to distinguish between important and less important constructions. It would be interesting to see how retrieval of unbounded dependencies in this corpus correlates with other metrics and pipeline evaluations. Another problem is of course that the metric is language-dependent. The advantage, however, is that retrieval of unbounded dependencies in most parsers depends heavily on the rest of the analysis and thus can be said to capture global aspects of the syntactic analysis.

Finally, we note that many researchers have proposed pipeline evaluation of parsers where parsers are evaluated in terms of their contribution to a particular application, e.g. machine translation [4] or textual entailment [14].

## 2 Dependency Treebanks

Dependency treebanks have become increasingly popular over the last five years. With the development of fast, reliable dependency parsers [5, 10], theoretically motivated dependency treebanks such as the 1M word Prague Dependency Treebank and two competitive shared tasks in dependency parsing at the Conferences on Natural Language Learning (CoNLL) in 2006–7, large scale evaluation of dependency parsers and pipeline evaluation in natural language processing applications have become possible. Dependency parsers have among other things been used for summarization and machine translation [4].

More formally, a dependency tree for a sentence $x = w_1, \ldots, w_n$ is a tree $T = \langle\{0, 1, \ldots, n\}, A\rangle$ with $A \subseteq V \times V$ the set of dependency arcs. Each vertex corresponds to a word in the sentence, except 0 which is the root vertex, i.e. for any $i \leq n$ $\langle i, 0\rangle \notin A$. Since a dependency tree is a tree it is acyclic. A tree is projective if every vertex has a continuous projection, i.e. if and only if for every arc $\langle i, j\rangle \in A$ and node $k \in V$, if $i < k < j$ or $j < k < i$ then there is a subset of arcs $\{\langle i, i_1\rangle, \langle i_1, i_2\rangle, \ldots, \langle i_{k-1}, i_k\rangle\} \in A$ such that $i_k = k$.

Characteristics of a large portion of the available dependency treebanks can be found in the CoNLL shared task organizers' papers [1, 9], but several other dependency treebanks now exist, incl. treebanks for Ancient Greek, Latin, Romanian and Thai. The treebanks differ in size and domain dependence, and they adopt different annotation guidelines. Different annotation guidelines sometimes complicate translation-oriented applications, and parallel dependency treebanks are therefore also being developed.

In our experiments, we use the treebanks from the CoNLL-X and CoNLL 2007 shared tasks.

## 3 Derivation Perplexity

A transition-based dependency parser $p$'s derivation perplexity on a text $T$ is defined as the perplexity of the derivation language of $p(T)$, where $p(T)$ is the 1-best parse trees of the sentences in $T$: The *derivation language* of $p(T)$ is the set of strings $\sigma : w_1 \ldots w_n$ such that for any $w_i, w_j$ with dependency structure $d$ if $w_i \prec w_j$ then $w_i$ was attached to $d$ in $p(T)$ prior to the attachment of $w_j$.

The derivation perplexity of a treebank $R$ over a text $T$ is the derivation perplexity of $f(T)$ where $f$ is a function from the strings in $T$ into the canonical parse of the corresponding trees in $R$ given some parsing algorithm. In this paper, the parsing algorithm used to obtain canonical parses will be the so-called Swap-Lazy algorithm introduced in [8].

The Swap-Lazy algorithm was chosen because it is a non-projective dependency parsing algorithm (and many of the treebanks used in our experiments contain non-projective dependencies) and because, as documented in [8], it has higher accuracy in terms of exact matches than other state-of-the-art transition-based dependency parsing algorithms.

The algorithm works as follows: We begin with a configuration (Stack, Buffer, Arcs) where Stack is a stack that initially only contains an artificial root element, Buffer is the string to be read, and Arcs is the dependency structure to be build (a set of dependency arcs of the form $(w_i, w_j)$). The initial configuration is thus $([w_0]_S, [w_1, \ldots, w_n]_B, \{\}_A)$ with $w_1 \ldots w_n$ the input sentence and where $w_0$ is the artificial root note in the dependency tree. The transition algorithm halts when a final configuration is reached of the form $([w_0]_S, []_B, A)$, i.e. when all words are read and removed from the stack. Below we only consider unlabeled parsing. Otherwise different Right-Arc and Left-Arc transitions must be introduced for each label. The possible transitions are:

$$\begin{aligned}
\textbf{Shift}\quad & ([\ldots, w_i]_S, [w_j, \ldots]_B, A)) \Longrightarrow ([\ldots, w_i, w_j]_S, [\ldots]_B, A) \\
\textbf{Right-Arc}\quad & ([\ldots, w_i, w_j]_S, B, A)) \Longrightarrow ([\ldots, w_i]_S, B, A \cup \{(w_i, w_j)\}) \\
\textbf{Left-Arc}\quad & [i \neq 0]\ ([\ldots, w_i, w_j]_S, B, A)) \Longrightarrow ([\ldots, w_j]_S, B, A \cup \{(w_j, w_i)\}) \\
\textbf{Swap}\quad & [0 < i < j]\ ([\ldots, w_i, w_j]_S, [\ldots]_B, A)) \Longrightarrow \\
& \qquad\qquad ([\ldots, w_j]_S, [w_i, \ldots]_B, A \cup \{(w_j, w_i)\})
\end{aligned}$$

Intuitively, Shift moves an element from the Buffer to the Stack. Right-Arc builds a dependency arc from the second element (the left word) to the top element. (The dependency arcs build using this transition therefore point to the right; hence, the name.) Left-Arc builds a dependency arc from the top element to the second element, i.e., left arcs. Swap, finally, reorders words by moving the second element on the stack back into the buffer. Consequently, Right-Arc and Left-Arc may build left arcs, resp. right arcs, relative to the original linear order of words. In other words, the intuition behind this parsing algorithm is to reduce discontinuity to adjacency by reordering input words.

Given a dependency structure, there is a canonical derivation of it using the Swap-Lazy algorithm. The derivation sequences constructed from the trees in a treebank are used to train the classifiers that decide which transition to apply in a particular configuration when parsing with the MaltParser [9, 8]. Since each derivation step corresponds to finding a syntactic head for a word, we use the derivation order as a linear reordering simply by printing the words in the order they are attached to the dependency structures. Put differently, a derivation will gradually expand the set of Arcs. For a derivation $([w_0]_s], [w_1, \ldots, w_n]_B, \{\}_A) \Longrightarrow^*$ $([w_0]_S, []_B, A)$ there is a linear order $\prec$ such that for any $w_i, w_j$ such that $w_i \prec w_j$, $w_i$ was added to $A$ before $w_j$. It is this linear order whose perplexity is computed in our experiments. In our first experiment, we use the dependency trees from treebanks. In our second experiment, we use the output from four different transition-based dependency parsers.

# 4 Experiments

## 4.1 Data

Our experiments cluster the treebanks in three groups: the treebanks used in the CoNLL-X Shared Task (excl. Chinese, which was not available to us), those used in the CoNLL 2007 Shared Task, and the treebanks that were used in both shared tasks and are genuine *dependency treebanks*, i.e. not converted constituent-based treebanks. Using only genuine dependency treebanks have become standard, e.g. [7], when parsing performance is evaluated in terms of exact matches. The third set of treebanks excludes very large treebanks (>200k tokens) and Greek, which our language modeling software (SRI Language Modeling Toolkit [12]) did not process correctly. The CoNLL-X treebanks is thus a set of 12 treebanks, the CoNLL 2007 treebanks a set of 10 treebanks, and the genuine dependency treebanks is a set of 7 treebanks. In sum, the three clusters of treebanks are as follows:

| name | number | languages |
|---|---|---|
| C06 | 12 | Arabic, Bulgarian, Czech, Danish, Dutch German, Japanese, Portuguese, Slovene, Spanish, Swedish, Turkish |
| C07 | 10 | Arabic, Basque, Catalan, Chinese, Czech, English, Greek, Hungarian, Italian, Turkish |
| gen.dt | 7 | Arabic(C06), Arabic(C07), Czech, Danish, Slovene, Turkish(C06), Turkish(C07) |

### 4.2 Language Model Parameters

We use a 3-gram language model with modified Kneser-Ney smoothing and interpolation as implemented in the freely available SRI Language Modeling Toolkit [12]. Kneser-Ney smoothing was found to consistently outperform other smoothing techniques in [3].

### 4.3 Perplexity and State-of-the-Art Accuracy

This experiment was designed to quantify to what extent state-of-the-art parsing accuracy can be predicted from the derivation perplexity of a treebank. For the experiment, we reimplemented the Swap-Lazy algorithm for training the oracle in MaltParser [8] and printed out words in the order of derivation (attachment). Briefly put, the Swap-Lazy algorithm can keep words on the buffer or place them on the stack, but at some point it will attach words to the dependency structure being build, and the words are simultaneously removed from the stack. It is at this point that the word is printed. The result is a linear reordering of the input text that corresponds to the derivation order. The perplexity of this derivation order is computed by training a language model on the reordered training data and running it on the reordered test data. The language model parameters are described above.

What is correlated with state-of-the-art accuracy is not string perplexity and derivation order perplexity, but treebank size over these perplexities. In particular, we correlate (i) treebank size with accuracy (as our baseline), (ii) treebank size over string perplexity with accuracy and (iii) treebank size over derivation order perplexity with accuracy.

The experiment was done for the CoNLL-X shared task treebanks, as well as for the CoNLL 2007 treebanks. Treebank sizes over perplexities are then correlated with state-of-the-art results, i.e. the best results obtained in the shared tasks. We also report correlations with the average results of the shared task participants. This may in fact be a more interesting measure for treebank evaluation, since a treebank where 10 participants achieve an UAS $> 90\%$ is probably "easier" than one where only one participant does so, even if the best scores are identical. It is, however, more common to focus on the best results obtained in the shared tasks or in the literature [6, 9, 13].

### 4.4 Perplexity as a Metric for Parser Evaluation

We ran our parsing evaluation experiments on the genuine dependency treebanks. This is a common practice used, for example, in [7] and [8]. We ran the MaltParser [10] with four different parsing algorithms (Arc-Eager, Arc-Standard, Swap-Eager and Swap-Lazy) and default feature settings to obtain four different outputs on each of the CoNLL test sections. UAS, EM and perplexity of derivation order were computed, and Pearson $\rho$ was calculated from these numbers. The average Pearson $\rho$ which is reported below, is the average Pearson $\rho$ for the seven treebanks.

## 5 Results

We first list the average perplexities and derivation perplexities of the treebanks.

| name | av. perplexity | av. deriv. perpl. | increase |
|---|---|---|---|
| C06 | 292.5 | 532.8 | 82.2% |
| C07 | 320.3 | 508.8 | 37.0% |
| gen.dt | 278.2 | 447.8 | 96.6% |

Note the higher increase from perplexity to derivation perplexity with genuine dependency treebanks.

### 5.1 Perplexity and State-of-the-Art Accuracy

The correlation coefficients for state-of-the-art parse accuracy and treebank size over string/derivation perplexity are presented in the table below.

| Pearson $\rho$ | C06 | | C07 | |
|---|---|---|---|---|
| | best | av | best | av |
| treebank size | 0.21245816 | 0.037759424 | 0.72245934 | 0.55952737 |
| string perplexity | **0.47495902** | 0.506099378 | 0.643438377 | 0.548117203 |
| deriv perplexity | 0.47015543 | **0.514647899** | **0.810661115** | **0.737857396** |

The results indicate that (treebank size over) derivation order perplexity is much better at predicting state-of-the-art parsing accuracy than treebank size only. While there is a strong correlation between (treebank size over) string perplexity and accuracy, the notion of derivation order perplexity seems more relevant than mere string perplexity.

Since the correlation between derivation perplexity and accuracy cuts across morphological complexity, and since morphological complexity is orthogonal to perplexity, this questions the importance of morphological complexity to parsing performance. Morphologically poor languages such as Chinese typically lead to very high perplexities (in our case $\sim$900), while morphologically rich languages such as Turkish typically exhibit moderate perplexities (in our case $\sim$120).

### 5.2 Perplexity as a Metric for Parser Evaluation

We only ran our parsing evaluation experiments on the seven genuine dependency treebanks. We ran the MaltParser [10] with four different parsing algorithms (Arc-Eager, Arc-Standard, Swap-Eager and Swap-Lazy) and default feature settings to obtain four different outputs on each of the CoNLL test sections, thus running a total of 28 dependency parsers. We computed the Pearson $\rho$ correlations between UAS, EM and perplexity of derivation order for each treebank and averaged over these numbers. Perplexity of derivation order (P) correlates as well with UAS and EM as they correlate internally. All correlations were significant ($p < 0.05$), where significance is derived from $\rho$.

| | $\rho$ | $p$-value |
|---|---|---|
| P/UAS | **-0.5670** | 0.0172 |
| P/EM | -0.5464 | 0.0217 |
| UAS/EM | 0.5510 | 0.0206 |

Of course this result only says that the three metrics are correlated, but not which of the three is more useful. Since they capture different aspects and have different weaknesses, as argued above, we suggest to use all three metrics in liaison. It would again be interesting to correlate these metrics with pipeline evaluations.

## 6 Conclusion

We have introduced the notion of derivation order perplexity which is much better at predicting state-of-the-art parsing accuracy than treebank size only. It was shown that there is a strong correlation between state-of-the-art parsing accuracy and derivation order perplexity across morphologically poor and rich languages. Derivation order perplexity can also be used as a metric for parser evaluation and has the advantages that it captures global aspects of syntactic analyses and is fine-grained enough to obtain statistically significant results on small data sets.

## References

[1] Sabine Buchholz and Erwin Marsi. CoNLL-X shared task on multilingual dependency parsing. In *CoNLL*, 2006.

[2] John Caroll, Ted Briscoe, and Antonio Sanfilippo. Parser evaluation: a survey and a new proposal. In *LREC*, 1998.

[3] Stanley Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University, 1998.

[4] Michel Galley and Christopher Manning. Quadratic-time dependency parsing for machine translation. In *ACL*, Singapore, 2009.

[5] Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Nonprojective dependency parsing using spanning tree algorithms. In *HLT-EMNLP*, 2005.

[6] Joakim Nivre. Data-driven dependency parsing across languages and domains: perspectives from the CoNLL 2007 shared task. In *IWPT*, 2007.

[7] Joakim Nivre. Non-projective dependency parsing in expected linear time. In *ACL-IJCNLP*, 2009.

[8] Joakim Nivre. An improved oracle for dependency parsing with online reordering. In *Proceedings of the 11th international conference on parsing technologies*, pages 73–76, Paris, France, 2009.

[9] Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The CoNLL 2007 shared task on dependency parsing. In *EMNLP-CoNLL'07*, pages 915–932, Prague, Czech Republic, 2007.

[10] Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. MaltParser: a language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135, 2007.

[11] Laura Rimell, Stephen Clark, and Mark Steedman. Unbounded dependency recovery for parser evaluation. In *EMNLP*, Singapore, Singapore, 2009.

[12] Andreas Stolcke. SRILM – an extensible language modeling toolkit. In *Intl. Conf. on Spoken Language Processing*, 2002.

[13] Reut Tsarfaty, Djame Seddah, Yoav Goldberg, Sandra Kuebler, Yannick Versley, Marie Candito, Jennifer Foster, Ines Rehbein, and Lamia Tounsi. Statistical parsing of morphologically rich languages. In *The 1st Workshop on Statistical Parsing of Morphologically Rich Languages, NAACL*, 2010.

[14] Deniz Yuret, Aydin Han, and Zehra Turgut. SemEval-2010 Task 12: parser evaluation using textual entailments. In *SemEval*, 2010.

# A Syntax-first Approach to High-quality Morphological Analysis and Lemma Disambiguation for the TüBa-D/Z Treebank

Yannick Versley, Kathrin Beck, Erhard Hinrichs, Heike Telljohann

Seminar für Sprachwissenschaft
University of Tübingen
E-mail: versley|kbeck|eh|telljohann@sfs.uni-tuebingen.de

**Abstract**

Morphological analyses and lemma information are an important auxiliary resource for any treebank, especially for morphologically rich languages since such information is a useful precondition for any task that needs to link surface forms to semantic interpretation (either through wordnets or distributional measures).

In contrast to common practice in parsing, the method used in the TüBa-D/Z treebank uses syntactic information for the morphological and lemma disambiguation. We argue that this approach has an advantage in the context of treebanking since many ambiguities in morphology and lemmas can be eliminated given the syntactic context.

## 1 Introduction

To use lexical resources, such as wordnets (e.g., Princeton WordNet, Miller and Fellbaum, 1991; GermaNet, Kunze and Lemnitzer, 2002) in conjunction with corpora, it is necessary to map surface word forms to lemmas (or dictionary forms). Princeton WordNet offers its own lemmatizer (formulated in a dozen rules and a list of exceptions – about 6 000 in total in Princeton WordNet 3.0). For languages with richer inflection, such as German, tools for morphological analysis are considerably more complex, yet the problem of linking surface forms to the entries in lexical resources remain.

Some researchers, such as Gurevych and Niederlich (2005) solve this problem by using stemming, but remark that, contrary to their expectations, stemming delivered no better results than no morphological processing at all.

One way to relieve this problem is to annotate corpora – in particular, when they already include a multitude of annotation levels – with gold-standard lemma information, which allows researchers to perform reproducible experiments linking corpora and lexical resources, without any concerns about lemmatization errors.

In the following sections, we will describe the existing annotation in the TüBa-D/Z (section 2), the system that we used to do high-quality pre-tagging of morphology and lemma information for the manual disambiguation (section 3), and provide some statistics on both the automatic and manual annotation (section 4).

## 2 The TüBa-D/Z

The TüBa-D/Z treebank of German[1] is a linguistically annotated corpus based on data from the German newspaper '*die tageszeitung*' (taz). The current Release 5 comprises approximately 45 000 sentences, with a new release of the treebank consisting of more than 55 000 sentences, including lemma information, to be be released before the end of 2010.

The annotation scheme of the TüBa-D/Z treebank comprises four levels of syntactic annotation: the lexical level, the phrasal level, the level of topological fields, and the clausal level. The primary ordering principle of a clause is the inventory of topological fields, which characterize the word order regularities among different clause types of German, and which are widely accepted among descriptive linguists of German (cf. Drach, 1937; Höhle, 1986). Below this level of annotation, i.e. strictly within the bounds of topological fields, a phrase level of predicate-argument structure is applied with its own descriptive inventory based on a minimal set of assumptions that has to be captured by any syntactic theory. A set of node labels describes the syntactic categories (including topological fields and coordinations). The context-free backbone of phrase structure (i.e. proper trees without crossing branches; Telljohann et al., 2004) is combined with edge labels specifying the grammatical functions of the phrases in question as well as long-distance relations. Phrase internal dependencies are captured by a hierarchical annotation of constituent structure with head/non-head distinctions. For more details on the annotation scheme see Telljohann et al. (2009).

Over the course of the last years, the syntactic annotation has been extended in various ways. Named entity information has been added. The basic Stuttgart-Tübingen tagset (STTS: Schiller et al., 1995) labels have been enriched by relevant features of inflectional morphology. A set of anaphoric and coreference relations referring to nominal and pronominal antecedents has been incorporated to link referentially dependent noun phrases (Hinrichs et al., 2004). Current work comprises both annotating new sentences as well as adding lemmas for each word form.

(1)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| *Wenn* | *alles* | *nach* | *Plan* | *läuft,* | *werden* | *sie* | *die* | *ersten* |
| If | everything | to | plan | goes, | will | they | the | first |

| | |
|---|---|
| *Umzugsbotschafter* | *sein.* |
| dislocation=ambassadors | be. |

If everything goes according to plan, they will be the first 'dislocation ambassadors'.

[1] For more information, see `http://www.sfs.uni-tuebingen.de/tuebadz.shtml`

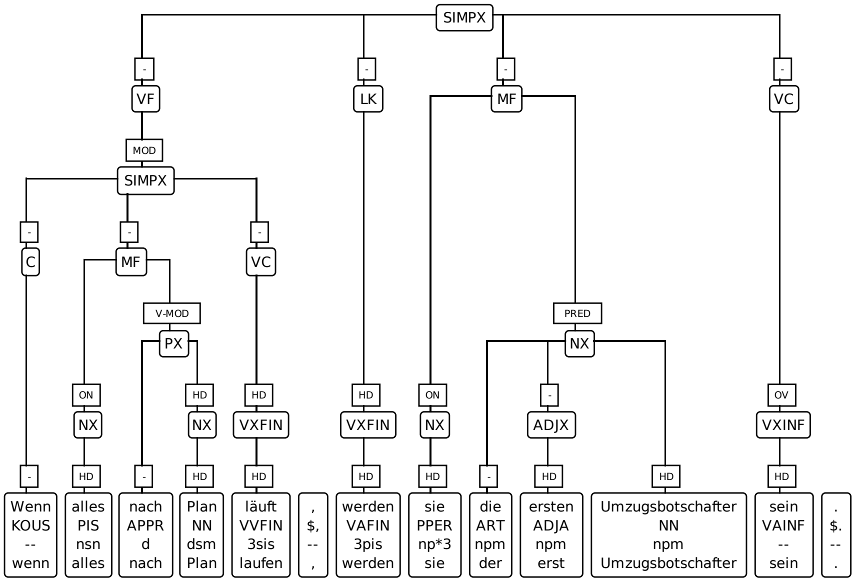


Figure 1: Example tree from the TüBa-D/Z

Figure 1 illustrates the linguistic annotation for the sentence in (1). The tree exemplifies the syntactic annotation scheme. The main clause (SIMPX) is divided into four topological fields: initial field (VF), left sentence bracket (LK), middle field (MF), and verb complex (VC). The finite verb in LK is the head (HD) of the sentence. The edge labels between the level of topological fields and the phrasal level constitute the grammatical function of the respective phrase: subject (ON), predicate (PRED), modifier (MOD), modifier of the verb (V-MOD). The modifying subordinate clause (SIMPX) in the initial field is again divided into the following fields: c-field (C), MF, and VC. The label V-MOD specifies the long-distance dependency of the prepositional phrase (PX) "*nach Plan*" on the main verb "*läuft*". Below the lexical level, the parts of speech, the morphological information, and the lemmata are annotated.

The presence of multiple layers of annotation has made it possible to use the TüBa-D/Z corpus in comparative evaluations for tasks including parsing and anaphora/coreference; the additional layers of annotation also make it possible to evaluate the impact of gold-standard versus automatic morphological information in these tasks.

# 3 Semi-automatic Morphological Tagging and Lemmatization

The annotation of the TüBa-D/Z corpus is carried out serially for different layers, beginning with the syntactic annotation (including part-of-speech, topological fields, and basic named entity annotation) and proceeding to morphological and lemma annotation as well as the anaphora/coreference annotation.

The annotation of syntactic structure prior to morphological annotation may appear unconventional since incremental annotation usually proceed from smaller units to larger units - thus annotation of part-of-speech and morphology typically precedes syntactic annotation. However, there are good reasons for adopting a syntax-first approach when it comes to the construction of a large treebank for a morphologically rich language.

A substantial part of the ambiguities that would occur in morphological tagging, especially with respect to case, are resolvable using syntax annotation; Furthermore, the integration of morphology only occurs after the assignment of syntactic structure, which means that we can profit from the information that has already been hand-corrected, feeding corrected syntactic information into the morphological disambiguation, and corrected morphological information into the disambiguation of lemmas.

## 3.1 Morphological Tagging

Morphological tags have been present in the TüBa-D/Z treebank already since the second release (Hinrichs et al., 2004); hence, the syntax-first version of the pre-tagging produces morphological tags according to the existing guidelines, but achieves greater accuracy thanks to the annotated syntactic information. Per-token analysis is based on SMOR (Schmid et al., 2004), a finite-state analyzer including derivational morphology, as well as additional heuristics that help in dealing with names and unknown words. The analyses assigned by SMOR and the heuristics are disambiguated both locally (within a noun phrase) and globally (using argument and modification information, enforcing consistent readings across coordination).

Since proper names contain morphological information (names of persons according to gender, the grammatical gender of locations is usually neutral, whereas the grammatical gender of organizations is best predicted by their organizational suffix – such as GmbH, AG, etc.), prediction of the morphology of named entities is done on the one hand by considering previous morphological tags assigned to this name string, and on the other hand by consulting gazetteer lists.

For certain classes, such as (invariant and regular) adjectives, simple suffix analysis is sufficient to predict the possible morphological tags. For nouns, which may be highly irregular, a maximally underspecified morphological tag is used so that the surrounding context may partially disambiguate them.

The first, local disambiguation step consists in disambiguating morphological tags within a single base NP: with very few exceptions, head and modifiers of a noun phrase share the same morphological tag. Additional disambiguation is performed on strong and weak inflections of adjectives: the so-called *weak inflections* occur with definite and most indefinite articles, whereas *strong inflections* only occur when no article is present or the determiner is an ending-less form (indefinite nominative masculine/neutral determiners such as *ein*, *kein*, *sein* etc.)

The following steps in morphological disambiguation make use of the syntactic annotation that is already present in a more extensive fashion. For disambiguating case, which would be error-prone in a system based on sequence labeling, we have exact information from the grammatical functions at our disposition: Subject and copula predicates universally occur in nominative case, accusative objects in accusative case; similarly, pre- or postpositions govern the case of the noun phrase in the corresponding PP.[2] We make this information explicit by projecting down the case information across any adjunction or coordination nodes, so that the base NP with the head receives case annotation. Finally, we can also enforce number and person agreement between the inflected verb and the subject, as well as reflexives and the predicate complements of copula sentences.

Prepositions that allow both accusative and dative case (corresponding to a directional and a locational sense, similar to English *into* and *in*) are disambiguated by assuming that PPs attached to a noun are either locative (i.e., dative case) or an argument to that noun, a case for which the combination of governing noun and preposition is checked against a list acquired from a large unannotated corpus (where unambiguous combinations of a noun, a preposition, and an accusative NP are seen as indicative that the noun-preposition combination plausibly occurs with an accusative PP).

### 3.2 Lemmatization: open-class words

In the case of content words, the purpose of lemmatization is to map differently inflected forms of the same stem into a single form which helps to find the corresponding entries in (paper) dictionaries, wordnets, or other electronic resources as well as obtaining accurate counts to compare the relative frequency of nouns irrespective of their inflection.

There are several choices to be made with respect to lemmatization of German. For open-class words, we aimed for maximal consistency with the lemmatization in GermaNet; in particular, deadjectival nouns (such as '*Arbeitsloser*' [jobless person]: consider strong '*ein Arbeitsloser*' [*a* jobless person] versus weak '*der Arbeitslose*' [*the* jobless person]), and nouns with a corresponding inflection (such as '*Beamter*' [civil servant], which follows the weak/strong distinction normally found in adjectives and deadjectival nouns), are lemmatized to the *strong* forms.

The syntax-first strategy also provides valuable linguistic information in cases where lemmatization would be ambiguous if the lexical token is considered in isolation and not in its syntactic context: In the example below, *Summen* is ambiguous

[2] Many prepositions allow both accusative and dative case, in which case further disambiguation is necessary whenever the NP chunk is case-ambiguous.

between a singular analysis as *Summen* (humming) and a plural analysis of *Summe* (sum). Subject-verb agreement constraints in morphological disambiguation yield the necessary information to remove this ambiguity, which would still be present if only local disambiguation had been applied.

(2) *"Da hätten Summen von 165.000 Mark schon auffallen*
"There have.IRR sums of 165 000 Mark already be_conspicuous
*müssen".*
must".

"In such a situation, sums of 165 000 Mark should have been conspicious".

Verb particles are attached to the verb to which they belong syntactically. Furthermore, verbs such as *haben* (to have), *sein* (to be) and *werden* (to become) have uses as a main verb (as a verb of possession, or as copula verbs, respectively) and as an auxiliary, which are not distinguished in the part-of-speech tags according to the STTS guidelines. To help in the identification of main-verb uses of these verbs, the lemmas of word tokens used as auxiliary are suffixed with an additional tag (%passiv for passive constructions and %aux for other non-main-verb uses of auxiliaries and modals).

Again, the syntax-first strategy makes it possible to provide such a fine-grained lemma analysis of these items. This in turn allows users to perform searches for full verb uses of auxiliaries or constructions such as the passive.

The lemmatization also distinguishes between separable and inseparable verb prefixes (which can be helpful in cases where both separable and inseparable versions are possible, since the meanings of these versions are generally distinct from each other) by putting a # marker between a separable verb prefix (reattached or not) and the verb. To make this distinction in cases where SMOR returns ambiguous analyses (for example, *unter-* can be used both as a separable and as an inseparable verb prefix, as in *unter#buttern* – to ride roughshod over someone, and *untermauern* – to underpin). However, most verbs only allow, or have a strong preference for, only one of these possibilities. As a result, disambiguation is possible in most cases using frequency data[3] for unambiguous forms (in this case, the *zu*-infinitive form, which would be *unterzubuttern* in the separable case and *zu untermauern* in the inseparable case).

Reconstruction of verb and adjective lemmas from SMOR's analysis is normally possible by transforming the FST analysis. For nouns, in contrast, this is not generally possible, since the SMOR analysis is less informative than the original string and omits information about linking elements ('*Fugenelement*') in compounds, which may be semantically significant (for example, consider *Kindsmutter* – a child's mother, to *Kindermutter* – a nanny, which both get the same analysis consisting of their two stems *Kind+Mutter*).

To get around this weakness regarding linking elements, we adopt a regeneration approach similar to the one used by Versley (2007) for generating plural forms:

[3] The frequency data is extracted from the *Web 1T 5-gram, 10 European Languages Version 1* dataset produced by Google that is distributed by LDC as catalog no. LDC2009T25.

we construct an analysis string that corresponds to the desired result (i.e., nominative, singular, with weak inflection for deadjectival nouns), use SMOR in generation mode to get all possible strings (including those that SMOR overgenerates), and use a set of heuristics to predict the correct lemma out of the overgenerated set of strings. Besides similarity to the original word form (in terms of edit distance), we select for lemma candidates whose word forms are similar in frequency to the original word form, while preferring those with higher frequencies. While this approach is somewhat complicated by features of SMOR (underspecification of case for some, but not all analyses, inclusion of markers for old-standard and new-standard orthography into the analysis which need to be removed or added), we find that this approach yields high-quality lemmas for all analyzed word forms.

Finally, the lemmas of truncated items should include the understood completion. For example in the following example 3, the token *Bau-/TRUNC* should receive the lemma *Bauplanung%N* (construction planning) so that the inferred lemma and its part-of-speech are made explicit.

(3) *“Bei bedeutenden Bau- und Verkehrsplanungen müssen unsere*
“with important construction and traffic=plannings must our
*Einflußmöglichkeiten gestärkt werden”, fordern die*
influence=possibilities strengthened become”, demand the
*Behindertenverbände.*
disabled=associations.

“In the domain of important construction and traffic plans, our influence must become stronger”, demand the associations for the disabled.

The automatic completion of truncated items comprises two parts: on the one hand, finding a corresponding item that represents the context in which the lemma is interreted; on the other hand, determining the most likely completion of the truncated item given the context item (consisting of the truncated part plus a suffix of the context item).

For the first part, we simply consider the first content word following the separating comma or conjunction token as the completing context item. The second part, determining likely completions, is done by checking concatenations of the truncated item and suffixes of the potential context item for plausibility using frequency data (from the Google n-gram dataset). Among the possible completions constructed in this way, the most frequent one is considered most likely to be correct. While this frequency-based approach works very well in most cases, there are cases which result in incorrect solutions that can only be recognized considering both coordinated parts (i.e., the proposed completion and the context item).

## 3.3 Lemmatization: closed-class words

While there is considerable consensus about the lemmatization of open word classes, there is substantial variation in the lemmatization guidelines for closed-class words. This is largely due to the fact that it is not always clear what the division of labour should be between morphological tags and lemmatization. The lemmatization of definite article tokens is an example for the case in point: The TIGER treebank, for instance, uses only one lemma for each of definite and indefinite articles (mapping articles to either "*der*" for definite articles or "*ein*" for indefinite articles, corresponding to the male nominative singular form), but keeps the unmodified surface form in the case of personal pronouns.

For the TüBa-D/Z, the lemmatization guidelines prescribe that articles, possessives, and definite and indefinite pronouns are normalized to nominative singular, but keep gender and root. In cases of plurals that are unmarked for gender (e.g., *die Studierenden* the students/lit. the studying, which has only one form for both masculine and feminine), the possible strings for the determiner are all listed, separated by the diacritic '|'. This makes it possible to find these ambiguous items when searching for either the masculine or feminine article.

# 4 Empirical Results

Thus far, we have focused on the empirical issues to be solved, but have not discussed the division of labour between automatic pre-tagging – both morphology and lemmas are proposed by an automatic system, either as a set of several tags to choose from in the case of morphology, or in the form of a proposed lemma – and the subsequent manual annotation. As the amount of work needed for manual correction also depends on the error rate of the automatic component, it is useful to assess the quality of our lemmatizer. To do this, we compare the hand-corrected gold standard of the upcoming Release 6 of the treebank against the lemmas proposed by the semi-automatic system on one hand, and against lemmas proposed by TreeTagger (Schmid, 1995) based on the model that comes with it.

As our lemmatization guidelines include elements that go beyond morphological analysis by itself – consider the attachment of separable verb prefixes, and the completion of truncated items – we provide evaluation figures in two different settings:

- In the *strict* setting, a lemmatizer is required to provide the exact string that is to become part of the treebank annotation.
- In the *lenient* setting, a lemmatizer is not required to mark the difference between separable and nonseparable verb prefixes; separable verb prefixes that occur as separate tokens are not required to be attached; and whenever the guidelines require a split analysis because of morphological underspecification, a result that provides only one analysis (or a subset of the analyses that make up the correct tag) is counted as correct.

For our *TreeTagger* baseline, we used the output of TreeTagger with two modifications that are common practice for lemmatization: we replaced any unknown number (`@card@`) by its surface form, and any unknown other word (TreeTagger lemma `<unknown>`) was replaced by the corresponding surface word form.

| Category | TüBa-D/Z lemmatizer | TreeTagger |
|---|---|---|
| overall[1] | 99.4 | 77.7 |
| full verbs (VV...) | 99.1 | 74.8 |
| NN | 98.3 | 92.5 |
| NE | 99.4 | 96.5 |
| TRUNC | 63.6 | – |
| VVFIN | 99.4 | 77.4 |
| VVPP | 98.1 | 69.7 |
| VVIZU | 99.6 | – |
| VVIMP | 99.0 | 62.1 |

[1]: All STTS part-of-speech categories are included in this evaluation.

Table 1: Strict evaluation (≙ necessary edits)

| Category | TüBa-D/Z lemmatizer | TreeTagger |
|---|---|---|
| overall[2] | 99.4 | 94.2 |
| full verbs (VV...) | 99.1 | 91.4 |
| NN | 98.3 | 92.5 |
| NE | 99.4 | 96.5 |
| VVFIN | 99.4 | 86.0 |
| VVPP | 98.2 | 96.2 |
| VVIZU | 99.6 | 96.7 |
| VVIMP | 100.0 | 64.1 |

[2]: TRUNC, pronouns, and determiners are omitted in this evaluation.

Table 2: Lenient evaluation (≙ correctness of coarse-grained information)

The *lenient* setting (cf. table 2) is better suited as a comparison to other work on German lemmatization: In the *strict* setting (table 1), TreeTagger takes an accuracy penalty for not providing the additional information required by the treebank (reattaching separable verb prefixes and/or marking them, marking auxiliary versus full verb uses, or completing truncated items) and not producing pronoun and determiners according to the guidelines of the treebank. In the *lenient* setting, pronouns, determiners and truncated words are completely left out of the evaluation, as are separated verb prefixes; and the marking of separable verb prefixes is ignored. As can be seen in table 2 we still see an error reduction of 80%, which is mostly due to nouns, with a more modest error reduction of about 44% for verbs.

Among work that uses lemmatization in treebanking for German, no accuracy figures can be found in the published literature: the mechanisms used for the TIGER treebank (Brants et al., 2002) involve interactive selection by the user (either of complete LFG parses in the LFG-to-Tiger toolset, or of morphological entries using a program named TigerMorph on which no further details are provided), whereas the Smultron parallel treebank (Volk et al., 2009) uses the GerTwoL system (Haapalainen and Majorin, 1995) and post-corrects the predictions according to a set of guidelines that stay close to the output of the system.

In the more recent literature, Chrupala (2008) claims a lemmatization accuracy of 95% by cross-validation on the TIGER treebank using a memory-based learning approach to predict editing operations.

## 5 Generalizing to Unannotated Data

One important use case for treebanks (or, in general, corpora with rich linguistic annotation) is the investigation of complex phenomena that benefit from the additional annotation levels; however, in many cases the size limitations of a manually annotated treebank limit the potential usefulness. This is true for phenomena which are very rare in themselves, but also for the kind of linguistic phenomenon where multiple confounding factors make quantitative analysis a more challenging enterprise. As an example, primarily temporal discourse connectives such as *nachdem* (after/since) or *während* (while) occur relatively often in the TüBa-D/Z treebank, with more than 300 occurrences each, but a quantitative analysis that takes into account lexical and aspectual information can benefit immensely from the additional examples that would be found in larger unannotated corpora.

Leaving behind the realm of the carefully curated treebank would normally also entail rewriting most or all of the feature extraction, since neither the finer-grained lemmas nor the syntactic structure would be reproduced by a pipeline built from off-the-shelf components.

Using a parser that integrates SMOR for lexical prediction and yields the grammatical function labels necessary for the case prediction in morphology (Versley and Rehbein, 2009), however, allows us to use a syntax-first approach even for completely automatic annotation, as we have all the information that is needed for the morphological disambiguation. Remaining ambiguities (which would be left open for annotators to choose from in the case of treebank annotation) can be resolved using a simple CRF sequence tagger, as the most important global ambiguities are resolved using syntax information. Figure 2 shows an example where syntax information (phrase structure and edge labels) from the parser was automatically enriched with morphological tags and lemmas.

## 6 Conclusions

In this paper, we presented a lemmatization procedure devised for the TüBa-D/Z treebank, as well as a tagger that performs partial morphological disambiguation and lemmatization steps automatically, taking advantage of the existing syntactic annotation. The scope of the lemmatization guidelines incorporates some features that go beyond pure morphological analysis (such as reattaching separable verb prefixes, marking auxiliary use of verbs, and completion of truncated items), but which are squarely within the intended purpose of lemmatization as recovering the

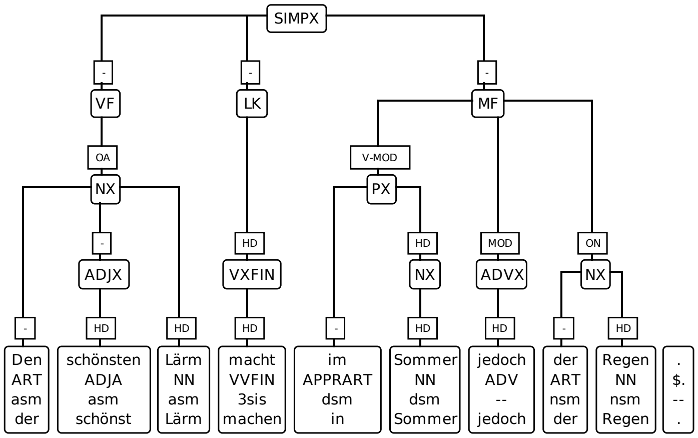


Figure 2: Example tree from the parser

terms which are mentioned in a text. The evaluation in section 4 shows that this more ambitious task definition is well within the reach of automatic tools.

Despite the high quality of fully automatic lemmatization, it is very useful to have gold-standard lemmas as part of the treebank, as this allows, in the context of more complex tasks such as coreference resolution, discourse tagging, or word sense disambiguation, to simplify the overall complexity of the task by removing the class of errors due to incorrect lemmatization. Moreover, the presence of both system-provided and gold-standard lemmatization allows to quantify the difference in the performance of these more complex applications that can be ascribed to the unsolved part of the lemmatization problem. The high quality of the automatic annotation as well as the multiple sessions of manual correction of all layers account for the gold standard quality of the TüBa-D/Z.

# References

Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). The TIGER treebank. In *Proc. TLT 2002*.

Chrupala, G. (2008). *Towards a Machine-Learning Architecture for Lexical Functional Grammar Parsing*. PhD thesis, Dublin City University.

Drach, E. (1937). *Grundgedanken der Deutschen Satzlehre*. Diesterweg, Frankfurt/M.

Gurevych, I. and Niederlich, H. (2005). Accessing GermaNet data and computing semantic relatedness. In *ACL 2005 short papers*.

Haapalainen, M. and Majorin, A. (1995). GERTWOL und Morphologische Disambiguierung fürs Deutsche. In *NODALIDA 1995*.

Hinrichs, E., Kübler, S., Naumann, K., Telljohann, H., and Trushkina, J. (2004). Recent developments of Linguistic Annotations of the TüBa-D/Z Treebank. In *Proceedings of TLT 2004*.

Höhle, T. N. (1986). Der Begriff 'Mittelfeld'. Anmerkungen über die Theorie der topologischen Felder. In Schöne, A., editor, *Kontroversen alte und neue. Akten des 7. Internationalen Germanistenkongresses Göttingen*, pages 329–340.

Kunze, C. and Lemnitzer, L. (2002). GermaNet – representation, visualization, application. In *Proceedings of LREC 2002*.

Miller, G. A. and Fellbaum, C. (1991). Semantic networks in English. *Cognition*, 41:197–229.

Schiller, A., Teufel, S., and Thielen, C. (1995). Guidelines für das Tagging deutscher Texte mit STTS. Technical report, Univ. Stuttgart / Univ. Tübingen.

Schmid, H. (1995). Improvements in part-of-speech tagging with an application to German. In *Proc. ACL-SIGDAT Workshop*.

Schmid, H., Fitschen, A., and Heid, U. (2004). SMOR: A German computational morphology covering derivation, composition and inflection. In *LREC 2004*.

Telljohann, H., Hinrichs, E. W., and Kübler, S. (2004). The TüBa-D/Z treebank: Annotating German with a context-free backbone. In *Proc. LREC 2004*, pages 2229–2232, Lisbon, Portugal.

Telljohann, H., Hinrichs, E. W., Kübler, S., Zinsmeister, H., and Beck, K. (2009). *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)*. University of Tübingen.

Versley, Y. (2007). Using the Web to resolve coreferent bridging in German newspaper text. In *Proceedings of GLDV-Frühjahrstagung 2007*, Tübingen. Narr.

Versley, Y. and Rehbein, I. (2009). Scalable discriminative parsing for German. In *Proc. IWPT 2009*.

Volk, M., Marek, T., and Samuelsson, Y. (2009). Smultron (version 2.0) - the Stockholm MULtilingual parallel TReebank. `http://www.cl.uzh.ch/research/paralleltreebanks_en.html`. An English-German-Swedish parallel Treebank with sub-sentential alignments.

# Mining Discourse Treebanks with XQuery

Xuchen Yao [*]
Johns Hopkins University
Computer Science
xuchen@cs.jhu.edu

Gosse Bouma
University of Groningen
Information Science
g.bouma@rug.nl

**Abstract**

We argue that a combination of XML and XQuery provides a generic and powerful method for representing and querying complex (multilevel) annotated corpora. XML is a widely used standard for coding and distributing annotated corpora, but the advantages of techniques for processing XML are not always realized in full. We show that XQuery, a completely generic query language for XML, has the expressive power required for advanced linguistic queries, its modular nature can help in providing corpus-specific functionality to users, and support of XQuery by XML database systems allows large corpora to be searched efficiently.

## 1 Introduction

Annotated corpora can contain information on many different aspects of linguistic structure, and as a consequence, tools to search annotated corpora differ quite substantially. Even for corpora with the same level of annotation, many different search tools exist. Lai and Bird [4], for instance, evaluate six different query languages (Tgrep2, TIGERSearch, Emu, CorpusSearch, NiteQL and LPath) on seven common search tasks for syntactic treebanks. All languages succeed in at least five of the seven tasks. The source code for each of the languages shows clearly, however, that these languages have differences in syntax and need special attention to work properly. As each corpus tends to support only a single query language, this means that users, especially those working with multiple corpora, must learn to work with a different query language each time they need to use a different corpus.

[*] We would like to thank the Erasmus Mundus European Masters Program in Language and Communication Technologies (EM-LCT) for support. This work was carried out while the first author was an EM-LCT student at the University of Groningen.

For corpora encoding different levels of annotation, the situation can be even more frustrating. The Penn Discourse TreeBank (Prasad et al. [9]), which we discuss in more detail below, provides annotation of discourse structure for data that is syntactically annotated in the Penn Treebank (Marcus et al. [7]). Although pointers from the discourse annotation to the syntactic annotation are given, it is not the case that both levels of annotation are available in a single format. Tools are provided which support searching the discourse annotation while imposing syntactic constraints, but the functionality of these tools is limited, and many naturally occurring questions require additional programming.

A final problem with many linguistic query languages is that they are exactly that: they allow the formulation of queries that return fragments of the corpus satisfying the constraints formulated in the query, but they allow little or no control over the output of the query. Many lexical acquisition tasks require pairs of items (such as pairs of verb (stems) and (the (stem of the) head of its) direct object) to be extracted. For the Penn Discourse Treebank, one might be interested in listing all discourse connectives, along with the syntactic category of their arguments. Such tasks require languages which not only support selection of fragments, but also support for selection of elements in (the context of) matching fragments, and some form of control over the resulting output.

The majority of modern corpora are made available in XML or can be converted to XML. Given the fact that very powerful languages for processing XML are available (most notably, Xpath for searching, and XSLT and XQuery for processing and querying), the question naturally arises to what extent such languages make corpus specific search tools superfluous, and to what extent these generic languages can overcome some of the shortcomings of linguistic query languages.

In this paper, we argue that XML and generic XML processing languages, XQuery in particular, allow a uniform method for representing complex linguistic data, and for searching and extracting data from complex corpora. We use the Penn Discourse Treebank (PDTB) as an example. We use an XMLized version of the PDTB and show that data discussed in recent research using the PDTB can be extracted from the corpus using XQuery. Finally, we discuss efficiency issues.

The original PDTB is encoded in plain text and shipped with PDTB API, a Java package, to accomplish common query tasks. Conversion of the PDTB to XML means that both the discourse information and the syntactic information of the corresponding constituents can be represented in a uniform way in a single XML file. Consequently, querying such files with XQuery becomes possible. This has several advantages:

- XQuery is capable of extracting both tree-based information (such as syntactic trees) and nontree-like information (such as discourse relations) or more generally, “non-tree navigation” (Lai and Bird [4]).

- XQuery is capable of extracting information from the discourse annotation and syntactic annotation simultaneously (given a linking between the two, which is already provided by the original encoding), which is beyond the ability of the original PDTB APIs.

- As a functional programming language, XQuery code can be modular, reusable and extensible, and thus answers the call for “reusable, scalable software” in Lai and Bird [4]. A Javadoc style documentation mechanism[1] exists. Corpus-specific modules allow developers to hide much of the complexity of the underlying XML and can help users to access relevant parts of the annotation easily.

- XQuery is the *de facto* standard for querying XML databases. By storing the corpus in an XML database, fast and easy-to-manage retrieval becomes available, while XQuery can still be used to perform complex queries.

In section 2 the format of PDTB-XML is introduced. Queries which need to access both the syntactic and discourse annotation and the way in which these can be implemented in XQuery are introduced in section 3. XML databases and their efficiency on linguistic queries are tested in section 4. The final section concludes and addresses future development. Most of the algorithms described in this paper are implemented as XQuery APIs.[2]

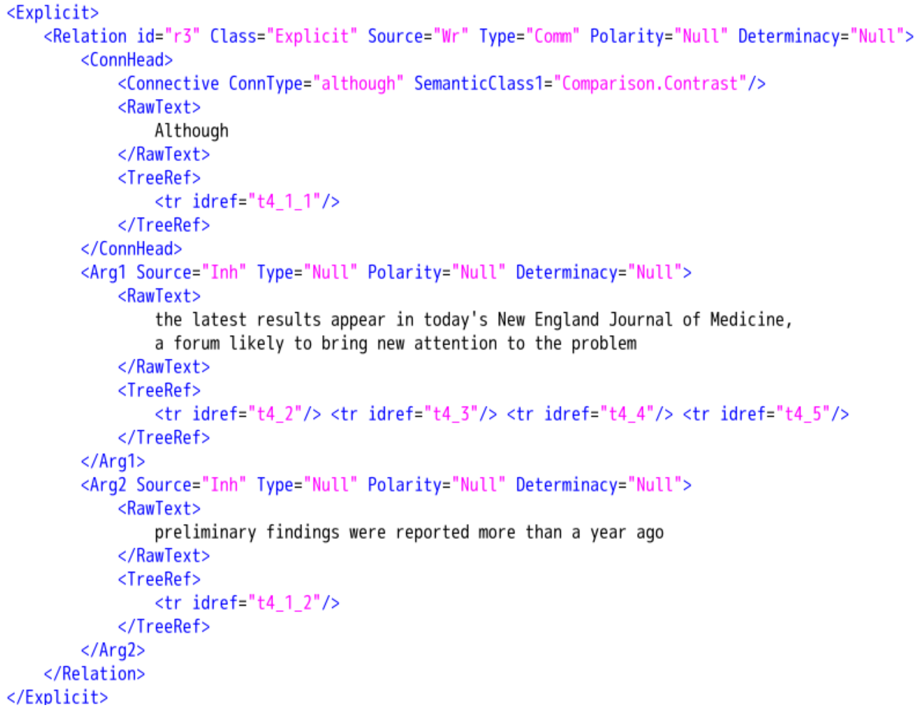
## 2 PDTB-XML

The XMLized Penn Discourse TreeBank (PDTB-XML, Yao et al. [12]) is an XML version of the PDTB (Prasad et al. [9]), created to support unrestricted access to both discourse and syntactic annotation. The original PDTB corpus uses a specific format[3] for its annotation. Each annotated article corresponds to three files, containing the original sentences, syntactic trees and discourse relations. A set of PDTB APIs is provided to support access to and visualisation of the annotation. Due to this architecture, the PDTB itself is not easily extensible or modifiable. By converting the annotation to XML files, the three separate annotation layers can be stored in a single XML file and annotations can be made more explicit and thus easier to understand and use by introducing elements and attributes. PDTB XML also inherits the merit of extensibility from XML.

[1] http://xqdoc.org/

[2] code.google.com/p/pdtb-xml/source/browse/trunk/xquery/pdtb.xq

[3] www.seas.upenn.edu/~pdtb/PDTBAPI/pdtb-annotation-manual.pdf



```
<Explicit>
    <Relation id="r3" Class="Explicit" Source="Wr" Type="Comm" Polarity="Null" Determinacy="Null">
        <ConnHead>
            <Connective ConnType="although" SemanticClass1="Comparison.Contrast"/>
            <RawText>
                Although
            </RawText>
            <TreeRef>
                <tr idref="t4_1_1"/>
            </TreeRef>
        </ConnHead>
        <Arg1 Source="Inh" Type="Null" Polarity="Null" Determinacy="Null">
            <RawText>
                the latest results appear in today's New England Journal of Medicine,
                a forum likely to bring new attention to the problem
            </RawText>
            <TreeRef>
                <tr idref="t4_2"/> <tr idref="t4_3"/> <tr idref="t4_4"/> <tr idref="t4_5"/>
            </TreeRef>
        </Arg1>
        <Arg2 Source="Inh" Type="Null" Polarity="Null" Determinacy="Null">
            <RawText>
                preliminary findings were reported more than a year ago
            </RawText>
            <TreeRef>
                <tr idref="t4_1_2"/>
            </TreeRef>
        </Arg2>
    </Relation>
</Explicit>
```

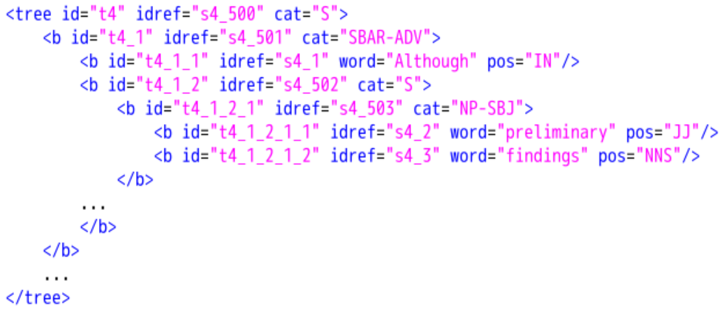
Figure 1: Example of an *Explicit* relation in XML.

There are five discourse relations in the PDTB (*Explicit, Implicit, AltLex, EntRel, NoRel*). Each relation has two arguments (*Arg1* and *Arg2*) and two optional supplements (*Sup1* and *Sup2*). Sentence (1) contains an *Explicit* discourse relation (the connective is underscored, *Arg1* is in italics and **Arg2** is in bold):

(1) Although **preliminary findings were reported more than a year ago**, *the latest results appear in today's New England Journal of Medicine, a forum likely to bring new attention to the problem.* (wsj_0003)

Figure 1 contains the annotation of (1) in XML. Instead of using fields, the role of each text fragment in the relation is made explicit by means of element names and attribute names. The link with the syntactic annotation is given in the `<TreeRef>` element associated with each `<ConnHead>, <Arg1>` and `<Arg2>`. A `<TreeRef>` contains one or more `<tr>` elements pointing to a node in a syntactic tree.

The syntactic annotation (i.e. the PTB) is encoded using the TIGER-XML format (Brants et al. [1]). Every sentence is syntactically represented by a *graph* containing *terminal*s and *nonterminal*s, where nonterminals have *edge*s connecting to terminals. As this graph does not give a direct tree structure an extra *tree*



```
<tree id="t4" idref="s4_500" cat="S">
    <b id="t4_1" idref="s4_501" cat="SBAR-ADV">
        <b id="t4_1_1" idref="s4_1" word="Although" pos="IN"/>
        <b id="t4_1_2" idref="s4_502" cat="S">
            <b id="t4_1_2_1" idref="s4_503" cat="NP-SBJ">
                <b id="t4_1_2_1_1" idref="s4_2" word="preliminary" pos="JJ"/>
                <b id="t4_1_2_1_2" idref="s4_3" word="findings" pos="NNS"/>
            </b>
        ...
        </b>
    </b>
    ...
</tree>
```

Figure 2: Fragment of a syntactic tree

element has been added to build the cross references between the syntactic part and the discourse relation part, as illustrated in Figure 2.

The PDTB-XML re-organizes the annotation format of the PDTB without loss of information, but with the advantage of a uniform, integrated, representation, and the possibility of future extensions. In the next section, we show how this uniform XML representation supports search and extraction tasks which need to refer to both discourse and syntax.

# 3 Search and Extraction with XQuery

## 3.1 XQuery and XPath

XQuery (Walmsley [11]) is a W3C recommendation[4] for querying XML databases. It uses the XPath standard[5] for locating elements in an XML document. A simple example of an XQuery script, which returns all relations with an *Explicit* connective of type *although*, is given below.

```
for $rel in //Relation[@Class="Explicit" and ConnHead/
   Connective[@ConnType="although"]]
return $rel
```

The `for` loop iteratively loops over all `<Relation>` elements somewhere inside the document. The @ symbol refers to attributes of an element and restricts the relations we are interested in to those of `Class` "`Explicit`". Furthermore, we require that the `<Relation>` must contain a `<ConnHead>` element which contains a `<Connective>` element whose `ConnType` attribute has the value "`although`".

[4] http://www.w3.org/TR/xquery/

[5] http://www.w3.org/TR/xpath/

### 3.2 Navigation in Trees

A variety of programs was developed to query syntactic trees ((Lai and Bird [4]; Lai and Bird [3]). Although the query languages are different, they are conceptually very similar. All languages contain operators or connectives for selecting mother, sibling, ancestor, and descendant nodes in a tree relative to a given node.

One common ground between syntactic trees and XML is tree-based structure. XPath, the XML navigation language incorporated in XQuery, has extensive support for selecting XML elements that are siblings, ancestors, or descendants relative to some given XML element. Table 1 shows how some of the operators of Tgrep2 (Rohde [10]) can be expressed using XPath and XQuery.

Even more functionality can be obtained by using the possibility in XQuery to add user defined functions. For instance, to select only elements that are a *leftmost descendant of* a given element, we can add the function `left-desc` to the module `pdtb`:

```
declare function pdtb:left-desc($desc, $top)
{ if ($top = $desc) then true()
  else if ($top/* ) then  pdtb:left-desc($desc, $top/*[1])
  else false()
} ;
```

Note that `left-desc` is a boolean function that checks whether `$desc` is a leftmost descendant of `$top`. The function is recursive in that it returns `true` if `$desc = $top` and else calls `left-desc` with the leftmost daughter of `$top` as second argument. If `$top` hs no daughters, the function returns `false`. A slight variant of this function returns the *set of leftmost daughters* of a given node.

In the PDTB-XML Query API, most of the tree patterns in (Rohde [10]) are implemented in less than 100 lines of code. This offers users functionality equivalent to that of other tree query languages with minimal development effort. The efficiency problem will be addressed in Section 4.

### 3.3 A Case Study: Range Relations

A number of discourse researches deal with positional relations between two arguments. Lee et al. [5] investigate the occurrences of shared discourse structures with a special focus on subordinate clauses. The relevance of their study is based on the necessity of describing not a single tree discourse hierarchy, but broader structures with complex dependencies. The authors identify four non-tree-like dependencies: shared argument, properly contained argument, pure crossing, and partially overlapping arguments. Lee et al. [6] investigate to what extent discourse arguments introduced by a subordinating conjunction (*while, because, after, since, ...*) can be

| Axis | XQuery example | Tgrep2 | Meaning |
|---|---|---|---|
| child:: | $b:=$a/child::* | A < B | A immediately dominates B |
| descendant:: | $b:=$a/descendant::* | A<<B | A dominates B |
| following:: | $b:=$a/following::* | A .. B | A precedes B |
| following-sibling:: | $b:=$a/following-sibling::* | A $.. B | A is a sister of B and precedes B |
| parent:: | $b:=$a/parent::* | A > B | A is immediately dominated by B |
| ancestor:: | $b:=$a/ancestor::* | A>>B | A is dominated by B |
| preceding:: | $b:=$a/preceding:: | A ,, B | A follows B |
| preceding-sibling:: | $b:=$a/preceding-sibling::* | A $,, B | A is a sister of B and follows B |

Table 1: Tree-based navigation with XPath and XQuery

used as argument of a following discourse relation. That is, in examples such as (2), the discourse particle *however* connects the clause *All of the ... period* and the subordinated clause *while other ... results*. Lee et al. [6] found 349 instances of this configuration in the PDTB, about 4% of the relevant cases (in 12% of the cases only the matrix clause was selected as argument of a following sentence, and in 84% of the cases the complete preceding clause was selected).

(2) GM also had dismal results in the first 10 days of the month, <u>while</u> **other auto makers reported mixed results**. All of the Big Three suffered in the just-ended period, <u>however</u>. (wsj_1139)

Note that gathering the relevant data to study this phenomenon requires access to both discourse annotation and syntax. Here, we will demonstrate that we can do this using only the PDTB-XML and XQuery.

The XQuery script we used is given in Figure 3. It searches a corpus file for relations `$rel` which contain a discourse connective that is one of the frequent subordinating conjunctions in the corpus, as suggested by Lee et al. [6]. To find such relations, we use a regular expression (i.e. the `match` function) that searches the text of connectives for *although, however, after, etc.* Next, the variable `$shared` is a following discourse relation, which must meet the requirement that its first argument (`Arg1`) must be shared with the second argument (`Arg2`) of the discourse relation introduced by the subordinating conjunction. A complication of the PDTB annotation is that `$shared/Arg1` is always the concatenation of `$rel/ConnHead` (i.e. the conjunction word) and `$rel/Arg2`. Thus,

```
for $c in collection($dir)/corpus
for $rel in $c/Relations/*/Relation[ConnHead/RawText[
     matches(.,"(although|however|after|as|....)","i")]]
let $shared := $c/Relations/*/Relation[
                 pdtb:gorn2tree(Arg1/TreeRef) =
                     pdtb:gorn2tree($rel/Arg2/TreeRef)/.. ]
where $shared
return
  <shared> <first>{$rel}</first>
          <second>{$shared}</second>
  </shared>
```

Figure 3: XQuery script to find subordinate clauses linked to a discourse relation introduced by a following sentence (following Lee et al. [6]).

we cannot simply check for identity of the text of the two arguments. Instead, we check whether the syntactic tree that corresponds to `$shared/Arg1` is the mother of the tree that corresponds to `$rel/Arg2`. Syntactic trees are found by the corpus-specific function `gorn2tree`[6] which uses the `id/idref` mechanism linking discourse relations to PTB annotation. The XPath expression `/..` locates the mother of an XML element. The `where` statement checks whether a discourse relation introducing a shared argument indeed exists, and the `return` statement returns the results.[7]

We can do even better, however. Note that Lee et al. [6] restrict their search to the *"12 most common subordinating conjunctions in the* PDTB." The practical reason for this restriction is that it allows finding the relevant cases by string matching over the text of connectives. There seems to be no principled reason, however, for this restriction. In the PDTB-XML we can also require that `$rel/ConnHead` must introduce a subordinate clause. Thus, instead of using a regular expression, we can select the relevant `$rel` relations as follows:

```
$Arg2Tree[@cat="S" and starts-with(../@cat, "SBAR")]]
```

Here, we use `$Arg2Tree` as shorthand for the `$rel/Arg2` tree. If `$Arg2Tree` is of category S and is dominated by an SBAR (or one of the subcategories of SBAR used in the PTB), we assume `Arg2` is a subordinate clause.[8] Now, we also find cases where the subordinating conjunction is a less frequent, such as *so that* in the example below.

[6] Tree nodes are numbered using the method of Gorn [2].

[7] The actual script uses a more detailed `return` statement, which normalizes the results and returns only relevant parts of the two discourse relations.

[8] We also defined a case where the tree corresponding to `$rel/Arg2` is of category S-NOM and dominated by a category PP-TMP (to cover the *after/before/since V-ing* cases.

(3) Computers have increasingly connected securities markets world-wide, so that **a buying or selling wave in one market is often passed around the globe**. So investors everywhere nervously eyed yesterday's opening in Tokyo, where the Nikkei average of 225 blue-chip stocks got off to a rocky start (wsj_2276)

The case-study in this section has concentrated on finding data that meets certain syntactic requirements (i.e. subordinate clauses), and that is used in a specific way in the discourse relations. Lee et al. [5] study cases where a discourse argument is shared, properly contained, crossing or overlapping with another discourse argument. Such studies can be carried out using the PDTB-XML and XQuery as well. All arguments in the PDTB-XML are linked to one or more syntactic constituents. All syntactic constituents have a yield which consists of words that have an `@id` attribute reflecting their position in the sentence. Given a set of pointers to syntactic constituents, we can easily obtain its *span* by collecting the `@id` in the yield of these constituents and selecting the smallest and largest member (using the sort function of XQuery and numerical comparisons). Given the *range* of a discourse argument, it is straightforward to define notions such as containment or overlap.

# 4 Performance Test

The PDTB-XML consists of files 2159 files (376MB in total). Running an XQuery script withn an XQuery processor such as Saxon[9] requires a scan of all the files and a tree-traversal of each file. For large amounts of data, this can lead to substantial memory consumption and long processing times. Indexing the PDTB-XML as a native XML database can improve performance. Here we chose to experiment with two open source XML database systems, eXist[10] (Meier [8]) and Oracle Berkeley DB XML.[11] A database system such as eXist brings two benefits to querying. Firstly, eXist has implemented an index-driven XQuery mechanism, where structural indices are used to identify relationships between nodes and thus in order to compute path expressions, nodes on the disk do not even need to be loaded into memory. This guarantees a high speed performance. Secondly, this index-based approach causes the axis (an axis defines a node-set relative to the current node, such as *parent*, *child*, etc.) to have only a minimal effect on performance. For instance, the XPath expression A//D is supposed to be even faster than A/B/C/D.

[9] `http://saxon.sourceforge.net/`

[10] `http://www.exist-db.org`

[11] `http://www.oracle.com/database/berkeley-db/xml/index.html`

The testing scenario is intended to investigate whether an XML database can facilitate a user's everyday usage. An open source XQuery processor, Saxon-HE 9.2, is compared against eXist (v1.4) and Berkeley DB XML (BDB in short, v2.5.13). The hardware setting is Intel Xeon X5355 @2.66GHz with 16G RAM. All of the systems execute exactly the same XQuery and the time is recorded. All output is diverted to a null device (/dev/null) to avoid a slow-down caused by flushing standard output.

We have selected seven query tasks covering all parts of the PDTB-XML (see Table 2). Tasks 1-3 are from Lai and Bird [4], and focus on the syntactic part of the PDTB-XML. Task 4 extracts only discourse relations. Task 5 first queries syntax and then queries corresponding discourse elements, while task 6 works the other way around. Task 7 is a complex example inspired by the case study in section 3. From the results in Table 2 the following observations can be drawn:

1. Indexing can greatly reduce query time, especially in simple tasks (such as task 1-4).

2. In complex tasks, computing references on-the-fly determines the total query time. Saxon outperforms eXist in task 6 and 7 and BDB in task 5 and 6.

3. BDB is not good at resolving coreferences, thus it was slow in task 5 and 6. The reason for this is unknown.

Our experiments lead to mixed results. For most simple queries, results can be retrieved within two minutes in a deployed database system. But as the complexity goes up, the execution time also increases. This increase is mostly due to the fact that relations between syntax and discourse must be computed during retrieval. Thus the advantages of indexing do not apply. Efficiency of querying complex and interlinked XML is still a problem that needs to be addressed in future work.

## 5 Conclusion

In this paper, we have investigated the merits of XQuery for processing corpora with complex linguistic annotation. We have concentrated on the PDTB, a corpus that combines annotation of discourse relations with links to the corresponding syntactic annotation (from the PTB). The PDTB-XML combines the two annotations in a single XML format, thus offering a uniform representation, with possibilities for future extensions. The general purpose XML query language XQuery offers the functionality to query the syntactic part of the PDTB-XML, and can be used to formulate queries that need to address both the syntactic and the discourse annotation. Finally, we have compared two native XML databases, eXist and Oracle Berkeley

|  | Query | Saxon | eXist | BDB |
|---|---|---|---|---|
| 1 | sentences that include the word *saw* | 5m51s | 19s | 15s |
| 2 | NPs whose rightmost child is a noun | 6m23s | 55s | 1m20s |
| 3 | VPs that contain a verb immediately followed by an NP immediately followed by a PP | 6m43s | 1m18s | 1m20s |
| 4 | all *Explicit* relations whose connective type is *because* | 2m9s | 0.5s | 0.1s |
| 5 | return connectives and corresponding POS tags of all *Explicit* relations | 7m17s | 2m27s | 30m30s |
| 6 | all words with POS="CC" that function as connectives | 7m3s | 15m21s | 21m33s |
| 7 | all arguments that are in a *range-contains* relation to another argument | 32m26s | did not finish in 1h | 7m13s |

Table 2: Execution time of common queries

DB XML against a stand-alone XQuery processor, Saxon. The evaluation shows that by indexing the elements and attributes in the XML database the query time can be greatly reduced for simple tasks. Computing dependencies between XML elements on the fly, as is typically required for queries that address both syntax and discourse annotation, means that the advantages of indexing are lost to a large extent, and processing times go up sharply. Efficiency of such queries needs to be addressed in future research.

## References

[1] Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. The tiger treebank. In *In Proceedings of the Workshop on Treebanks and Linguistic Theories*, pages 24–41, 2002.

[2] S. Gorn. Explicit definitions and linguistic dominoes. In J. Hart & S. Takasu, editor, *Systems and Computer Science*, pages 77–115. 1967.

[3] C. Lai and S. Bird. Querying linguistic trees. *Journal of Logic, Language and Information*, 19(1):53–73, 2010.

[4] Catherine Lai and Steven Bird. Querying and updating treebanks: A critical survey and requirements analysis. In *In Proceedings of the Australasian Language Technology Workshop*, pages 139–146, 2004.

[5] Alan Lee, Rashmi Prasad, Aravind Joshi, Nikhil Dinesh, and Bonnie Webber. Complexity of Dependencies in Discourse: Are Dependencies in Discourse More Complex than in Syntax? In *the 5th International Workshop on Treebanks and Linguistic Theories*, Prague, Czech Republic, December 2006.

[6] Alan Lee, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. Departures from Tree Structures in Discourse: Shared Arguments in the Penn Discourse Treebank. In *Proceedings of the Constraints in Discourse III Workshop*, 2008.

[7] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330, 1994.

[8] Wolfgang Meier. eXist: An Open Source Native XML Database. In Akmal Chaudhri, Mario Jeckle, Erhard Rahm, and Rainer Unland, editors, *Web, Web-Services, and Database Systems*, volume 2593 of *Lecture Notes in Computer Science*, pages 169–183. Springer Berlin / Heidelberg, 2003.

[9] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008.

[10] D.L.T. Rohde. Tgrep2 user manual. *URL: http://tedlab.mit.edu/˜dr/Tgrep2*, 2004.

[11] Priscilla Walmsley. *XQuery*. O'Reilly, 2007.

[12] Xuchen Yao, Irina Borisova, and Mehwish Alam. PDTB XML: the XMLization of the Penn Discourse TreeBank 2.0. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC 10)*, Malta, 2010.

# NEALT Proceedings Series

The NEALT Proceedings Series publishes peer-reviewed proceedings of scientific events that have a thematic and geographical relation to the purposes of NEALT.

### List of Published and Forthcoming Volumes

1. Proceedings of the **Sixth International Workshop on Treebanks and Linguistic Theories**. December 7–8, 2007, Bergen, Norway. Editors: Koenraad De Smedt, Jan Hajič and Sandra Kübler

2. Proceedings of the **Second Workshop on Anaphora Resolution (WAR II)**. August 29–31, 2008, Bergen, Norway. Editor: Christer Johansson

3. Proceedings of the **Workshop on NLP for Reading and Writing — Resources, Algorithms and Tools**. November 20, 2008, Stockholm, Sweden. Editors: Rickard Domeij, Sofie Johansson Kokkinakis, Ola Knutsson and Sylvana Sofkova Hashemi

4. Proceedings of the **17th Nordic Conference of Computational Linguistics NODALIDA 2009**. May 14-16, 2009, University of Southern Denmark in Odense, Denmark. Editors: Kristiina Jokinen and Eckhard Bick

5. Proceedings of the **NODALIDA 2009 workshop Nordic Perspectives on the CLARIN Infrastructure of Common Language Resources**. May 14, 2009, University of Southern Denmark in Odense, Denmark. Editors: Rickard Domeij, Kimmo Koskenniemi, Steven Krauwer, Bente Maegaard, Eiríkur Rögnvaldsson and Koenraad de Smedt.

6. Proceedings of the **NODALIDA 2009 workshop Multimodal Communication – from Human Behaviour to Computational Models**. May 14, 2009, University of Southern Denmark in Odense, Denmark. Editors: Costanza Navarretta, Patrizia Paggio, Jens Allwood, Elisabeth Alsén and Yasuhiro Katagiri.

7. Proceedings of the **NODALIDA 2009 workshop WordNets and other Lexical Semantic Resources — between Lexical Semantics, Lexicography, Terminology and Formal Ontologies**. May 14, 2009, University of Southern Denmark in Odense, Denmark. Editors: Bolette Sandford Pedersen, Anna Braasch, Sanni Nimb and Ruth Vatvedt Fjeld.

8. Proceedings of the **NODALIDA 2009 workshop Constraint Grammar and robust parsing**. May 14, 2009, University of Southern Denmark in Odense, Denmark. Editors: Eckhard Bick, Kristin Hagen, Kaili Müürisep and Trond Trosterud.

9. Proceedings of the **Ninth International Workshop on Treebanks and Linguistic Theories**. December 3-4, 2010, University of Tartu, Estonia. Editors: Markus Dickinson, Kaili Müürisep and Marco Passarotti