Proceedings of the

Third Workshop on Treebanks and Linguistic Theories (TLT 2004)

Tübingen, December 10–11, 2004

organized by the

Collaborative Research Centre 441
"Linguistic Data Structures"
University of Tübingen, Germany

and the

Nordic Treebank Network

Editors

Sandra Kübler
Joakim Nivre
Erhard Hinrichs
Holger Wunsch

Kübler/Nivre/Hinrichs/Wunsch
Proceedings of the Third Workshop on Treebanks and Linguistic Theories (TLT 2004)

ISBN: 3-9809183-2-7

For additional copies please contact:

Seminar für Sprachwissenschaft / Abteilung Computerlinguistik
Sekretariat
Wilhelmstr. 19
72074 Tübingen
Germany
E-Mail: `sekr@sfs.uni-tuebingen.de`

Workshop Programme Committee

Emily Bender, USA
Thorsten Brants, USA
Koenraad de Smedt, Norway
Eva Ejerhed, Sweden
Tomaz Erjavec, Slovenia
Annette Frank, Germany
Jan Hajic, Czech Republic
Erhard Hinrichs, Germany
Kimmo Koskenniemi, Finland
Tony Kroch, USA
Matthias Trautner Kromann, Denmark
Sandra Kübler, Germany (co-chair)
Yuji Matsumoto, Japan
Detmar Meurers, USA
Joakim Nivre, Sweden (co-chair)
Karel Oliva, Austria, Czech Republic
Petya Osenova, Bulgaria
Beatrice Santorini, USA
Kiril Simov, Bulgaria
Martin Volk, Sweden

Table of Contents

Papers

Posters

Author Index

Arborest – a VISL-Style Treebank Derived from an Estonian Constraint Grammar Corpus

Eckhard Bick*, Heli Uibo+ and Kaili Müürisep+

*Institute of Language and Communication
University of Southern Denmark
lineb@hum.au.dk

+Institute of Computer Science
University of Tartu, Estonia
{heli.uibo, kaili.muurisep}@ut.ee

1 Introduction

Treebank creation is a very labor-consuming task, especially if the applications intended include machine learning, gold standard parser evaluation or teaching, since only a manually checked syntactically annotated corpus can provide optimal support for these purposes. There are, however, possibilities to make the annotation process (partly) automatic, saving (manual) annotation time and/or allowing the creation of larger corpora. Whenever possible, existing resources – both corpora and grammars – should be reused.

In the case of the Estonian treebank project Arborest, we have therefore opted to make use of existing technology and experiences from the VISL project[1], where two-stage systems including both Constraint Grammar (CG)- and Phrase Structure Grammar (PSG)-parsers have been used to build treebanks for several languages (Bick, 2003 [1]). Moreover, the VISL annotation scheme has been adopted as a standard for tagging the parallel corpus in Nordic Treebank Network[2]. For Estonian, there already exists a shallow syntactically annotated – and proof-read – corpus, allowing us to bypass the first step in treebank construction (CG-parsing).

This paper describes how a VISL-style hybrid treebank of Estonian has been semi-automatically derived from this corpus with a special Phrase Structure Grammar, using as terminals not words, but CG function tags. We will analyze the

[1] URL: http://visl.sdu.dk

[2] URL: http://w3.msi.vxu.se/~nivre/research/nt.html

results of the experiment and look more thoroughly at adverbials, non-finite verb constructions and complex noun phrases.

The questions we will try to answer are:

- How much can we automatize the process of treebank creation on the basis of the existing morphologically and shallow syntactically tagged corpus?
- What kind of additional information could the PSG rules obtain from morphological analysis, if implemented in the compiler formalism?
- What kind of information is principally missing in the Estonian CG corpus and what kind of enrichment of categories is needed to facilitate the automatic treebank creation?

2 Estonian Constraint Grammar Corpus

The shallow syntactically annotated corpus was considered necessary for training and evaluation of the Constraint Grammar based shallow syntactic parser of Estonian, the detailed description of which is given in the subsection 2.1. The development of the corpus started in 1998 with the gold standard corpus, consisting of 20 000 words of Estonian original fiction from 1980s. During 1999-2003 the corpus has been extended to ca 200 000 words, including 177 000 words of fiction, 10 000 words of newspaper texts and 6 000 words of legal texts. The process of creation of Estonian CG Corpus is described in (Uibo, 2004, [11]). 65 000 words of newspapers from 1996-99 are being added in 2004.

2.1 Estonian Constraint Grammar Parser

The Estonian Constrain Grammar parser (Müürisep et al, 2003 [8]) has been developed in 1996-2000 by T. Puolakainen and K. Müürisep. It is the first attempt to automate the syntactic analysis of Estonian.

The main idea of the Constraint Grammar (Karlsson et al, 1995 [5]) is that it determines the surface-level syntactic analysis of the text which has gone through prior morphological analysis. The process of syntactic analysis consists of three stages: morphological disambiguation, identification of clause boundaries, and identification of syntactic functions of words. Grammatical features of words are presented in the forms of tags which are attached to words. The tags indicate the inflectional and derivational properties of the word and the word class membership, the tags attached during the last stage of the analysis indicate its syntactic functions. The underlying principle in determining both the morphological interpretation and the syntactic functions is the same: first all the possible labels are attached to words and then the ones that do not fit the context are removed by applying special rules or constraints. Constraint Grammar consists of hand written rules which by checking the context decide whether an interpretation is correct or has to be removed.

A number of rules are clearly of a heuristic nature – the rule might not be 100 % true but its proficiency rate is very high, compared to the number of errors. Several rules have been compiled solely on the statistical information about the word order in the sentence. The rules are grouped in such a way that the most reliable ones or those that cause least errors are in the main part of the grammar; the heuristic rules have been grouped based on their reliability.

The grammar consists of 1,240 morphological disambiguation rules, 47 clause boundary detection rules, 180 morphosyntactic mapping rules and 1,118 syntactic constraints. The morphological disambiguation rules are commented in (Puolakainen, 2001 [9]) and syntactic constraints in (Müürisep, 2000 [7]).

As the result of tests, 86.6 % of words become morphologically unambiguous, and the error rate of the morphological disambiguator is 1.8 %.The results of the full analysis show an ambiguity rate of 17 % (83 % of all wordforms are unambiguous) and error rate of 3.5 % (Müürisep et al, 2003 [8]).

2.2 Estonian Constraint Grammar Tagset

Estonian Constraint Grammar (EstCG) uses the following set of syntactic tags:
@+FMV – finite main verb, @-FMV – non-finite main verb
@+FCV – finite modal/auxiliary verb, @-FCV – non-finite modal/auxiliary verb
@NEG – negator (particles *ei, ära* as a part of a negative verb-form)
@SUBJ – subject, @OBJ – object, @PRD – predicative complement
@ADVL – clause level adverbial or modifier of an adverb or an adjective
@AN> or @<AN – an adjective or ordinal as a modifier
@NN> or @<NN – noun as a modifier (of a noun)
@AD> or @<AD – adverb as a modifier (of a noun)
@VN> or @<VN – participle as a modifier (of a noun)
@INF_N> or @<INF_N – infinitive as a modifier (of a noun)
@PN> or @<PN – an adpositional phrase as a whole as a modifier (of a noun)
@<P or @P> – noun belonging to the adpositional phrase (*on the <u>table</u>*)
@<Q or @Q> – noun belonging to the quantifier (*five <u>men</u>*)
@J – conjunction, @I – interjection

**CLB marks a very likely clause boundary and **CLB-C a less likely clause boundary. The analysis is performed inside the clause (sentential clause) boundaries only. No attempt is made to connect the clauses.

2.3 Representation Formats of EstCG Corpus

Part of EstCG Corpus is available as a directory of text files in the web[3]. In these files one word-form occupies two lines: the word-form itself is on the first line and the lemma+inflectional endings, morphological analysis and syntactical tag are on the second line (cf. Figure 1).

EstCG Corpus has also been converted to NEGRA export format (Brants, 1997 [2]) by Kaarel Kaljurand[4], thus now it can be searched and visualized with the TIGERSearch tool (Lezius, 2002 [6]). However, the trees are very flat – the smallest unit for grouping is a subclause and all the subclauses are at one and the same level. It is because CG markup includes clause boundary tags only; it does not contain information about the hierarchy of subclauses.

```
Mälestustes
   mälestus+tes //_S_ com pl in #cap //  **CLB @ADVL
muutus
   muutu+s //_V_ main indic impf ps3 sg ps af #FinV #Intr //  @+FMV
kõik
   kõik+0 //_P_ det sg nom //  @SUBJ
vapustavalt
   vapustavalt+0 //_D_ //  @ADVL
kauniks
   kaunis+ks //_A_ pos sg tr //  @ADVL
$.$.$.
   $.$.$. //_Z_ Ell //
```

Figure 1: Example sentence from EstCG Corpus.
(*Everything became strikingly beautiful in the memories...*)

3 VISL-style treebanks

The VISL annotation principles and set of labels (**Cafeteria Categories**) have been motivated by the need for a common set of grammatical categories within the multilingual project. Each VISL language and each VISL annotator have striven to make use of existing Cafeteria core categories wherever possible, adding subcategory extensions where necessary. Like the Nordic Treebank Network in general, the Arborest treebank project has chosen, wherever possible, to adhere to VISL style categories, adopting the following principles:

[3] URL: http://lepo.it.da.ut.ee/~heli_u/SA.html

[4] URL: http://psych.ut.ee/~kaarel/Programs/Treebank/EstCG2Negra

- Each node is annotated with both a function and a form label. Optimally, only branching nodes are used, i.e. the form of the daughter in a non-branching node is raised and expressed as the mother's function.
- **Function labels** have upper case key letters, **form labels** have lower case key letters. A complete node label in constituent grammar notation fuses form and function with a colon, e.g. S:np (subject noun phrase).
- **Subcategories** are attached to function labels in lower case, and to form labels with a hyphen.
- If crossing branches are unwanted, **discontinuous constituents** (crossing branch nodes) are marked with hyphens pointing towards the constituent's other part(s), e.g. P:vp- fA -P:vp.

The core categories for clause level function are the following:

- **S** Subject, subcategories e.g.: **Ss** Situative subject, **Sf** Formal subject
- **P** Predicator or Verbal constituent (function of "small vp")
- **O** Object, subcategories: **Od/Oacc** direct (accusative) object, **Oi/Odat** indirect (dative) object, **Op** prepositional object, **Ogen** genitive object
- **C** Predicative or complement, subcategories: **Cs** Subject complement, **Co** Object complement, **fC** free (subject) complement
- **A** Adverbial, subcategories e.g.: **fA** Free adverbial, **As** Subject-bound adverbial, **Ao** Object-bound adverbial

Form categories are divided into complex forms and word class forms. Complex forms are clauses **(cl)**, groups **(g)** and paratagmata or compound units **(par)**. Core categories are **fcl** Finite clause, **icl** Non-finite clause, **acl** Averbal (verb-elliptic) clause, **np** Noun phrase, **adjp** Adjective phrase, **advp** Adverb phrase, **pp** Prepositional phrase, **vp** Verb phrase, **par** Paratagma (Coordinated unit)

At the group level, the minimal annotation is dependency based, with one **H** (head) and one or more **D** (dependent) constituents.

The **vp** has special constituents, rather than head and dependent, since a syntactic/dependency view and a semantic "main verb" view can't agree on what the head is – **Vm** Main verb, **Vaux** Auxiliary, **Vpart** Verb integrated particle

Finally, word class form operates with a cafeteria consisting of **n**, **prop**, **v** (**v-fin**, **v-inf**, **v-pcp**), **adj**, **adv**, **pron** (with subclasses), **prp**, **art**, **num**, **conj** (**conj-s**, **conj-c**) and **intj**. The syntactic top-node receives the default function of **UTT** (utterance), but may be subdivided into **STA** statement, **QUE** question, **COM** command, **EXC** exclamation, **PER** performative. For undefined or unclear functions, (uppercase) **X** is used, undefined or unclear forms are **x**.

4 Conversion of EstCG Corpus to Arborest

4.1 The cg2tree compiler

The automatic creation of Arborest analyses is handled by a context free PSG, using VISL's open source cg2tree compiler. The formalism allows rewrite rules, which can address function and form tags, as well as word forms and base forms, all of which can be combined among themselves or with each other. Each rule can be conditioned by additional operators, like '!' (not as top node) or '+' (at least 2 daughters). Each daughter node expression can be suffixed by regex style existential operators (?, *, +). Since cg2tree grammars typically expect CG-annotated input, terminals will typically be function:form expressions, making use of word or base forms only as form restrictors.

```
    FM:fm = A:a.{'w1', 'w2', ...} B[->B2]:b[->b2] .... C*/+/? .... {D1, D2, ...}:^{d1,d2 ...}
```

In the rule above, FM and fm are the mother node's function and form, respectively, rewritten as a chain of daughters A ... D, where A is conditioned by a specific set of words, and D is given as a set of functions and a negated (^) set of forms. For B, tags are rewritten as B2 and b2, if the rule is instantiated, and C is an example of regular expression operators.

While the compiler formalism is language independent and has successfully been used to create CG-to-PSG grammars in a number of languages (dk, de, en, fr, cf. Bick, 2003 [1]), the grammar rules themselves have to be more language specific, and obviously also depend on the kind of CG input they receive – its tag granularity, level of dependency specification etc. Finally, the grammar will depend on the descriptive linguistic tradition it is set to implement (small or large VP, use of non-finite clauses etc). Luckily, since all Constraint Grammars so far share most of their core function tags and all adhere to the same structural paradigm (flat dependency grammar), at least rule **types** can be ported from one language to another, especially for lower level constituents. For Estonian, for instance, pp-rewriting is basically the same as for English, but left hand arguments have to be provided for, since the language uses adpositions rather than (only) prepositions.

4.2 The PSG grammar

The example rule creates object subclauses from underspecified input by drawing on complementizer words (the conjunctions "et+0" and "kas+0").

```
OBJ:fcl = $,? CLB ADVL:d? {SUB,ADVL}.{"et+0","kas+0"} {ADVL,OBJ,PRD}* P
{ADVL,OBJ,PRD}* SUBJ {ADVL,OBJ,PRD}* ARGS? CLB? ; # OVS, VSO, VOS (only
OSV lacking!)
```

Individual tags can be rewritten one-to-one inside a rule, if and when it is instantiated. Rules allow both function and form variables (X and x, respectively), which are, however, in the current formalism not unified across the right hand side of a rewriting rule.

The current PSG grammar comprizes 110 rules, roughly a quarter of which are finite clause rules, another quarter are phrase (group) rules, and a third quarter covers coordination patterns. With variable unification, the number of coordination rules could be reduced by using general rules like X:cu = X+ CO X.

In other VISL grammars, notably Germanic ones, the uniqueness principle has been implemented by specifying allowed constituent orders. For Estonian, however, which has a much freer word order, clause level constituent chains have to accommodate for all S-V-O combinations but the infamous OSV. Therefore, possible constituent chains have been lumped by using {ADVL, OBJ, PRD} or similar sets with the *-operator. As a result, current rules have a laxer uniqueness constraint, at clause level basically limited to subordinators, predicator and subject.

Though linguistic theory treats auxiliaries and verb chains in various ways, for the sake of notational compatibility, the VISL treebank convention of “small vp” was adopted, with a predicator constituent (P) consisting of finite and non-finite main verbs (MV), chain verb “auxiliaries” (CV) and negation particles, leaving objects and other verb complements outside the vp.

Not least in newspaper text, embedded sentences occur fairly frequently, often marked by parenthesis or pairs of quotes or hyphens. In order to reduce the complexity of the grammar, such punctuation is not ignored but rather used to delimit embedded sentences.

5 Results of Conversion

We have examined and manually revised 149 trees – the corpus *Estonian-best*, containing articles from an issue of the Estonian weekly newspaper "Eesti Ekspress" (August, 1996). 61 trees were correct, i.e. had both correct branching structure and correct labels for forms as well as for functions. Among the correct sentences the following subclause structures were represented (unified):

(1) (A) S (A) P A*
(2) S P (A) C (A)
(3) S P (A) O A*
(4) O S A P
(5) O P S A
(6) A O A P A+ (no subject)

(7) A+ P (A)S A*
(8) (A) P A*(S)A*O A*
(9) A P O S
(10) A* P C A O S
(11) C P A S

Generalizing, we could add A* everywhere in between S, P, O and C in the structures.

Estonian is a free-word-order language and that has been taken into account in the rules. Simple sentences with the word order S-P-O, S-O-P and P-S-O plus maybe A* everywhere have been correctly parsed. The predicative complement (C) can occur either after or before predicate. The structure (4), where the predicate is in the end, occurred in subordinated clauses only. However, a predicate may also occur at anterior positions in subordinated clauses. The subject is not an obligatory clause constituent in Estonian, and the subject is "inflexion-included" in the verb form (1rd or 2rd person verb forms).

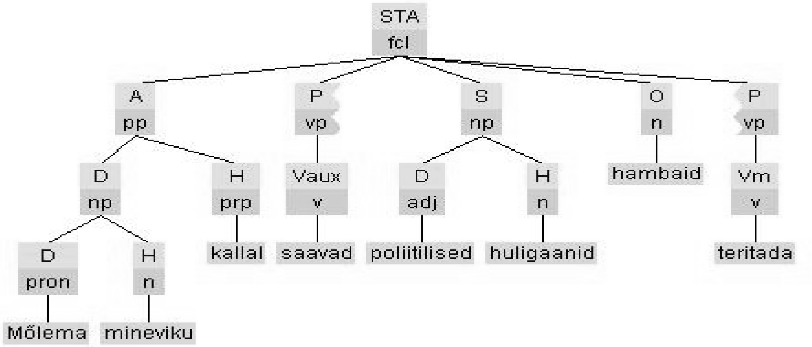


Figure 2: Example of a discontinuous verb phrase (*saavad teritada*). (*Political hooligans can sharpen their teeth on the past of both (persons).*)

In Estonian discontinuous verb phrases where object or adverbial(s) occur in the middle of the verb phrase are quite common. There is a convenient way to represent discontinuous structures in the VISL tag set and a comprehensible format to represent it graphically (cf. figure 2).

The trees for composite sentences (subclauses bound with *ja, ning (and)*, *või*, *ehk* (*or*) or comma) and complex sentences with subordinated clauses in the function of adverbial (*kui* ... *siis* (*if* ... *then*)) or object (beginning with the subordinating conjunction *et* or an interrogative-relative pronoun *kes*, *mis*) have also been correctly built.

In the sections 5.1 – 5.3 the entities that caused the largest numbers of false structures will be analyzed.

5.1 Adverbials

The family of adverbial constituents is represented by only two tags in EstCG – @AD> / @<AD – as adverbial modifiers of nouns (mostly state adverbials) and @ADVL – for all other adverbials (including adjective-phrase-internal adverbial modifiers, like "very big"). Therefore, it is sometimes unclear, where to attach adverbs. In the corpus *Estonian best* an adverb modified an adjective only in two sentences out of 149, but it was erroneously attached to the NP in more than 10 sentences (e.g. sentence 52 which is visualized in figure 3). Thus, the adverbial attachment rules are overgenerating and should be revised. Some PSG errors occurred, because a correcting rule turning ADVL into group dependents like DN or DA, overgenerated. Provided a 99% consistent adverbial tagging in the CG source corpus, such rules should, of course, be abolished, and the risk of overgeneration be reduced as a consequence.

The list of adverbs that can be only phrase-attached – *kõige, liiga, üpris, üsna* – can be exploited by PSG rules, but there is a considerably longer and open list of adverbs that can act both as free adverbials and adverbal modifiers.

Another solution to the adverbial problem is to subcategorize the ADVL tag. There are at least two different principles of classification of adverbials – by semantics and by syntactic function. For example, in Functional Dependency Grammar (Järvinen & Tapanainen 1998, [4]) tagset there are twenty different adverbial tags, classified by the semantic role of the adverb. Alternatively, we could divide the adverbials according to their syntactic functions, e.g. as follows:

1. AdjP or AdvP-dependent adverbials (*very big*, *too quickly*) [VISL: DA]
2. predicate-dependent adverbials (*He painted the wall green*) [VISL: Co, As, Ao. In Estonian syntax (Erelt et al, 1993 [3]) this is called "dependency adverbial" or "valency adverbial", as in Estonian syntax the object can be only in nominative, genitive or partitive case.]
3. non-predicate verb dependent adverbials (*Walking in the park was his favorite hobby.*) [VISL: fA within a non-finite rather than a finite clause]
4. free adverbials (*It is raining outside*.) [VISL: fA]

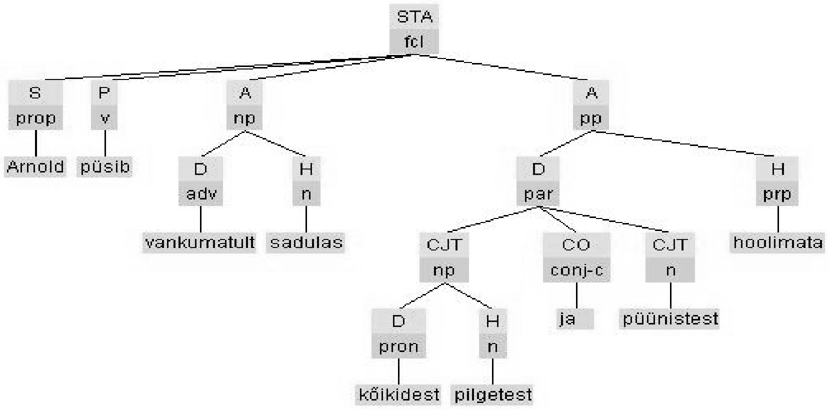


Figure 3: Tree where an adverb is falsely attached to a NP.
The adverb *vankumatult* (*immovably*) is actually a free adverbial.
(*Arnold sits immovably on his horse, regardless of all gibes and traps.*)

As one of the motivations for building Estonian treebank is the research on predicate-argument structures it is significant to distinguish at least between verb-dependent and independent adverbials.

5.2 Non-finite clauses

Non-finite clausal constructions (infinitival and averbal clauses, short clauses with participles as a predicate, ma-supine infinitival clauses, participles as noun modifiers) are not easy to recognize in Estonian, especially when they are not separated by a comma. This problem caused 8 errors in the *Estonian-best* corpus (example in figure 4).

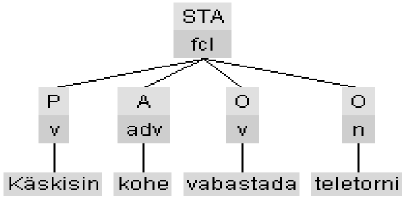


Figure 4: Sentence with unidentified non-finite subclause. (*(I) gave an order to vacate the television tower immediately*.) Here, *kohe vabastada teletorni* is an infinitival subordinate clause, which should be separately grouped in the sentence tree.

The solution can be to add an explicit CG-tag for the start word of such clauses. However, the automatic detection of non-finite clause boundaries is far from trivial. But for propositional semantics it would be very useful to have all the dependent objects and adverbials determined not only for finite but also for non-finite verbs (which often take arguments similarly to finite verbs).

5.3 Noun phrases

It is quite difficult to guess the structure of a complex NP relying on the CG tags @NN> and @<NN, because we only know the direction, in which the head is situated but we don't know, which word exactly is the head (sometimes a word, tagged as @NN> can be a head for another word tagged @NN>, etc.

Sometimes the head can be determined relying on the morphological information. If an NP consists of a proper or common noun in genitive case + adjective + substantive, with the latter two agreeing in case, e.g. "Ida-Virumaa raskest olukorrast" the structure is A:np(D:prop H:np (D:adj H:n)) but not A:np(D:adjp (D:prop H:adj) H:n). However, the present version of the open source VISL psg-compiler does not allow explicit reference to morphological features (even where they are known from CG input), unless cumbersome new 'word classes' are 'invented' for only this purpose (e.g. n-acc, n-gen, etc.). The necessary changes in the compiler formalism have been discussed in the VISL user community, but not yet implemented.

The CG-to-PSG rules demonstrated quite good results in NP extraction. We have compared the list of NP-s that were determined by the rules against the correct list of noun phrases from a part of the corpus *Estonian best*. The number of NP-s in the correct NP list was 253. The rules had the recall 93,3 % and the precision 92,5 % on noun phrase extraction. The errors were caused by false adverbial attachment described in section 5.2. The errors in the NP-internal structure have not been counted as this is not the matter of the NP extractor. Thus, as a side product, we have got quite a good noun phrase recognizer.

6 Comparison of (the expressive power of) CG and PSG

We can bring forth the following principal differences between CG and PSG (specifically, Arborest) which make it difficult to automatically convert the CG annotated corpus to PSG annotated corpus:

- CG: syntactic function and morphological form of each word determined
  Arborest: In addition, complex forms (phrases, subclauses, co-ordinated units) are established and their syntactic function annotated

- Attachment uncertainty. CG: no explicit dependencies, directional dependency markers only for group-level modifiers, not clause level dependents (e.g. @AN> and @<NN looking for NP-heads, but not @<ADVL looking for main verbs). Arborest, on the other hand, has to resolve all attachments, in connection with its constituent bracketing.
- CG: finite clause boundaries are determined but not non-finite clause boundaries. PSG-rules can therefore address the former, but not the latter, and has here to rely on functional relations, uniqueness principle etc.
- Attachment of subclauses. CG: The hierarchy of subclauses is not expressed, and subclause function is not annotated. As implemented in the VISL family of CGs, such information could be added to head verbs or complementizer words. So far, however, we have used a partial solution, exploiting a list of subordinating conjunctions and pronouns typical of, for instance, adverbial, relative or averbal constructions.

7 Conclusions and Future Developments

The experiment to derive a hybrid form+function treebank from Estonian Constraint Grammar corpus has been quite successful. The semi-automatic procedure is usable for treebank creation, although in the present stage it is still time-consuming. The revision of the corpus *Estonian best* (149 trees) took one week of full-time linguist's work (including the learning of the category set and textual representation format of the trees). The manual correction job could be made significantly easier with a graphical interactive tree editing tool (like *Annotate* or a planned interactive version of VISL's tree visualiser).

We believe that a particular strength of our method is that it, to a certain degree, processes function and structure separately, exploiting the robustness of syntactic-function tagging at the CG-level (and in this case, pre-existing manual revision), while adding structural information through a separate (PSG) grammar, allowing a more focussed linguistic revision. It may be of interest to point out, that our approach differs from other hybrid methods not only by employing a Constraint Grammar base, but also with respect to the order of steps, inverting the maybe more traditional progression from chunking to parsing to function labelling (edge labels).

The CG-to-PSG conversion rules have been most accurate on NP detection and simple sentence analysis consisting of the usual sentence constituents subject, object, predicate, predicative complement and adverbials in any order. The composite sentences and subordinate clauses have also been well analyzed, using the condition that a subordinate clause begins with one of the subordinating conjunctions or interrogative-relative pronouns given in the lexicon.

There are three possibilities to improve the CG-to-PSG treebank conversion results, best, if combined:

- revise cg2psg rules taking into account the results of the current evaluation
- refine CG markup (subcategorize adverbials, add non-finite and averbal clause boundaries)
- use more morphological (especially case) information in the PSG rules

During 2004–2008, it is planned to create a larger treebank using existing text corpora. We plan to turn the EstCG Corpus (200.000 words) into a treebank using the CG-to-PSG grammar. A kernel of 1000 sentences will be hand-corrected at the gold-standard level and used for documentation and exemplification. Part of the remaining treebank will also be revised, but in a somewhat looser fashion (e.g., no cross-revision), relying on the fact that at least with regard to syntactic function, the corpus has already been revised at the CG-level.

The main research plans connected to the Estonian treebank include the examination of the predicate-argument structures in the corpus and the revision of Rätsep's sentence templates (Rätsep, 1978 [10]) in the light of corpus data. In perspective, the nodes will also be provided by semantic information. We are also planning to work on phrase level alignment of Estonian-German-Swedish parallel treebank to take first steps towards machine translation.

References

[1] E. Bick. A CG & PSG Hybrid Approach to Automatic Corpus Annotation. In Kiril Simow & Petya Osenova: *Proceedings of SProLaC2003* (at Corpus Linguistics 2003, Lancaster), pp. 1-12

[2] T. Brants. *The Negra Export Format for Annotated Corpora*, Version 3. Techinal report. Dept of Computational Linguistics, University of Saarland.

[3] M. Erelt, R. Kasik, H. Metslang, H. Rajandi, K. Ross, H. Saari, K. Tael, S. Vare *Eesti keele grammatika II. Süntaks*. (The Grammar of Estonian II: Syntax) Institute of Estonian Language. Tallinn 1993.

[4] T. Järvinen and P. Tapanainen.Towards an implementable dependency grammar. In *Proceedings of the Workshop "Processing of Dependency-Based Grammars"*, (eds.) Sylvain Kahane and Alain Polgučre, Université de Montréal, Quebec, Canada, 15th August 1998, pp. 1-10.

[5] F. Karlsson, A. Anttila, J. Heikkilä, A. Voutilainen. *Constraint Grammar: a Language-Independent System for Parsing Unrestricted Text.* Mouton de Gruyter, 1995.

[6] W. Lezius. TIGERSearch – Ein Suchwerkzeug für Baumbanken. In: S. Busemann (editor): *Proceedings der 6. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2002)*. Saarbrücken, 2002.

[7] K. Müürisep. *Computer Grammar of Estonian: Syntax*. Dissertationes Mathematicae Universitatis Tartuensis – 22. Tartu, 2000.

[8] K. Müürisep, T. Puolakainen, K. Muischnek, M. Koit, T. Roosmaa, H. Uibo A New Language for Constraint Grammar: Estonian. *RANLP 2003 Proceedings*. Borovets, Bulgaria, 10-12 September 2003, pp. 304-310.

[9] T. Puolakainen *Computer Grammar of Estonian: Morphological Disambiguation*. Dissertationes Mathematicae Universitatis Tartuensis–27.Tartu, 2001.

[10] H. Rätsep. *Eesti keele lihtlausete tüübid.* (The templates of Estonian simple sentences) Tallinn, 1978.

[11] H. Uibo. Syntactically annotated corpora of Estonian. In *The First Baltic Conference "Human Language Technology – the Baltic Perspective*", Riga, Latvia, April 21-22, 2004, pp. 45-48.

Treebank Evidence for the Analysis of PP-Fronting

Gosse Bouma
Computational Linguistics
Rijksuniversiteit Groningen

1 Introduction

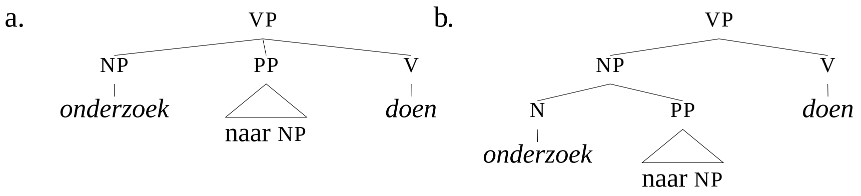
A long-standing discussion in Dutch linguistics is concerned with the status of the PP in sentences like (1). In (1-a), a full PP appears in sentence initial position, and in (1-b), the initial pronoun *daar* is interpreted as the object of the preposition *naar*.

(1) a. **Naar** deze gebeurtenis wordt nu nader **onderzoek** gedaan.
Into this event is now further research done
Further research on this event is now done

b. **Daar** is echter nauwelijks **onderzoek naar** verricht
There is however hardly research into carried out
However, hardly any research on this has been carried out

The PP can be seen as a dependent of the noun *onderzoek* or of the main verb, thus the basic structure for sentence (1-a) could be either as in (2-a) (V+PP) or as in (2-b) (N+PP).

(2)



The proper analysis of examples like those in (1) has been the topic of a heated debate (in Klein and van den Toorn (1977, 1979), and Kooij and Wiers (1979), among others), initiated by the observation in Bach and Horn (1976) that, according to the then current version of Transformational Grammar, extraction from NPs should not be possible in languages like Dutch. Thus, they are forced to adopt analysis (2-a) for both (1-a) and (1-b).

A similar issue arises in German. De Kuthy (2000) notes that sentences such as (3) (which she refers to as 'NP–PP *split*') have often been analyzed as involving extraction out of NP, but also as involving extraction out of a VP (perhaps after reanalysis).

(3) Über Syntax hat Hans sich ein Buch ausgeliehen
about syntax has Hans self a book borrowed
Hans borrowed a book on syntax

The N+PP analysis is intuitively plausible, as there seems to be a strong semantic relation between the noun and preposition. Furthermore, N+PP may precede the finite verb in main clauses, and thus clearly forms a constituent in some cases. Also, when N is preceded by certain definite determiners, fronting of the PP is almost impossible. This suggests PP-fronting is subject to a constraint on extraction from NP, something which seems highly problematic for a V+PP analysis. The V+PP analysis, on the other hand, is supported by the fact that PP-fronting seems to occur only with certain verbs. Furthermore, some nouns clearly select a PP, but do not allow fronting of this PP.

When constructing a treebank for Dutch, one frequently encounters examples such as (1) and a decision has to be made as to how to annotate these. The syntactic annotation of the Corpus of Spoken Dutch (CGN) (Moortgat, Schuurman and van der Wouden 2000) adopts an N+PP analysis for the following type of example:

(4) **daar** heb ik helemaal geen **zin in**
there have I totally no desire for
I have no desire for that at all

The Alpino-treebank of written Dutch,[1] on the other hand, has opted for the V+PP analysis. As one of the design goals of the Alpino-treebank was to produce output compatible with CGN, it seems that the annotation guidelines for either Alpino or CGN need to be reconsidered.

In this paper, we investigate to what extent corpus data can be used to decide on this matter. A corpus-based approach seems appropriate for at least two reasons. First, the claim that certain determiners block PP-fronting as well as the claim that PP-fronting occurs only with certain verbs, can be verified using corpus data. Second, there has been considerable disagreement between authors on the status of examples that were crucial in arguing for one or the other position. Examples considered ungrammatical in one paper were considered to be acceptable by authors arguing for a different analysis.[2] Coppen (1991) notes that the examples in his paper show varying acceptibility, and that linguistic intuitions with respect to these

[1] see van der Beek *et al.* (2002) and `www.let.rug.nl/~vannoord/trees`

[2] See Klein and van den Toorn (1977, p. 432), Klein and van den Toorn (1979, p. 105) and Kooij and Wiers (1979, p. 488).

data even seem to change over time.

In section 2, we describe the construction of a syntactically annotated corpus. Next, we investigate the role of the verb and the determiner in PP-fronting. We conclude that the verb plays an essential role in PP-fronting and that the preference for indefinite NPs PP-fronting may be related to this. We also observe that PPs may be included in relatives modifying the noun. This seems highly problematic for an N+PP analysis. A number of patterns which have been used as arguments for a particular analysis, are practically absent in the corpus. We conclude that the corpus data suggest that the V+PP analysis is more likely than the N+PP analysis, and that these expressions are best analyzed as phrasal verbs involving a prepositional complement.

2 Treebank Construction

We used the newspaper sections of the Twente News Corpus[3] (TWNC) as our initial corpus. The corpus contains text from major Dutch newspapers in the period 1994-2001, and has a size of approximately 300 million words. We believe that, at least for the phenomenon we are interested in, this corpus is representative for Dutch in general.

The discussion referred to in the introduction has focused on N+PP combinations displaying a strong semantic relation between the noun and the PP. Our first goal was to identify a number of such nouns in the corpus. To find N+P pairs with strong collocational properties, we ranked all N+P bigrams from the corpus using the *log-likelihood-test* of Dunning (1993). From the resulting list, we selected 16 bigrams as suitable candidates for our research (see table 1). Highly ranked bigrams which we discarded were parts of names (*ministerie van (ministry of)*), parts of complex prepositions (*(in) tegenstelling tot ((as) opposed to)*), and bigrams which did not occur in PP-fronting sentences.

Next, we automatically constructed two treebanks. The **general** corpus consists of sentences containing the relevant N+P combination. From the TWNC, we initially extracted 10.000 sentences per N+P collocation containing both N and P. 155.000 sentences were selected in total (as some bigrams did not occur 10.000 times). The *dependency tree* for each sentence was computed using the Alpino-system.[4] From the resulting treebank, we selected[5] those cases where the NP headed by N and the PP headed by P are both dependents of the same verb, or the PP is a dependent of N, and the NP headed by N is a dependent of a verb (i.e. and

[3] wwwhome.cs.utwente.nl/~druid/TwNC/TwNC-main.html

[4] www.let.rug.nl/~vannoord/alp

[5] using the XML-tool for searching dependency trees described in Bouma and Kllosterman (2002).

behoefte aan	*need for*	kritiek op	*critique on*
belangstelling voor	*interest in*	onderzoek naar	*investigation into*
bezwaar tegen	*objections against*	protest tegen	*protest against*
contact met	*contact with*	relatie tussen	*relation between*
discussie over	*discussion about*	twijfel aan	*doubt about*
gebrek aan	*lack of*	verhaal over	*story about*
gesprek over	*conversation about*	verschil tussen	*difference between*
informatie over	*information about*	vraag naar	*demand for*

Table 1: Selected N+P collocations

not part of a PP or other non-verbal constituent). Almost 51.000 sentences (containing 2.000 to 4.000 examples per N+P collocation) satisfy the syntactic selection criteria.

The second, **split**, corpus, consisted of PP-initial sentences and sentences containing a discontinuous PP consisting of an R-pronoun and a preposition (i.e. as in (1-b)). All sentences from the TWNC containing both N and P, but where P was also the first word in the string, were collected. Initially, this set consisted of almost 6.000 sentences. After syntactic analysis and selection, this was reduced to approximately 2.400 cases. The corpus was extended with all cases from the *general* corpus, containing a discontinuous PP. This gave rise to another 1.100 cases. Thus, the *split* corpus contains about 3.500 sentences (containing between 35 and 615 examples per N+P bigram).

Using an automatically constructed treebank, rather than raw or POS-tagged text, is essential for our purposes for two reasons. First, not all sentences containing N and P are actually valid instances of the pattern we are interested in (P might be heading a PP containing an NP headed by N, or the PP might be part of another NP, for instance). Second, we want to investigate which verbs co-occur with these N+P collocations. Therefore, we must be able to determine which verb actually selects for the NP headed by N. As we are interested in investigating the status of the PP, we need to consider both the case where the PP is analyzed as a dependent of N and the case where the the PP is analyzed as a dependent of V.

The main reason why we opted for using an automatically constructed treebank, instead of a manually annotated treebank, is size. Eventhough we used the full 300 million word TWNC as our source, the *split* corpus is relatively small, often containing less than 100 examples per N+P bigram. The largest manually annotated treebank for Dutch, CGN, contains only 1 million words, and contains only a few N+P bigram occurring more than 3 times in a *split* configuration. Still, one might wonder whether automatic analysis is sufficiently reliable to create a representative corpus. Allthough automatic analysis is not completely error-free,[6] the effect it has on the task at hand seems small. Automatic analysis does reliably filter cases where the NP is not a dependent of a verb, or where the NP and PP are dependents of a different verb. Also, the main verb selecting NP or both NP and PP is identified reliably. Finally, note that manually annotated corpora are not error free either. For the Alpino-corpus, for instance, at the end of the project an inter annotator agreement of 94.6% was achieved. Thus, the difference in reliability between manually and automatically annotated data is a gradient, rather than absolute. Nevertheless, errors do sometimes occur, and thus we did manually inspect many of the results found in the experiments below, especially cases involving small numbers.

3 The role of the verb

The idea that fronting of a PP or a discontinuous PP is possible only with certain verbs, has been used as argument for the V+PP analysis. In this section, we investigate whether corpus-data confirm this intuition.

Using the information provided by automatic syntactic analysis, as described in the previous section, we we counted how often a specific verb occurs with a specific N+P collocation in the *general* and in the *split* corpus. In particular, we counted the verbs with a dependent NP headed by N and containing a PP (headed by P) functioning as a dependent of the verb or the noun. To avoid inclusion of (verbs functioning as) auxiliaries and modals, verbs with a VP-dependent were excluded. If the possibility of a *split* configuration is determined by the verb, only a limited number of verbs should be found in the *split* corpus, and in *split* these verbs should occur more frequently than in the general corpus. In table 2, we present an overview of verbs found more than once in *split* and *general*, for 4 representative N+P combinations.

Table 2 shows that the combination *behoefte aan* mainly occurs with *hebben* en *zijn*. There are significant differences in the distribution of these verbs between *split* and *general*, however. For allmost all N+P collocations we investigated, statistically significant differences in distribution can be observed for the most frequent verbs. In some cases, significant differences for low frequent verbs can be observed as well.

The verbs *hebben*, *zijn* and *bestaan* are special in that they seem to allow *split* with almost all investigated N+P combinations. The role of *bestaan* is remarkable: this otherwise rather infrequent verb occurs frequently with 10 of the 17 inves-

[6] Van Noord and Malouf (2004) report that the Alpino-system identifies dependency relations with an accuracy of 87.8% on a representative 500 sentence subset of the TWNC.

		Split	**Gen**
behoefte aan	N=	583	5699
hebben (*have*)	●	73.6	53.8
zijn (*be*)	●	19.0	24.5
bestaan (*exist*)		4.6	4.6
blijken (*turn out to be*)		0.5	0.5
blijven (*remain*)		0.5	0.6
toenemen (*increase*)	●	0.3	1.7
belangstelling voor	N=	428	5124
hebben (*have*)	●	33.4	28.4
zijn (*be*)	●	31.8	23.5
bestaan (*exist*)	●	20.6	5.2
tonen (*show*)	○	4.9	7.4
komen (*come*)		1.4	1.1
blijken (*turn out to be*)		0.9	0.8
verwachten (*expect*)		0.7	0.4
ontstaan (*come up*)		0.5	1.1
blijven (*remain*)		0.5	0.5

		Split	**Gen**
discussie over	N=	164	3857
zijn (*be*)	●	40.2	15.3
bestaan (*exist*)	●	10.4	0.7
voeren (*be engaged in*)		10.4	7.3
gaan (*go*)		9.8	7.4
hebben (*have*)	●	5.5	2.0
woeden (*rage*)		4.3	2.4
ontstaan (*come up*)		4.3	3.7
losbarsten (*burst out*)		2.4	1.9
ontbranden (*ignite*)		1.2	0.6
houden (*hold*)	●	1.8	0.6
gebrek aan	N=	303	2574
zijn (*be*)	●	62.4	32.8
hebben (*have*)	●	28.4	9.1
bestaan (*exist*)	●	2.6	0.6
liggen (*lay*)		1.0	0.3
heersen (*rule*)		1.0	0.5
lijken (*seem*)		0.7	0.5

Table 2: Distribution of verbs for several N+P collocations. Differences marked with • (○) are significant according to the χ^2 test at p=0.05 (p=0.10).

tigated N+P combinations. In Haesereyn *et al.* (1997), Broekhuis (2004) and Loonen (2003) (non-exhaustive) lists of phrasal verbs involving a PP-complement are given. A considerable number of V+N+P combinations in the *split* corpus are presented as phrasal verbs in at least one of these sources, e.g. *een gesprek voeren met (be engaged in a conversation with), informatie verstrekken over (provide information on), een onderzoek instellen naar (start an investigation into), een onderzoek loopt naar (an investigation is being carried out into), protest rijst tegen (protest is raised against), een verhaal gaat over (a story is about), een verhaal doet (de ronde) over (a story goes around about)*, en *(er) zit een verschil tussen (there is a difference between).*

The V+PP analysis also predicts that for some verbs, PP-fronting should be impossible. This prediction is hard to test, as the absence of a verb in *split* might be due to lack of data. Nevertheless, in table 3 we provide a list of verbs missing in *split* which occur with more than 1% of the relevant N+P example sentences in the general corpus. All verbs listed for *gebrek aan* seem to resist PP-fronting. In other cases, fronting seems marked (*aan* NP *groeit er behoefte (for* NP *grows the demand), naar* NP *leidt/eist* NP *een onderzoek (into* NP, NP *demands an in-*

behoefte aan	Split=583 G=5699	
onstaan (*come up*)	•	1.4
groeien (*grow*)	•	1.3

belangstelling voor	Split=428 G=5124	
wekken (*wake*)	•	1.1

discussie over	Split=164 G=3857	
beginnen (*start*)	•	2.7
aanzwengelen (*start up*)	○	1.8
volgen (*follow*)		1.0
aangaan (*engage in*)		1.0
krijgen (*get*)		1.0

gebrek aan	Split=303 G=2574	
verwijten (*blame*)	•	8.7
compenseren (*compensate*)	•	2.4
opbreken (*stumble over*)	•	1.6
noemen (*mention*)	•	1.6
leiden (*lead to*)	•	1.5
vinden (*find*)	•	1.3
spelen (*play*)	•	1.3
hekelen (*criticize*)	○	1.2
worden (*become*)	○	1.0

kritiek op	Split=215 G=4077	
toenemen (*increase*)	○	1.3

onderzoek naar	Split=228 G=3570	
leiden (*lead*)	•	1.8
willen (*want*)	•	1.8
gelasten (*demand*)	○	1.5
eisen (*demand*)	○	1.2
aankondigen (*announce*)		1.1

twijfel over	Split=154 G=1714	
uiten (*utter*)	•	2.5
wegnemen (*take away*)	○	1.9
groeien (*grow*)		1.6

verschil tussen	Split=130 G=3925	
bedragen (*amount to*)	○	2.7
worden (*become*)		2.0
weten (*know*)		1.8
kennen (*know*)		1.5

Table 3: Frequent N-P-V-combinations in the general corpus (G), absent in *split*. Differences marked with • (○) are significant according to the χ^2 test at p=0.05 (p=0.10).

vestigation), naar NP *kondigt* NP *een onderzoek aan (into* NP, NP *announces an investigation), over* NP *nam* NP *alle twijfel weg (on* NP, NP *took all doubts away)*). For other verbs and N+P combinations, it seems that fronting is at least theoretically possible. The limited size of the *split* corpus might be the reason why these are absent in our data.

The corpus data clearly suggest that the verb plays a role in the possibility of PP-fronting and discontiuous PPs. The distribution of verbs in *split* and *general* shows large differences for most investigated N+P combinations. For frequent verbs, these differences are often statistically significant. Furthermore, there seem to be a number of verbs which easily combine with certain N+P combinations, but which do not allow *split* configurations.

	Split	Gen
determiner N=	3.601	50.892
geen (*no*)	30.7	8.0
NULL	27.7	31.8
een (*a*)	14.4	16.5
veel (*many/much*)	7.7	2.1
de/het (*the*)	7.3	32.9
weinig (*few/little*)	3.8	0.7
enkele (*some*)	2.1	0.8
meer (*more*)	0.8	1.0
minder(*less*)	0.6	0.2

Table 4: Frequency of determiners preceding the relevant noun in *split* and the general corpus.

4 The role of the determiner

It has been argued that so-called *specified subjects* within the NP block extraction:

(5) a. Over Piet herinnerde hij zich een verhaal.
About Piet remembered he REFL a story
He remembered a story about Piet
b. *Over Piet herinnerde hij zich Jans verhaal.
About Piet remembered he REFL Jan's story

An NP contains a specified subject if its determiner is a genitive NP or a possessive pronoun. The existence of a constraint like this would be a strong argument for the N+PP analysis. In this section, we investigate whether there is a relationship between the distribution of determiners and PP-fronting.

In table 4, a comparison of the frequencies in *split* and *general* is made of the most common determiners preceding the relevant noun. The indefinite determiners *geen*, *veel* and *weinig* occur relatively frequently in *split*, whereas the definite determiner *de/het* is relatively infrequent in *split*.

We believe that the difference in distribution of determiners in *split* and the *general* corpus can be explained to a large extent by the fact that the verbs in *split* and *general* have a very different distribution (as shown in the previous section). If we restrict attention to N-P-V combinations that contain a verb which is relatively frequent in *split*, we see that the definite determiner is much less frequent in *general* as well. This is illustrated in table 5.

At first sight, the corpus seems to confirm the observation that PP-fronting requires an NP which does not contain a 'specified subject' in the form of a possessive pronoun or genitive NP. Table 4 does not contain any of these determiners. Genitives are in fact absent in *split*, while possessives are scarce, and restricted to the N+P combinations *verhaal over* en *twijfel over*:

N+V+P	N=	determiners
behoefte hebben aan *have need for*	3001	NULL 60.4, geen 25.5, ...,de 1.0
behoefte zijn aan *be need for*	1051	NULL 71.6, een 10.5, geen 5.6, ..., de 2.1
behoefte bestaan aan *exist need for*	259	NULL 52.1, een 18.9 geen 11.6, de 5.8
belangstelling hebben voor *have interest in*	1343	NULL 70.3, geen 12.9 ..., de 0.5
bezwaar hebben tegen *have objection against*	1431	geen 53.9, NULL 34.2 ..., het 0.0
contact zoeken met *seek contact with*	462	NULL 93.9, geen 4.1 het 1.1
discussie zijn over *be discussion about*	257	NULL 36.6, de 16.3 geen 14.0, een 12.8
gesprek voeren met *be engaged in discussion with*	250	een 90, het 4.8 geen 1.6

Table 5: Frequency of common indefinite and definite determiners in the general corpus for frequent N-P-V-combinations in *split*.

(6) a. Over die worsteling gaat mijn verhaal
about that struggle goes my story
My story is about that struggle

b. Over adverteren in de verzorgingssfeer heeft hij zijn twijfels
About advertising in the health sector has he his doubts
He has his doubts about advertising in the health sector

Only the phrase *twijfels hebben over* is relatively frequent in *split*.

One might argue that the absence of genitives and the apparently highly restricted use of possessives, is evidence for the claim that PP-fronting is blocked for certain NPs. It should be noted, however, that NPs introduced by a possessive pronoun or genitive are not very frequent in the *general* corpus either: 2.1% of the relevant NPs in *general* contains a possessive pronoun and 0.9% a genitive NP. Furthermore, those verbs which do occur with this type of NP seem to be highly infrequent in *split*.

The preference for indefinite determiners in the *split* data correlates strongly with the preference for indefinite determiners in the general corpus, if one restricts attention to those verbs which are frequent in *split*. Furthermore, the absence of NPs introduced by a genitive and the restricted possibilities for using possessive pronouns seems to be a consequence of the fact that these are scarce in general, especially if one also takes the verb into account. It seems therefore that the differences in determiner distribution are for the most part a consequence of the differences in the distribution of the verbs in both corpora.

5 Related Corpus Observations

In this section we briefly discuss various corpus observations that are relevant for the analysis of PP-fronting.

We encountered one construction which has not been discussed in the literature but which seems problematic for an N+PP analysis. In relative clauses modifying the noun, the PP is sometimes clearly embedded in the relative clause (7). For PPs which are unambiguously part of the NP (and which cannot be fronted) this is not possible (8).

(7) a. De **kritiek** die hier **op** het boek wordt uitgeoefend
the critique that here on the book is offered
the critique on the book which is offered here

b. de **belangstelling** die Eduard **voor** het nazisme toonde
the interest which Eduard for the nazism showed
the interest which Eduard showed for Nazism

(8) *de **demonstratie** die **tegen** de hoge werkdruk in chaos ontaardde
the demonstration which agains the high work-load in chaos ended

Thus, the possibility of a PP to appear inside a relative clause is evidence for the fact that the PP can be interpreted as a dependent of the verb.

In the general corpus, for most of the N+P combinations we investigated, several examples can be found where the PP is included in a relative clause. This seems problematic for a N+PP analysis. Under such an analysis, it seems that the relative pronoun would have to inherit the selection or subcategorization properties of the noun it modifies. Furthermore, a mechanism needs to be established which allows the PP to appear in a position non-adjacent to the relative pronoun (i.e. head-movement, remnant movement, or argument transfer from the pronoun to the verbal head). We believe the syntactic literature does not provide evidence for assuming that such processes are at work here.

One argument for the V+PP analysis has been the suggestion that one also finds cases of ‘NP-fronting’, where the PP occupies a position in the ‘Mittelfeld’:

(9) Een **roman** heb ik **van** Vestdijk gelezen
A novel have I of Vestdijk read
I have read a novel by Vestdijk

Such examples are practically absent (i.e. we were able to find only 3 convincing examples) in the general corpus. The difference in frequency between PP-fronting and ‘NP-fronting’ is puzzling.

Another argument for the V+PP analysis has been the claim that the NP and PP may be separated from each other within the Mittelfeld. In the general corpus, we did not find a single example of an NP-XP-PP word order, however. We found only 4 examples of PP-XP-NP word order. On the other hand, PP-NP orders, as in (10), are relatively common in the general corpus (with 10-50 examples per N-P combination, except for *protest tegen*, for which we found only a single example):

(10) De Marokkanen hebben **aan** groepsvorming geen **behoefte**.
The Maroccans have on group formation little interest
the Maroccans have little interest in such group formation

Although this pattern seems equally problematic for an N+PP analysis as PP-XP-NP order, it has not been mentioned as such in the literature.

6 Concluding Remarks

Corpus investigation suggests that PP-fronting and discontinuous PPs are best analyzed as involving a PP which is a dependent of the verb. Certain verbs are far more frequent in *split* sentences than in general sentences containing the relevant N+P combination. The difference in the distribution of determiners in the *split* and *general* corpus seems to be mainly a consequence of the difference in distribution of verbs in both corpora. The corpus also contains a fair number of sentences containing PPs within relative clauses of the noun. Such examples seem problematic for an N+PP analysis. A number of patterns which have been used as argument for a specific analysis of PP-fronting are hardly encountered in a large corpus, except for PP-NP patterns.

The strong semantic relation between the noun and the preposition suggests that the examples we have investigated are examples of *phrasal verbs*, involving a verb with a more or less fixed NP-complement and a PP-complement. A similar conclusion was reached by Coppen (1991), who argues for an analysis which treats the PP as an argument selected by the combination of NP+V, i.e. a (pseudo) phrasal verb.

References

[1] E. Bach and G. Horn. Remarks on conditions on transformations. *Linguistic Inquiry*, 7:265–299, 1976.

[2] Gosse Bouma and Geert Kloosterman. Querying dependency treebanks in XML. In *Proceedings of the Third international conference on Language Resources and Evaluation (LREC)*, Gran Canaria, 2002.

[3] Hans Broekhuis. Het voorzetselvoorwerp. *Nederlandse Taalkunde*, 9(2), 2004.

[4] Peter-Arno Coppen. Over vooropstaande PP's is het laatste woord nog niet gesproken. *Gramma*, 15(3):209–225, 1991.

[5] Kordola de Kuthy. Splitting PPs from NPs. In Tibor Kiss and Detmar Meurers, editors, *Constraint-Based Approaches to Germanic Syntax*, pages 25–70. CSLI Publications, Stanford University, 2000.

[6] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational linguistics*, 19(1):61—74, 1993.

[7] W. Haesereyn, K. Romijn, G. Geerts, J. De Rooy, and M.C. Van den Toorn. *Algemene Nederlandse Spraakkunst*. Martinus Nijhoff Uitgevers Groningen / Wolters Plantyn Deurne, 1997. Tweede, geheel herziene druk.

[8] M. Klein and M.C. van den Toorn. Van NP-beperking tot XP-beperking; een antwoord op Kooij en Wiers 1978. *De Nieuwe Taalgids*, 72:97–102, 1979.

[9] J. Kooij and E. Wiers. Beperkingen en overschrijdingen: een antwoord aan Klein en Van den Toorn. *De Nieuwe Taalgids*, 72:488–493, 1979.

[10] L.J.M. Loonen. *Stante pede gaande van dichtbij langs AF bestemming @*. PhD thesis, Universiteit Utrecht, Utrecht, 2003.

[11] Robert Malouf and Gertjan van Noord. Wide coverage parsing with stochastic attribute value grammars. In *IJCNLP-04 Workshop Beyond Shallow Analyses - Formalisms and statistical modeling for deep analyses*, Hainan, 2004.

[12] Michael Moortgat, Ineke Schuurman, and Ton van der Wouden. CGN syntactische annotatie, 2000. Internal Project Report Corpus Gesproken Nederlands, see http://lands.let.kun.nl/cgn.

[13] L. van der Beek, G. Bouma, R. Malouf, and G. van Noord. The alpino dependency treebank. In *Computational Linguistics in the Netherlands (CLIN) 2001*, Twente University, 2002.

CAT3LB: a Treebank for Catalan with Word Sense Annotation

Montserrat Civit, Núria Bufí, M. Pilar Valverde
CLiC Centre de Llenguatge i Computació – Universitat de Barcelona
Adolf Florensa s/n (Torre Florensa) 08028 Barcelona
{civit, nuria, pilar}@thera-clic.com

1 Introduction

It is widely admitted that Treebanks constitute a crucial resource both to develop NLP applications and to acquire linguistic knowledge about how a language is used. As for minority languages, a new parameter has to be taken into account when we analyse their normalization degree: that of their presence or their absence in Internet and the number and kind of computational resources and tools it processes.

In this paper[1] we present the work done in the building of CAT3LB, a Treebank for Catalan with word sense annotation, which is part of the 3LB Project[2]. The paper is organised as follows: section 2 deals with previous processes (morphological annotation, tagging and chunking); section 3 deals with the syntactic annotation itself (constituents and functions); section 4 deals with word sense annotation; and section 5 presents some conclusions.

2 Previous processes

CLiC-UB [3] and TALP-UPC [4] groups have developed so far a framework for the automatic processing of Catalan and Spanish, based in a pipeline structure[7]. Firstly, the raw text is morphologically analysed with MACO (see section 2.1); secondly,

[1] We would like to thank the three reviewers for their useful and interesting comments about the abstract: they have helped us a lot in the writing of the full paper.

[2] This research has been partially funded by the Spanish Research Department: X-Tract2 (BFF2002-04226-C03-03) and PROFIT (FIT-150500-2002-244).

[3] URL:http://clic.fil.ub.es/index_en.shtml

[4] URL:http://www.talp.upc.es/TALPAngles/index.html

it is disambiguated with RELAX (see section 2.2); and finally, it is chunked with TACAT and a handwritten grammar for Catalan (see section 2.3).

2.1 Morphological Analysis

MACO is a Morphological Analyser for Catalan, Spanish and English that provides both lemma(s) and POS-tag(s) for each word and whose output has the following form:

$$word \quad lemma_1 - tag_1 \quad ... \quad lemma_n - tag_n$$

The tagset for Catalan codifies 13 part-of-speech categories (noun, verb, adjective, adverb, pronoun, determiner, preposition, conjunction, interjection, dates, punctuation marks, numbers and abbreviations) as well as subcategories and morphological features, as it is proposed by Eagles [13]. The total amount of tags is 321[5].

2.2 Morphological Tagging

Once the text has been morphologically annotated, RELAX, a constraint-based probabilistic tagger [17], selects the best pair lemma-tag; Relax is trained from manually annotated texts and it allows the introduction of manually written constraints. The accuracy of the output varies between 94-96%, but it is increased up to 95-97% after the introduction of handwritten constraints [11].

2.3 Chunking

Finally, the chunking is done with TACAT [4] and a context free grammar for Catalan of about 1920 handwritten rules. Catalan has a rich inflexional morphology, so the concept of chunk can be extended (i.e. we use a larger conception of chunk than Abney's, [2], [3]): the grammar puts together words if, according to their form, one can be sure they go together. For instance, a noun phrase may include:

$$[(Determiner) + (adjective) + head + (noun|AdjP|PP_{de})]$$

that is: an (optional) determiner in a pre-head position; an (optional) adjective before the head; and another element after the noun: either another noun, or an adjective or a prepositional phrase headed by the preposition *de*[6].

[5] They can be found, with a large explanation, at http://clic.fil.ub.es/doc/categorias_eagles_cat03.htm

[6] Meaning 'of' or 'from'. This PP was included in the nominal chunk after a detailed analysis of about 300 examples: the PP was almost always depending on the immediately preceding noun.

This extended conception of chunk is similar to that proposed in [15] and largely reduces the annotation time, even if it is not error free (this chunking produces some errors in the analysis of the PP_{de} attachment (1-2%) but we consider that it is assumable if, in contrast, it largely reduces the annotator's work).

3 Syntactic annotation

Catalan is a romance, pro-drop language in which the constituent order is quite free and, within the constituents, the word order is quite fixed (i.e. in the noun phrase, for instance, adjectives can precede or follow the noun, but the most frequent case is the postposition; the determiner is always the first word in such a phrase; it is extremely rare to find two adjectives preceding the noun head; all relative clauses and prepositional phrases follow the head; etc.). Linguistic tradition deals with Catalan in terms of constituents ([14], [5]) and we do so; moreover, in the sentence, movement phenomena are basically related to constituents.

In order to do the syntactic annotation, we used the AGTK interface [12] developed at the University of Pennsylvania. It has been slightly modified in order to allow both the processing of special characters and the processing of xml text format[7].

3.1 Constituents

In a first step, 25.000 words[8] were syntactically annotated in parallel by two linguists. This first annotation was then used, on the one hand, to refine the annotation criteria and, on the other, to enlarge the annotation guidelines previously established [9]. The comparison between the two annotations gave the results shown in table 1, in which **LP** stands for *labelled precision*; **BP** for *bracketed precision* and **CB** for *consistent bracketing*[10].

One of the main sources of disagreement was whether to consider as a single word certain complex structures, like *posar èmfasi*[11]. As annotators adopted different criteria, the length of the final sentence was different from one annotation to the other. Since our agreement measures take into account the starting and finishing points of each constituent in the sentence, the fact that the length of the sentence varied implied a substantial decrease of the results. This issue was accurately analysed and very strict criteria were established in the guidelines to deal

[7] Trees are stored in two formats: bracketed text, like the Penn TreeBank and xml format.

[8] This corresponds to 640 sentences and the average number of words per sentence is 39.

[9] [19] is the last version of the guidelines for the constituent annotation.

[10] As they are used in Parseval.

[11] To emphasise.

LP	0.876478
BP	0.90953004
CB	0.943214
same-length Sentences	
LP	0.9198125
BP	0.93964505
CB	0.96512

Table 1: Annotators' agreement (1)

with multiwords. In order to evaluate the annotators' agreement for those first sentences, we also carried out the evaluation taking into account only the sentences having the same length.

As for concrete aspects of the annotation, we would like to point out some of the most significant issues: types of sentences, coordinated structures and discontinuity. As the Cat3LB treebank has been developed within a larger project (**3LB**) and after the development of the Spanish Treebank (Cast3LB) [9], [10], we have take advantage of the previous annotation process.

The root node of sentences is always **S**, standing for *sentence*. If the sentence has no verb, then the tag is **S***. Regarding clause types, we distinguish finite (**S.F.**) and non-finite clauses (**S.NF.**), on the one hand, and, on the other, completive (**S.F.C, S.NF.C**), relative (**S.F.R**) and adverbial ones (**S.F.A, S.NF.A**). Finite adverbial clauses, moreover, are splitted into three groups: those considered as being a verbal adjunct (namely those meaning time, place, cause, purpose or manner **S.F.A**), those considered to be adjuncts of the predicate (conditional, concessive and consecutive ones **S.F.ACond, S.F.AConc, S.F.ACons**) and those being adjuncts of a noun or an adjective, the comparative clauses (**S.F.AComp**).

We pay special attention to the treatment of coordinated structures: we consider coordinated elements to be equivalent in the syntactic structure[12], so they are represented as siblings, which means that there is no head in such constructions. Shared complements are another issue related to coordination (i.e.: complements shared by two or more verbs); in these cases our solution is to adjoin the complement to the coordinated node.

Cases of discontinuity have been dealt with in two different ways: some of them at the constituent level, and the others at the function one (see section 3.2). Discontinuity dealt with in the first level is mostly related to the noun phrase and

[12] The solution in [1] is completely different, since they consider the first element to be the head in the coordinated nodes.

involves a noun complement which is separated from the head by a (verbal) complement; in this case, the separated complement formally depends, in the representation on the nearest S node, but the **.1** index marks where it must be interpretated. An example of this situation appears in the sentence *en detectar-se la presència d'un brot infecciós a principis del mes de maig que va afectar 12 malalts*[13] in which the relative clause *que va afectar 12 malalts* depends on the noun *brot* but is separated from it by a verbal complement *a principis del mes de maig*. In this case, we add an index **.1** to both elements involved in the discontinuity, so that the resulting analysis is as it appears in figure 1.

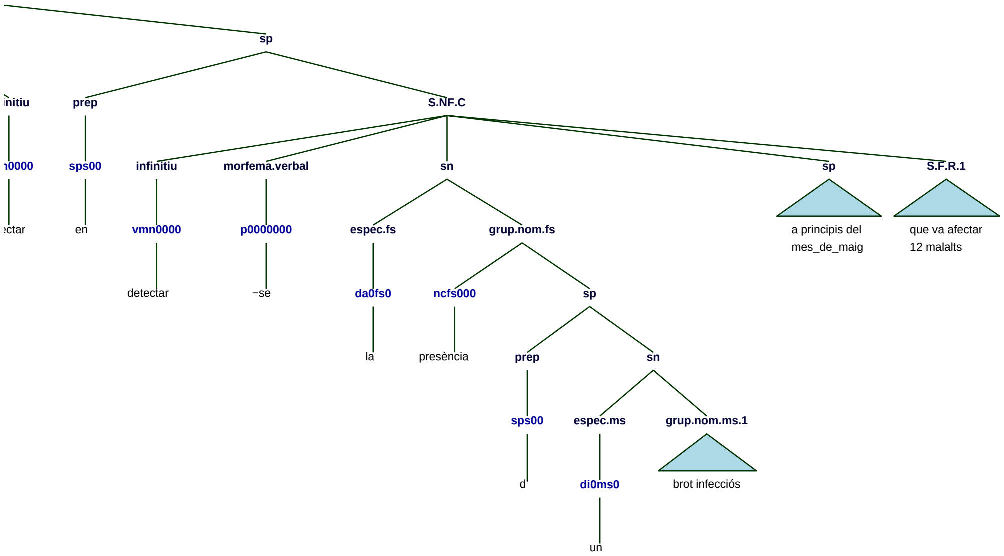


Figure 1: Constituent Discontinuity (1)

3.2 Functions

We have extended our constituency-based scheme with the annotation of grammatical functions (as it was done in the Susanne Corpus [18] or in the Penn Treebank II [16]), that is, by adding functional tags to the phrase structure annotation[14].

[13] When detecting the presence of an infectious outbreak at the beginning of May that infected 12 sick people.

[14] [8] is the guidelines for the functional annotation

A quantitative analysis of the annotators' agreement was done, as it was for the constituent annotation. The results are shown in table 2. In this case, we only consider the labelled precision, because the annotators worked over the previous constituent annotation, so the bracketing was the same. This evaluation was done in two different times: at the first one, the guidelines were not yet complete, while at the second they were.

First phase: 698 sentences	
LP	0.9009
Second phase: 45 sentences	
LP	0.94915254

Table 2: Annotators' agreement (2)

Most of the discrepancies were due to errors in the annotation (i.e. one annotator forgot to put the functional tag or did not apply correctly what was said in the guidelines).

Only daughter nodes of sentences and clauses are given a functional tag (i.e. we do not deal with noun complements). We have established a set of 14 basic tags (see table 3), in order to cover all syntactic functions, and then, given specific marks (tag suffixes) to some of them in order to annotate specific cases of these functions. All in all, the total amount of tags at this level is 58. Basic tags are shown in table 3.

Tag	**Gloss**	**Tag**	**Gloss**
-SUJ	subject of a finite verb form	-AO	Sentence Adjunct
-CD	Direct Object	-ET	Textual Element
-CI	Indirect Object	-MOD	Modifier
-ATR	Attribute	-PASS	Passive Mark
-CPRED	Predicative Complement	-IMPERS	Impersonal Mark
-CREG	Prepositional Object	-VOC	Vocative
-CAG	Agent		
-CC	Circumstance		

Table 3: Basic Functional Tagset

The remaining cases of discontinuity are dealt with in the functional tagging. There are two of such cases. The first case is related to clitics and the second is related to raising movement.

When the direct object (with ergative verbs it may happen with the subject too) is an undetermined noun phrase (i.e. a noun phrase with an indefinite article or a quantifier), the substitution by the clitic is partial, and only the noun is replaced by the clitic, but not the determiner; so the direct object is splitted into two elements, one before and the other after the verb. For such cases, we have created a special tag suffix (**.d**), which appears in both the two elements. Figure 2 shows one of these cases. The sentence is *dels quals només se'n conserven dos*[15]

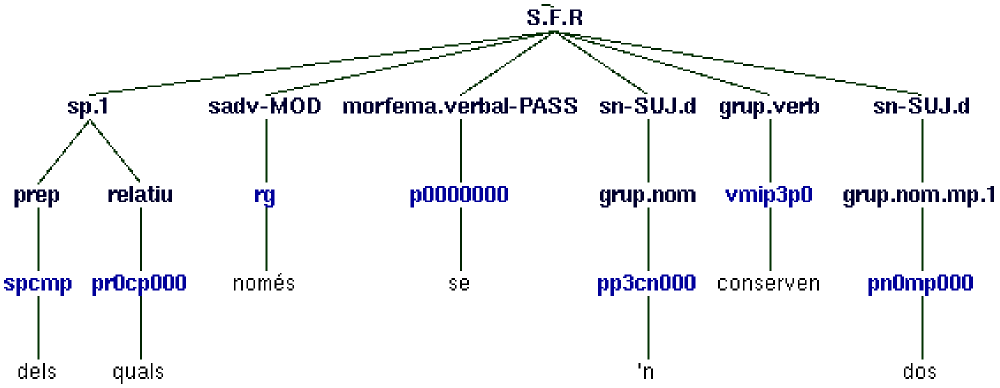


Figure 2: Constituent Discontinuity (2)

Another case of discontinuity appears in relative or interrogative clauses, in which the relative (or interrogative) pronoun of a (non-)finite clause raises to the first position of the sentence: *dels pagesos que hi vulguin anar* (figure 3)[16], in which the selected locative complement (*hi*) belongs to the non-finite clause *anar* but appears before the main verb. For these cases, the functional tag has a suffix **.F** or **.NF** (depending on the type of the clause -finite or non-finite-) and the whole tag must be read as follows: *complement of the first finite or non-finite clause to the right*.

In Catalan it is possible for a complement to appear twice in the sentence. It usually happens with direct an indirect objects (but also with other verb complements), when the phrase goes before the verb and it has to be repeated by a clitic[17]. This is related to the inversion of constituents in the sentence: the most usual word

[15] Literal: from which only [passive mark] [clitic] survive two
Translation: from which only two survive.

[16] Lit: from farmers who [locative-clitic] want to go; translation: from farmers who want to go there

[17] If there is no repetition the sentence is considered to be ungrammatical.

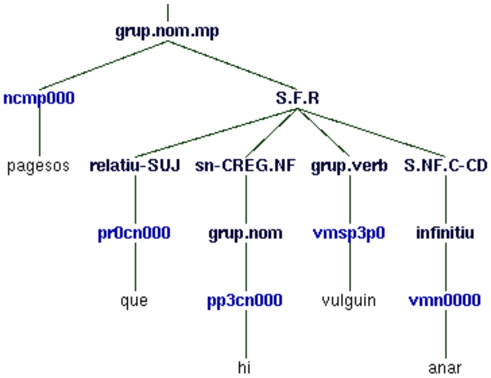


Figure 3: Constituent Discontinuity (3)

order in Catalan is SVO, and when it is inverted (OVS) we need to mark the inversion, so the complement is repeated by a clitic. In these cases we add a suffix **.r** to the function tag. An example of such phenomenon is shown in the sentence *El rànquing l' encapçala la final de la Champions_League*[18], in which the direct object appears twice at the beginning of the sentence (see figure 4).

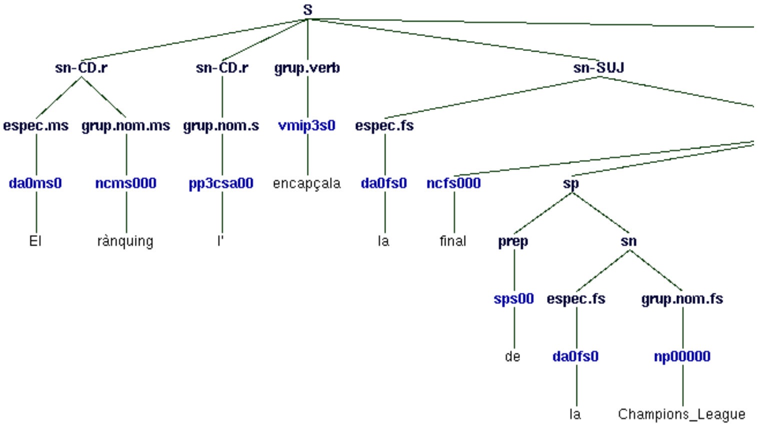


Figure 4: Doubled functions

One of the most controversial points related to functional tagging has been the

[18] Cat: El rànquing-CD l'-CD encapçala [la final de la Champions_League]-SUBJ
Lit.: 'The ranking-CD [clitic]-CD heads [the final of the Champions League]-SUBJ'
translation: 'the final of the Champions League heads the ranking'

distinction between prepositional complements selected or not by the verb. Linguistic criteria are not unanimous, especially those concerning the obligatoriness of the complement. It usually happens that locative complements are mandatory. This clearly appears when the answer to the question *Anirem al cine demà?*[19] has to contain the locative clitic *hi*: *hi anirem*[20]. Bearing in mind the state of the art about this point, we decided to give the adverbial tag (**-CC**) to those elements being optional, while the function tag **-CREG**, standing for 'selected PP' is used for the mandatory complements, no matter whether they are locative or not.

4 Semantic annotation

In a last step, a subset of the corpus of 10,000 words has been annotated with CatalanWordNet. Only nouns, verbs, and adjectives receive a semantic tag. In the annotation process words were annotated throughout the whole corpus, so as to ensure the consistency of their annotation.

The total amount of nouns in the subset of the corpus is 841 lemmas (some of them appearing 33, 30, 28 times; others appearing only once), 380 adjectives and 403 verbs. The most frequent nouns were *grup, govern, any, empresa, president*[21]; the most frequent adjectives were *català, nou, passat, polític, socialista*[22]; finally the most common verbs were *tenir, estar, presentar, poder, fer, donar*[23]. Some verbs were not annotated when they were the auxiliary for the compound tenses o complex verbal forms.

In order to do the annotation, we took a version of the EuroWordNet 1.5, that of December 2002, and built an interface to help the annotators: 3LB-SAT [6]. CatalanWordNet has 28,575 synsets (20,260 for nouns, 4,415 for adjectives and 3,900 for verbs). The ambiguity average is 1.790 senses per lemma if we considers all variants, and 3.182 if we consider only ambiguous lemmas (i.e. lemmas appearing in two or more synsets).

The main problem in the semantic annotation was that CatalanWordNet is incomplete and has not been extensively revised. For instance, for the word *president*, which referred in most of the sentences to the president of the Catalan Parliament or the president of a football team, it was impossible to assign a sense, because in CatalanWordNet there are only presidents for Republics, companies, meetings or the United States. Figure 5 shows the interface with the possibilities

[19] Will we go to the cinema tomorrow?

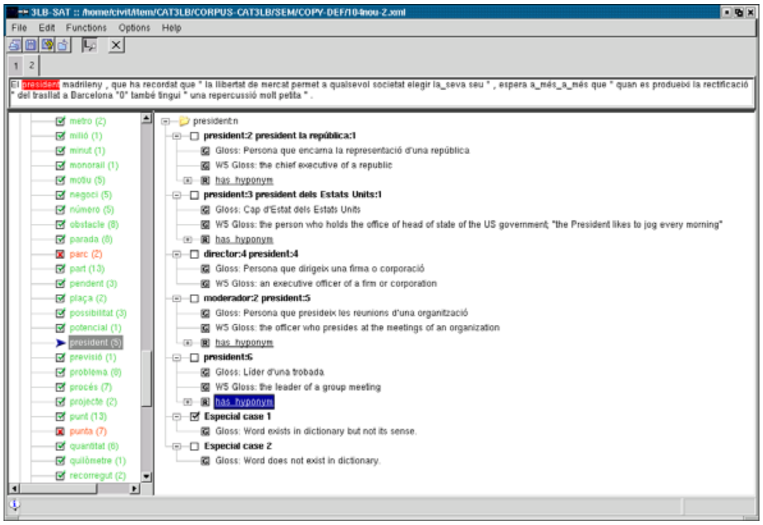
[20] We [clitic] will.

[21] Group, government, year, enterprise/business, president

[22] Catalan, new/nine, last/passed, political, socialist

[23] to have/to own, to be, to present/to introduce, to be able to/can, to do/to make, to give

for the word *president* displayed.


3LB-SAT :: /home/civit/Item/CAT3LB/CORPUS-CAT3LB/SEM/COPY-DEF/104nou-2.xml
File
Edit
Functions
Options
Help
1
2

El president madrileny , que ha recordat que " la llibertat de mercat permet a qualsevol societat elegir la_seva seu " , espera a_més_a_més que " quan es produeixi la rectificació " del trasllat a Barcelona "0" també tingui " una repercussió molt petita " .

```
☑ metro (2)           president:n
☑ milió (1)             □ president:2 president la república:1
☑ minut (1)                 Gloss: Persona que encarna la representació d'una república
☑ monorail (1)              WS Gloss: the chief executive of a republic
☑ motiu (5)                 has_hyponym
☑ negoci (5)            □ president:3 president dels Estats Units:1
☑ número (5)                Gloss: Cap d'Estat dels Estats Units
☑ obstacle (8)              WS Gloss: the person who holds the office of head of state of the US government; "the President likes to jog every morning"
☑ parada (6)                has_hyponym
☒ parc (2)              □ director:4 president:4
☑ part (13)                 Gloss: Persona que dirigeix una firma o corporació
☑ pendent (3)               WS Gloss: an executive officer of a firm or corporation
☑ plaça (2)             □ moderador:2 president:5
☑ possibilitat (3)          Gloss: Persona que presideix les reunions d'una organització
☑ potencial (1)             WS Gloss: the officer who presides at the meetings of an organization
➤ president (5)             has_hyponym
☑ previsió (1)          □ president:6
☑ problema (8)              Gloss: Líder d'una trobada
☑ procés (7)                WS Gloss: the leader of a group meeting
☑ projecte (2)              has_hyponym
☑ punt (13)             ☑ Especial case 1
☒ punta (7)                 Gloss: Word exists in dictionary but not its sense.
☑ quantitat (6)         □ Especial case 2
☑ quilòmetre (1)            Gloss: Word does not exist in dictionary.
☑ recorregut (2)
```

Figure 5: Semantic Annotation Tool

We added two more possibilities to the word sense annotation: **EC1** and **EC2**. EC1 stands for those cases in which the word appears in CatalaWordNet but not its sense; EC2 was thought to mark those words that did not appear in the CatalanWordNet.

The word sense annotation was done semiautomatically; on the one hand, the tag EC2 was assigned automatically if the word did not appear in CatalanWordNet; on the other hand, words being monosemous in CatalaWordNet were given the sense in an automatic way, but this assignation was manually checked because the given sense could not be the right one. The rest of the annotation was done manually. It was not possible to do any preliminary automatic annotation because there is no information in CatalanWordNet about the most frequent sense of the words.

5 Conclusions

We have presented the development of Cat3LB, a Treebank for Catalan. We have shown the main issues in both the syntactic and semantic annotation processes. One of the open issues, now, is to start the revision of the CatalanWordnet: we plan to develop it in parallel with the corpus annotation in order to add the synsets that do not appear at present, but also in order to modify its structure by removing unnecessary synsets or compact some others. We think that this parallel work is the best both for the corpora annotation process and the enrichment of the semantic net.

References

[1] A. Abeillé, F. Toussenel, and M. Chéradame. Corpus le Monde. Annotation en constituants. Guide pour les correcteurs. Technical report, LLF, UFRL, 2002. dernière mise à jour: 10-juillet-2002.

[2] S. Abney. Parsing by Chunks. In R. Berwick, S. Abney, and C. Tenny, editors, *Principle-Based Parsing*. Kluwer Academic, 1991. available: http://www.sfs.nphil.uni-tuebingen.de/ abney/.

[3] S. Abney. Partial Parsing via Finite-State Cascades. In *Proceedings of the ESSLLI'96 Robust Parsing Workshop*, 1996. availible: http://www.sfs.nphil.uni-tuebingen.de/ abney/.

[4] J. Atserias and H. Rodríguez. TACAT: TAgged Corpus Text Analyzer. Technical report, Software Department (LSI). Technical University of Catalonia (UPC), 1998.

[5] A. Bel. Les funcions sintàctiques. In J. Solà, M.R. Lloret, J. Mascaró, and M. Pérez, editors, *Gramàtica del català contemporani*, chapter S-2, pages 1075–1147. Empúries, 2002.

[6] E. Bisbal, A. Molina, L. Moreno, F. Pla, M. Saiz-Noeda, and E. Sanchís. 3LB-SAT: Una herramienta de anotación semántica. In *Procesamiento del Lenguaje Natural*, number 31, pages 193–199, Alcalá de Henares, 2003.

[7] J. Carmona, S. Cervell, L. Màrquez, M.A. Martí, L. Padró, R. Placer, H. Rodríguez, M. Taulé, and J. Turmo. An Environment for Morphosyntactic Processing of Unrestricted Spanish Text. In *Proceedings of the First Conference on Language Resources and Avaluation. LREC'98*, pages 915–922, Granada, 1998.

[8] M. Civit, N. Bufí, and M.P. Valverde. Guia per a l'anotació de les funcions sintàctiques de cat3lb: un corpus del català amb anotació sintàctica, semàntica i pragmàtica. Technical report, CLiC, 2004. available: http://www.clic.fil.ub.es/personal/civit/publicacions.html.

[9] M. Civit and M.A. Martí. Design Principles for a Spanish Treebank. In *Proceedings of the First Workshop on Treebanks and Linguistics Theories (TLT2002)*, pages 61–77, Sozopol, September 2002. available: http://clic.fil.ub.es/personal/civit.

[10] M. Civit, M.A. Martí, B.Navarro, N. Bufí, B. Fernández, and R. Marcos. Issues in the Syntactic Annotation of Cast3LB. In *Proceedings of the LINC03 Workshop*, Budapest, 2003. available: http://clic.fil.ub.es/personal/civit.

[11] L. Cots. Restriccions manuals de desambiguació en el corpus CLiC-TALP-CAT. Master's thesis, Universitat de Barcelona, Dpt. de Lingüística General, 2004.

[12] S. Cotton and S. Bird. An integrated Framework for Treebanks and Multilayer Annotations. In *Proceedings of the Second International Conference on Language and Evaluation LREC-2000*, Athens, Greece, 2000.

[13] EAGLES. Synopsis and Comparison of Morphosyntactic Phenomena Encoded in Lexicons and Corpora. A common Proposal and Applications to European Languages. EAG—CLWG—MORPHSYN/R, 1996. available: http://www.ilc.pi.cnr.it/EAGLES96/browse.html.

[14] M. L. Hernanz. L'oració. In J. Solà, M.R. Lloret, J. Mascaró, and M. Pérez, editors, *Gramàtica del català contemporani*, chapter S-1, pages 993–1073. Empúries, 2002.

[15] H. Kermes and S. Evert. Text analysis meets corpus linguistics. In *Proceedings of the Corpus Linguistics 2003*, Lancaster, UK, 2003.

[16] M. Marcus, G. Kim, M.A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. The Penn Treebank: Annotating Predicate Argument Structure. In *Proceedings of the ARPA Human Language Technology Workshop*, Princeton, New Jersey, March 1994.

[17] L. Padró. *A Hybrid Environment for Syntax-Semantic Tagging*. PhD thesis, Software Department (LSI). Technical University of Catalonia (UPC), 1998.

[18] G. Sampson. *English for the Computer. The SUSANNE corpus and Analytic Scheme*. Clarendon Press, Oxford, 1995.

[19] M.P. Valverde, M. Civit, and N. Bufí. Guia per a l'anotació sintàctica de cat3lb: un corpus del català amb anotació sintàctica, semàntica i pragmàtica. Technical report, CLiC, 2004. available: http://www.clic.fil.ub.es/personal/civit/publicacions.html.

Analyzing an Italian Treebank with State-of-the-Art Statistical Parsers

Anna Corazza[†], Alberto Lavelli[∗], Giorgio Satta[‡], Roberto Zanoli[∗]

[†]University "Federico II", Napoli, Italy
corazza@na.infn.it

[∗]ITC-irst, Trento, Italy
{lavelli, zanoli}@itc.it

[‡]University of Padova, Italy
satta@dei.unipd.it

1 Introduction

One of the main advantages of data-driven approaches to parsing is portability to new languages. However, such porting operation requires the availability of enough labeled data, i.e. treebanks. A lot of effort has been devoted to the development of English treebanks, resulting in large and reliable treebanks such as the PennTreebank (PTB), and in particular the Wall Street Journal (WSJ) portion of the PennTreebank. However, for other languages the situation is more problematic.

Even in cases where treebanks are available, efforts for applying statistical techniques to parsing produce worse results than for English [6, 2, 5, 13, 9]. This is first of all due to the fact that most parsing techniques have been developed and tuned mainly on English (and on WSJ) and that features that are effective for English may not be optimal for languages with different characteristics. Overall, little effort has been devoted to applying data-driven parsing approaches to individual language other than English. Recently, the application of parsing techniques to languages other than English is becoming a relevant topic, as demonstrated by the papers cited above. Therefore, experimental as well as theoretical analyses of the differences in the behaviour of such techniques, when applied to different languages, are of great interest. Moreover, even when applied to other English treebanks, e.g. portions of the PTB different from WSJ, the results are worse than those obtained on WSJ. In [10] it is reported that Collins' Model 1 trained and tested on the Brown corpus produces results which are 2 points worse than on WSJ; and when the parser is trained on Brown and tested on WSJ the results are almost 6 points worse.

In this paper we report work in progress on the application of state-of-the-art statistical parsing techniques to Italian. Our approach partially differs from previous efforts on other languages because our investigation plans to compare not only results with different parsing methods but also on two different treebanks. Moreover, we try to find information theoretic confirmation of the empirical difference of the experimental results. We hope that exploring the field along these two different dimensions can provide useful hints on the influence of language specificities vs. treebank idiosyncrasies.

2 The Italian Treebanks

As far as Italian is concerned, two treebanks have been recently developed: the Italian Syntactic-Semantic Treebank (ISST, [14]), and the Turin University Treebank[1] (TUT, [3], which is still under active development but not yet available in PTB-like format). ISST is annotated at four levels: morpho-syntactic, two syntactic levels (constituent structure and functional relation), and lexico-semantic. In this work we refer only to the part of the ISST syntactically annotated at the constituent structure level (about 3,000 sentences – 89,941 tokens). The TUT current size is 1,500 sentences (33,868 words) annotated with dependency relations.

3 Experiments on ISST

As a starting point, we considered Model 2 of Collins' parser [7], as implemented by Dan Bikel[2] [1], as its results on the WSJ are at the state-of-the-art. This model applies to lexicalized grammars approaches traditionally considered for probabilistic context-free grammars (PCFGs). Each parse tree is represented as the sequence of decisions corresponding to the head-centered, top-down derivation of the tree. Probabilities for each decision are conditioned on the lexical head.

Adaptation of Collins' parser to Italian included the identification of rules for finding lexical heads in ISST data, the selection of a lower threshold for unknown words (as the amount of available data is much lower), and the use of lemmas instead of word forms (useful because Italian has a richer morphology than English;

[1] `http://www.di.unito.it/~tutreeb/`

[2] `http://www.cis.upenn.edu/~dbikel/#stat-parser`

their use provides a non negligible improvement). At least at the beginning, we did not aim to introduce language-dependent adaptations. For this reason no tree transformation (analogous to the ones introduced by Collins for WSJ) has been applied to ISST.

Assessment of Collins' parser on ISST has been performed by using 10-fold cross-validation. As usual, only sentences with less than 40 words have been considered. For comparison, performance obtained on Section 23 of the WSJ are given. For a fair comparison, in this case only Sections 02 and 03 were used for training, as they are of comparable size with respect to the ISST. Results are presented in Table 1. Even if we expected results on Italian to be significantly worse than on English, the difference in performance is higher than the one obtained with other languages [5, 13, 9].

	WSJ	ISST
P	84.02	68.40
R	83.41	68.58
F_1	83.71	68.49

Table 1: *Results of Collins' parser on WSJ (training: sections 02 & 03; test: section 23) and on ISST.*

In search of an explanation for these disappointing results we studied some parameters extracted from both treebanks, including average number of children per node and depth of the trees, but we did not reach any sound justifications for the different behaviours. At the same time, we analyzed the coverage of the treebank grammars [4] extracted from the training set on the test set. Results are reported in Table 2.

Even if the coverage for Italian is lower than for English, this fact alone does not seem enough to explain such a considerable performance difference. It could also be the case that the Collins' parser is too biased towards PTB-like annotation. [3] This can be verified by repeating the experiment using a different parser.

For this further experiment, we chose the Stanford parser[4] [11, 12]. This lexicalized probabilistic parser implements a factored model, which considers separately the PCFG phrase structure and the lexical dependency. The preferences corresponding to these two different models are then combined by efficient exact inference, using an A* algorithm. In addition to English, the parser can be adapted to work with other languages.

[3] In Chapter 8 of [7], interesting remarks on the influence of annotation style on parser performance are reported, taking into account some of the parsing approaches available at the time.

[4] `http://nlp.stanford.edu/downloads/lex-parser.shtml`

	WSJ		ISST	
Training+Test				
total # rules	4,743		5,734	
# rules - freq=1	2,718	(57.3%)	3,641	(63.4%)
# rules - freq=2	558	(11.8%)	713	(12.4%)
# rules - freq=3	308	(6.5%)	339	(5.9%)
# rules - freq=4	164	(3.5%)	178	(3.1%)
# rules - freq>4	995	(20.9%)	863	(15.2%)
Test				
total # rules	2,741		1,342	
occurrences	44,276		7,663	
# rules not in Training	1,253	(45.7%)	404	(30.1%)
occurrences	1,541	(3.5%)	423	(5.5%)

Table 2: *Coverage of treebank grammar in WSJ (training: sections 02 & 03; test: section 23) and ISST.*

In the spirit of avoiding any language-specific adaptation, for Italian we considered only the basic available annotations, i.e., parent annotation for both nonterminals and tags and horizontal markovization (see [12] for details about the annotations). The head identification rules are the same as in Collins' parser. These preliminary results are presented in Table 3 and confirm that performance on Italian is substantially lower than on English.[5] This result seems to suggest that the differences in performance between the English and Italian treebanks are independent of the adopted parser.

The effects of the different annotations for English and for Italian are different not only from a quantitative point of view, but also in trend. For English the differences between the results are negligible (between 77.12 and 77.73). On the contrary, for Italian different annotations produce significantly different results, even if not in line with those reported in [12]. Parent annotation (PA) and horizontal markovization (with parameter $h = 2$) do not produce any significant difference. Tag parent annotation (tagPA) is the only annotation that significantly improves performance.

While comparing results with the two parsers on English, it is important to take into account that with Collins' parser we could not exclude all adaptations

[5] Note that in these experiments we have used word forms and not lemmas; with Collins' parser lemmas provided slightly better performance, so we expect to improve the performance of Stanford parser as well.

WSJ						
	noPA	PA	noPA tagPA	PA tagPA	noPA tagPA h=2	PA tagPA h=2
P	**77.04**	75.89	76.00	75.76	76.52	75.97
R	77.89	78.69	78.26	**79.70**	78.42	79.56
F_1	77.46	77.27	77.12	77.68	77.46	**77.73**

ISST						
	noPA	PA	noPA tagPA	PA tagPA	noPA tagPA h=2	PA tagPA h=2
P	60.00	59.36	**64.72**	62.15	64.68	62.19
R	59.88	60.78	**67.08**	66.42	66.96	66.31
F_1	59.94	60.06	**65.88**	64.21	65.80	64.18

Table 3: *Results of Stanford parser on WSJ (training: sections 02 & 03; test: section 23) and on ISST.*

(i.e., tree transformations) specific to the English language. On the other hand, none of the language-specific annotations proposed for English was used for the Stanford parser; for this reason performance on English reported in this paper are worse than the ones mentioned in [12].

After the results of the experiments with the second parser, our hypothesis is that the gap in performance between the two languages can be due to two different causes: intrinsic differences between the two languages or differences between the annotation policies adopted in the two treebanks.

To decide which of these two hypotheses is true, we planned to pursue two lines of research: (i) repeating the experiments running both parsers on a different Italian treebank; (ii) exploring information theoretic approaches to the comparison of the difficulties of different parsing tasks. As for the first activity we wanted to repeat the experiments running both parsers on the currently available part of TUT which, even if of smaller size, is based on a completely different linguistic approach, resulting in a different annotation style. In fact, TUT is annotated by following a dependency approach. For this reason, it needs to be converted to a constituent-based annotation, such as the PTB's, using e.g. the algorithm presented in [16]. Unfortunately, this activity has been postponed due to problems in this conversion process. The first results of the activity on information theoretic justification for differences in parsers' behaviour are reported in Section 4.

4 Assessing Parsing Difficulty

In the following we propose some information theoretic measures that may justify the differences of the experimental results on English and Italian.

4.1 Sentential and derivational cross-entropies

In [15] interesting work is presented using an information theoretic approach to the comparison of parsers that originate from different linguistic frameworks. In order to compare performance of such different parsers, the authors suggest to normalize precision and recall using an information theoretic measure associated with the parsers. Such a measure can be seen as an approximation of the cross-entropy of the unknown distribution underlying a corpus C of utterances and the distribution defined by the statistical model μ used by the parser, under the assumption that the parser itself has been induced on some annotation of C, obtained according to some linguistic framework of interest. The approximation of the cross-entropy proposed in [15] is defined by (logarithms in base 2):

$$H_{\mu}(C) = -\frac{1}{|C|} \sum_{u \in C} \log \mathrm{Pr}_{\mu}(u). \tag{1}$$

In the above equation, $\mathrm{Pr}_{\mu}(u)$ indicates the probability of the utterance u under the model μ and is given by the sum of the probabilities of all derivations of u in the model. Note that the above quantity is based on the utterance probability, i.e. the actual parse trees for the utterances in C are disregarded.

Following [15], the model μ corresponds to a “basic generative model” such as the treebank grammar extracted from the training set. Quantity (1) should then be computed on a test set distinct from the training set.

Alternatively to the approach in [15], it is possible to consider the actual parse trees associated with the utterances in a syntactically annotated corpus C. Let T be the collection of annotated parse trees for the utterances in C, obtained according to the linguistic framework of interest. We then define

$$H_{\mu}^{T}(C) = -\frac{1}{|T|} \sum_{t \in T} \log \mathrm{Pr}_{\mu}(t). \tag{2}$$

We now discuss why quantities $H_{\mu}(C)$ and $H_{\mu}^{T}(C)$ are approximations of certain cross-entropies, when the corpora at hand are large enough in size. Both C and T can be seen as sequences of independent and identically distributed random variables. In the case of C, each variable takes values on the set of utterances of the language L underlying C; in the case of T, each variable takes values on the set L^T of parse trees for utterances in L. Assume also that Pr is the unknown, or hidden, distribution for sets L and L^T. Under these conditions, we can apply the asymptotic equipartition property [8]. This guarantees that, at the growing of the corpus size, quantity $H_\mu(C)$ tends to the *sentential cross-entropy*

$$H_\mu = -\sum_{u \in L} \mathrm{Pr}(u) \log \mathrm{Pr}_\mu(u). \tag{3}$$

The same property also guarantees that, at the growing of the corpus size, quantity $H_\mu^T(C)$ tends to the *derivational cross-entropy*

$$H_\mu^T = -\sum_{t \in L^T} \mathrm{Pr}(t) \log \mathrm{Pr}_\mu(t). \tag{4}$$

The two cross-entropies in the left-hand side of Equations (3) and (4) are related, as discussed in what follows. Let u be some utterance in L, and let $T(u)$ denote the set of all parse trees of the utterance u. We define the *utterance cross-entropy* as

$$H_\mu(u) = -\sum_{t \in T(u)} \mathrm{Pr}(t|u) \log \mathrm{Pr}_\mu(t|u) = -\sum_{t \in T(u)} \frac{\mathrm{Pr}(t)}{\mathrm{Pr}(u)} \log \frac{\mathrm{Pr}_\mu(t)}{\mathrm{Pr}_\mu(u)} \tag{5}$$

where we have used the fact that $\mathrm{Pr}(t, u) = \mathrm{Pr}(t)$ in case t belongs to $T(u)$, since t derives u. The utterance cross-entropy is a measure of the uncertainty we experience when choosing a parse tree for u using Pr_μ, due to the ambiguity. A high value of $H_\mu(u)$ not only indicates that there is a big number of parse trees for that utterance, but also that the likelihoods of the parse trees under model μ are very similar, and therefore that it is difficult to decide which one is correct.

We can now relate the sentential and the derivational cross-entropies.

$$\begin{aligned}
&\mathrm{Pr}(u) H_\mu(u) = \\
&\quad = \; -\sum_{t \in T(u)} \mathrm{Pr}(t) \log \mathrm{Pr}_\mu(t) + \sum_{t \in T(u)} \mathrm{Pr}(t) \log \mathrm{Pr}_\mu(u) \\
&\sum_{u \in L} \mathrm{Pr}(u) H_u(u) = \\
&\quad = \; -\sum_{u \in L} \sum_{t \in T(u)} \mathrm{Pr}(t) \log \mathrm{Pr}_\mu(t) + \sum_{u \in L} \sum_{t \in T(u)} \mathrm{Pr}(t) \log \mathrm{Pr}_\mu(u) = \\
&\quad = \; -\sum_{t \in T} \mathrm{Pr}(t) \log \mathrm{Pr}_\mu(t) + \sum_{u \in L} \mathrm{Pr}(u) \log \mathrm{Pr}_\mu(u) = \\
&\quad = \; H_\mu^T - H_\mu
\end{aligned} \tag{6}$$

Equation (6) shows that the difference between the derivational cross-entropy and the sentential cross-entropy depends on the average sentence ambiguity.

4.2 Rule cross-entropy

Another aspect of the difficulty of the parsing task is related to the difficulty of choosing the next rule given the history of the derivation. A measure which aims at evaluating this is introduced in what follows. The treebank can be viewed as a string of random variables $T = r_1 r_2 \ldots r_n$ obtained by the concatenations of all derivations, e.g., the left-most derivations, where each random variable corresponds to a rule. Of course, T is no longer a string of independent and identically distributed random variables, since each rule in a derivation depends on the previous derivation history.

However, under the assumption that our grammar model μ is consistent, i.e. the set of infinite length derivations has zero probability, the derivation process is always reinitialized after some finite number of steps. We thus conclude that T is an ergodic stationary process, and then we can still apply the asymptotic equipartition property [8]. Therefore, the *rule cross-entropy* can be approximated by:

$$H_\mu^R(C) = -\frac{1}{n} \log \Pr_\mu(T) = -\frac{1}{n} \sum_{i=1}^{n} \log \Pr_\mu(r_i). \tag{7}$$

4.3 Data sparseness and smoothing

However, in real cases, in (7) the probability of some of the rules could be zero, as they are not present in the training set. In such cases, some smoothing technique needs to be adopted to estimate the probabilities also of the rules not seen in the training set. For simplicity, let us assume that all such rules are assigned the same probability p_e under model μ, and let us call n_e their total number, while n_r is the total number of the test set rules also present in the training set, so that $n_r + n_e$ gives the overall number of rules in the test set. Equation (7) becomes:

$$\begin{aligned} H_\mu^R(C) &= -\frac{1}{n_r + n_e} \log \Pr_\mu(T) = \\ &= -\frac{1}{n_r + n_e} \sum_{i=1, \Pr(r_i) \neq 0}^{n_r + n_e} \log \Pr_\mu(r_i) - \frac{n_e}{n_r + n_e} \log p_e = \\ &= \frac{n_r}{n_r + n_e} H_\mu(r) - \frac{n_e}{n_r + n_e} \log p_e. \end{aligned} \tag{8}$$

Therefore, we can assess the difficulty of the task by considering on one hand the cross-entropy computed on the string of only the rules with probability greater than zero (first term in (8)) and on the other hand the percentage $\frac{n_e}{n_r + n_e}$ of the new rules with respect to the size of the test set. In this way, our results do not depend on the value of p_e, which is a choice of the parsing system, and not a characteristic of the task.

There are also other approaches to evaluate data sparseness, such as counting all events occurring only once (the Good-Turing approach). However, in our case we are using a split into training and test sets also for the parsers assessment, and then it seems more consistent to use the same split to assess data sparseness too.

In order to use Equation (8) to study coverage and data sparseness, it is necessary to avoid any smoothing. Moreover, when data are very sparse, such as in the experiments on ISST, the risk is that the influence of the smoothing probabilities becomes predominant in determining the final results. Therefore, in the Stanford parser we considered the results of the (unsmoothed) PCFG instead of the results of the combined model, which also includes the (smoothed) dependency model.

Furthermore, for a better evaluation, in addition to the results on the whole test set, we also report the evaluation restricted to the sentences whose derivation in the test set consists only of rules also appearing in the training set: we call this part "covered".

4.4 Preliminary empirical results

In Tables 4 and 5 some preliminary results on cross-entropy and coverage and on parsing adopting a treebank grammar are reported. In both cases a subset of sections 02 and 03 of WSJ (i.e., the training set used in the experiments reported in Section 3) is chosen such that the coverage on the test set of the rules extracted from the training set is very similar to the coverage on the test set obtained with ISST. Such subset corresponds to one fourth of the overall size of sections 02 and 03.

	WSJ all	WSJ covered	ISST all	ISST covered
Rule entropy	4.53	4.42	3.53	3.76
Derivational entropy	230.17	75.46	352.64	69.56
# rules	47,119	14,908	7,757	1,441
# uncovered rules	2,731 (5.80%)	0	449 (5.79%)	0
# sents	2,409	874	293	78
% uncovered sents	63.72%		73.38%	

Table 4: *Cross-entropy and coverage.*

First of all, it is interesting to note how the same coverage on rules (about 94.2%) results in the Italian corpus in a sensibly lower coverage on sentences

(26.62% vs. 36.28%). This discrepancy suggests that missing rules are less concentrated in the same sentences, and that, in general, they tend to be less correlated the one with the other. This would not be contradicted by a lower entropy, as the entropy does not make any hypothesis on the correlation between rules, but only on the likelihood of the correct derivation. This could be a first aspect making the ISST task more difficult than the WSJ one. In fact, the choice of the rules to introduce at each step is easier if they are highly correlated with the ones already introduced.

Another aspect that we did not have the time to check is the degree of ambiguity of the grammar. As discussed in Section 4.1, this could be done by comparing the derivational entropy with the sentential one, computed on the training set by using the Inside algorithm for the treebank grammar.

The performance of the treebank grammar without any smoothing strategy as explained above is reported in Table 5 and shows a difference between WSJ and ISST which is similar to the one presented in Tables 1 and 3.

Comparing the performance on the whole test set and on the covered part of the test set, it can be noted that in the former case they are better both on WSJ and on ISST. However, the improvement on the Italian corpus is definitely lower than in the English case. It is our opinion that this is another evidence of the lower correlation among rules in the Italian treebank grammar. In fact, even in the covered sentences, the task seems to be more difficult.

	WSJ all	WSJ covered	ISST all	ISST covered
Precision	65.37	74.73	60.52	62.73
Recall	63.15	69.54	60.46	60.73
F_1	64.24	72.05	60.49	61.72

Table 5: *Treebank grammar and precision/recall results.*

5 Future work

After the activities presented in this paper, two hypotheses to explain the gap in performance between English and Italian are still available: intrinsic differences between the two languages or differences between the annotation policies adopted in the two treebanks. To decide which of the hypotheses is true, we plan to repeat the experiments in Section 3 running both parsers on the currently available part of TUT. We will pursue this activity as soon as the converted treebank will be available. Furthermore, we would also like to continue the investigation of information theoretic measures of the difference between treebanks.

Another more practical line of activity includes an error analysis to identify the classes of errors done by the two algorithms, so that strategies to cope with them can be designed. For Collins' parsers this would imply the introduction of (reversible) transformations of the trees. For Stanford parser, this will produce new annotations of the nodes of the parse trees. Some indications on the kind of analysis which can be applied are discussed in [13] for Chinese. For the sake of distinguishing language-specific aspects from idiosyncrasies of the particular treebanks, measures on the two Italian treebanks are to be compared.

Acknowledgments

We thank Dan Bikel and Christopher Manning for making available their own software tools and for kindly answering our questions about their usage. We also thank Roger Levy for providing useful feedback and Khalil Sima'an for some clarification about his work.

References

[1] Daniel M. Bikel. Intricacies of Collins' parsing model. *Computational Linguistics*, forthcoming.

[2] Daniel M. Bikel and David Chiang. Two statistical parsing models applied to the Chinese Treebank. In *Proceedings of the Second Chinese Language Processing Workshop*, Hong Kong, 2000.

[3] Cristina Bosco, Vincenzo Lombardo, D. Vassallo, and Leonardo Lesmo. Building a treebank for Italian: a data-driven annotation schema. In *Proceedings of 2nd International Conference on Language Resources and Evaluation (LREC-2000)*, Athens, Greece, 2000.

[4] Eugene Charniak. Tree-bank grammars. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, Portland. Oregon, 1996.

[5] David Chiang and Daniel M. Bikel. Recovering latent information in treebanks. In *Proceedings of the 19th International Conference on Computational Linguistics*, Taipei, Taiwan, 2002.

[6] M. Collins, J. Hajic, L. Ramshaw, and C. Tillmann. A statistical parser for Czech. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, College Park, Maryland, 1999.

[7] Michael Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999.

[8] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley & Sons, 1991.

[9] Amit Dubey and Frank Keller. Probabilistic parsing for German using sister-head dependencies. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 96–103, Sapporo, Japan, 2003.

[10] Daniel Gildea. Corpus variation and parser performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP 2001)*, pages 167–202, Pittsburgh, PA, 2001.

[11] Dan Klein and Christopher D. Manning. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, 2002.

[12] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, 2003.

[13] Roger Levy and Christopher D. Manning. Is it harder to parse Chinese, or the Chinese treebank? In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, 2003.

[14] S. Montemagni, F. Barsotti, M. Battista, N. Calzolari, A. Lenci O. Corazzari, A. Zampolli, F. Fanciulli, M. Massetani, R. Basili R. Raffaelli, M.T. Pazienza, D. Saracino, F. Zanzotto, F. Pianesi N. Mana, and R. Delmonte. Building the Italian Syntactic-Semantic Treebank. In Anne Abeillé, editor, *Building and Using syntactically annotated corpora*, pages 189–210. Kluwer, Dordrecht, 2003.

[15] Gabriele Musillo and Khalil Sima'an. Towards comparing parsers from different linguistic frameworks. An information theoretic approach. In *Proceedings of the LREC-2002 workshop Beyond PARSEVAL. Towards Improved Evaluation Measures for Parsing Systems*, Las Palmas, Spain, 2002.

[16] Fei Xia and Martha Palmer. Converting dependency structures to phrase structures. In *Proceedings of the Human Language Technology Conference (HLT-2001)*, San Diego, California, 2001.

Recent Developments in Linguistic Annotations of the TüBa-D/Z Treebank

Erhard Hinrichs, Sandra Kübler, Karin Naumann,
Heike Telljohann, Julia Trushkina
Universität Tübingen
Seminar für Sprachwissenschaft
Wilhelmstr. 19
72074 Tübingen, Germany
{eh, kuebler, knaumann}@sfs.uni-tuebingen.de,
{hschulz, jul}@sfs.uni-tuebingen.de

1 Introduction

The purpose of this paper is to describe recent developments in the morphological, syntactic, and semantic annotation of the TüBa-D/Z treebank of German.

The TüBa-D/Z annotation scheme is derived from the Verbmobil treebank of spoken German [4, 10], but has been extended along various dimensions to accommodate the characteristics of written texts. TüBa-D/Z uses as its data source the 'die tageszeitung' (taz) newspaper corpus.

The Verbmobil treebank annotation scheme distinguishes four levels of syntactic constituency: 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 [3, 6]. The TüBa-D/Z annotation relies on a context-free backbone (i.e. proper trees without crossing branches) of phrase structure combined with edge labels that specify the grammatical function of the phrase in question. The syntactic annotation scheme of the TüBa-D/Z is described in more detail in [12, 11].

TüBa-D/Z currently comprises approximately 15 000 sentences, with approximately 7 000 sentences being in the correction phase. The latter will be released along with an updated version of the existing treebank before the end of this year. The treebank is available in an XML format, in the NEGRA export format [1] and in the Penn treebank bracketing format. The XML format contains all types of information as described above, the NEGRA export format contains all sentence-internal information while the Penn treebank format includes only those layers of information that can be expressed as pure tree structures.

Over the course of the last year, more fine grained linguistic annotations have been added along the following dimensions: 1. the basic Stuttgart-Tübingen tagset, STTS, [9] labels have been enriched by relevant features of inflectional morphology, 2. named entity information has been encoded as part of the syntactic annotation, and 3. a set of anaphoric and coreference relations has been added to link referentially dependent noun phrases. In the following sections, we will describe each of these innovations in turn and will demonstrate how the additional annotations can be incorporated into one comprehensive annotation scheme.

2 Morphological Annotation

The STTS [9] provides the widely accepted inventory of part of speech (POS) categories for German. Its basic tagset distinguishes 54 POS labels but does not provide information about inflectional morphology, which is a necessary prerequisite for many natural language applications, such as, for example, the recognition of grammatical functions in German [13]. In order to incorporate such morphological information, the treebank annotation scheme has been enriched by morphological features such as *case, number, person, gender, tense,* and *mood*.

For each lexical token which exhibits inflectional morphology, a relevant combination of feature-value pairs has been assigned. Thus, for example, nouns have received information on case, number, and gender, finite verbs are annotated with person, number, mood, and tense information. A complete list of POS tags which have been assigned morphological features as well as feature combinations associated with each part of speech are provided in Table 1.

Lexical tokens	**Feature Combination**
nouns, adjectives, determiners, non-personal pronouns, prepositions with incorporated articles	case, number, gender
prepositions, postpositions	case
personal pronouns	case, number, gender, person
finite verbs	person, number, mood, tense
imperative verbs	person, number
truncated words	number, gender

Table 1: Feature combinations for lexical tokens in TüBa-D/Z.

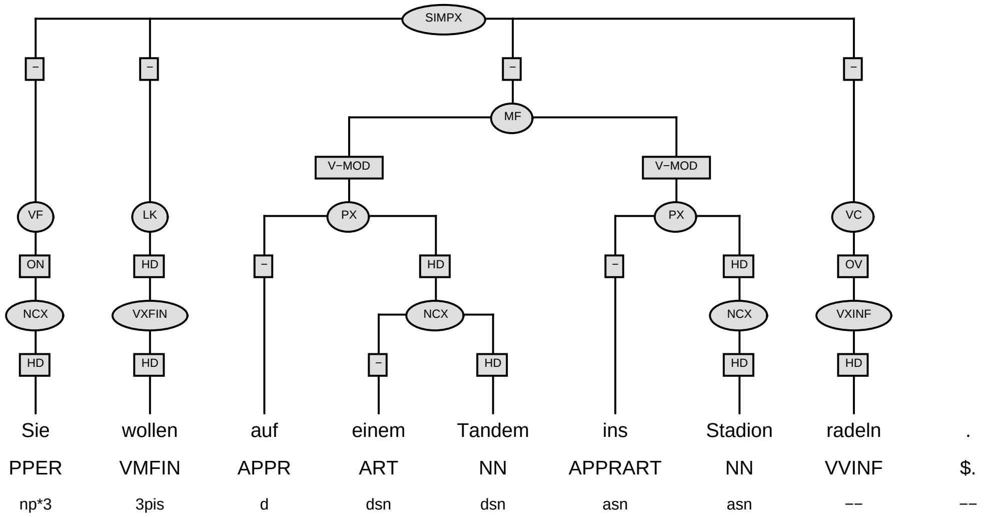


Figure 1: A morphologically annotated tree.

The tree in Figure 1 illustrates the annotation of the morphological information in the treebank for the sentence in example (1). Values of morphological features are presented in the treebank explicitly on the level below the level of lexical tokens. Features that correspond to the values can be uniquely identified by a position of a value in a cluster, given the POS tag. Thus, a cluster `3pis` assigned to the verb *wollen* in Figure 1 stands for "**3**rd person, **p**lural number, **i**ndicative mood, pre**s**ent tense". The order of features in the morphological cluster corresponds to the order in Table 1. Possible values for each feature are presented in Table 2. Apart from specific features such as *masculine* or *singular*, values for case, gender, and number features include an underspecified value. The underspecified value is used for the annotation of tokens if an appropriate concrete value cannot be recovered for a morphological feature. Typical examples of the use of an underspecified value are plural pronouns, such as *sie* (*they*) in Figure 1 or first person pronouns, such as *ich* (*I*). In both cases, gender cannot be determined.

(1) Sie wollen auf einem Tandem ins Stadion radeln.
They want to on a tandem into the stadium bike.
' They want to bike into the stadium on a tandem.'

In total, 433 distinct morphological value clusters can be generated. Combined with POS information, they result in a tagset of 1 317 tags. The number of actual tags which occur in the treebank amounts to 555 tags.

Features in TüBa-D/Z	**Values**
case	n (nominative), g (genitive), d (dative), a (accusative), * (underspecified)
gender	m (masculine), f (feminine), n (neutral), * (underspecified)
number	s (singular), p (plural), * (underspecified)
mood	i (indicative), k (subjunctive)
person	1 (first), 2 (second), 3 (third)
tense	s (present), t (past)

Table 2: Set of feature values in TüBa-D/Z.

Currently, approximately 13 000 trees have been enriched with morphological information. Annotation was performed semi-automatically by using the rule-based morphological disambiguator of Hinrichs and Trushkina [5] as a pre-filtering module that limits the number of candidate analyses for each lexical token to those that are contextually valid. This rule-based disambiguation greatly reduces the number of analyses from an overall ambiguity rate of 5.8 analyses to 1.91 analyses per token and by providing full disambiguation for 70% of all tokens. As a result, the human annotators have to consider a much smaller set of analyses, which significantly speeds up the annotation process.

The morphologically annotated treebank data have in turn been used for the training of hybrid models of morphological disambiguation that combine rule-based and statistical disambiguation [13].

3 Named Entities

For a variety of NLP applications, the robust annotation of named entities is an important prerequisite. To facilitate the use of the TüBa-D/Z data for such tasks, the level of named entity annotation has been added to the annotation scheme. This additional layer of annotation is conservative and monotonic in the following sense: It respects all syntactic boundaries that have been imposed on the elements of named entity expressions by existing layers of syntactic annotation. Named entity annotation thus amounts to mere insertion of an intermediate level of representation. At the same time, named entity annotation is fully compliant with the STTS labeling assigned to the elements of named entity expressions. These two constraints on named entity annotation ensure that it can be easily removed if such information is irrelevant for the task to which the treebank is to be applied.

Named entities are annotated on the morpho-syntactic level via the STTS tags and/or on the syntactic level. The STTS tagset uses the label NE for proper names and NN for common nouns. The classification of NE in the STTS guidelines comprises specific categories (e.g. first name, last name, names of companies, geographical names). By contrast, categories like names of products or compounds which consist of NE + NN are POS-tagged as NN. Moreover, complex German names have to be POS-tagged according to their distribution.

Named entities either occur as single names consisting of one lexical element or as complex names consisting of phrases or sentences. Complex names are annotated on the syntactic level by the label EN-ADD or the secondary edge EN, single elements are either marked on the morpho-syntactic level as NE or they receive the label EN-ADD.

Figure 2 gives an example of the annotation of named entities for the sentence in example (2). Here, the two person names are marked as names in the POS tags NE and as complex names by the label EN-ADD, the movie title is marked by the label EN-ADD. The geographical name within the movie title is POS-tagged as NE.

(2) Seit “Schlaflos in Seattle” gelten Tom Hanks und Meg Ryan als
Since “Sleepless in Seattle” pass Tom Hanks and Meg Ryan for
Dream-Team des Biedersinns.
dream team of petty bourgeois mentality.

’Since “Sleepless in Seattle” Tom Hanks and Meg Ryan are said to be the dream team of petty bourgeois mentality.’

In the treebank, the following classes of named entities exist:

1. Names consisting of one lexical element: They are POS-tagged as NE if they belong to one of the categories of proper names defined in the STTS guidelines. Otherwise, they are POS-tagged according to their distribution and assigned the additional node label EN-ADD. For example, nouns which are names of products (“Opel” NN) or compounds which consist of NE + NN like names of streets or places ("Sögestraße" NN), institutions ("Zeit-Stiftung" NN), or events ("Golfkrieg" NN).

2. Complex names consisting of more than one lexical element, each of them POS-tagged as NE: This class comprises complex names of persons (e.g. "Hans Taake") and foreign language material which can be recognized as a proper name (e.g. "New York", "Karel van Miert", "Tour de France"). All of them are assigned the additional node label EN-ADD.

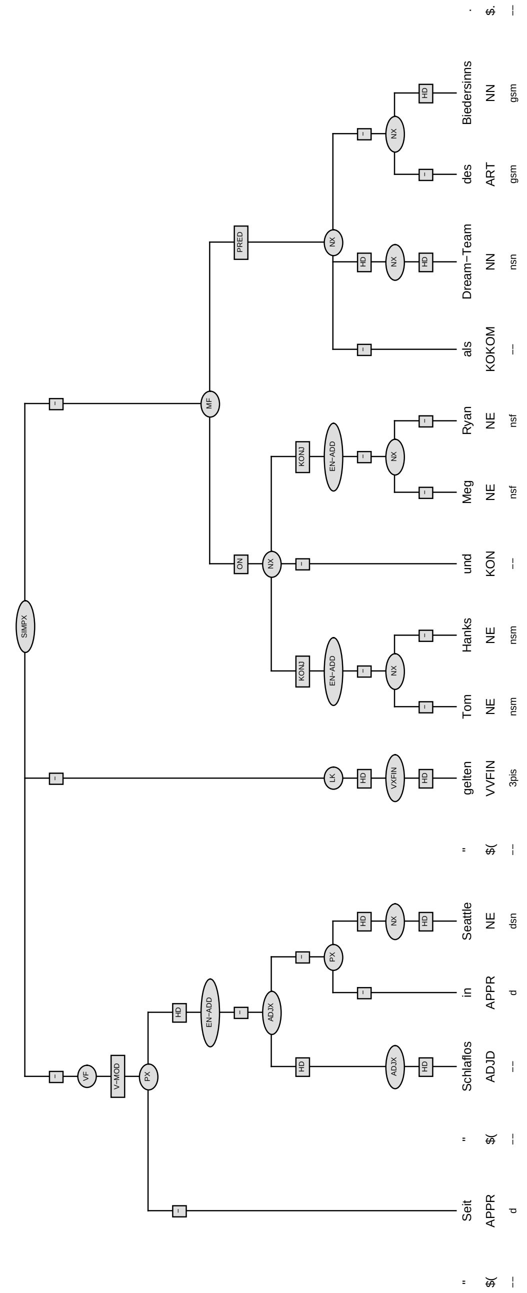


Figure 2: A tree containing named entities.

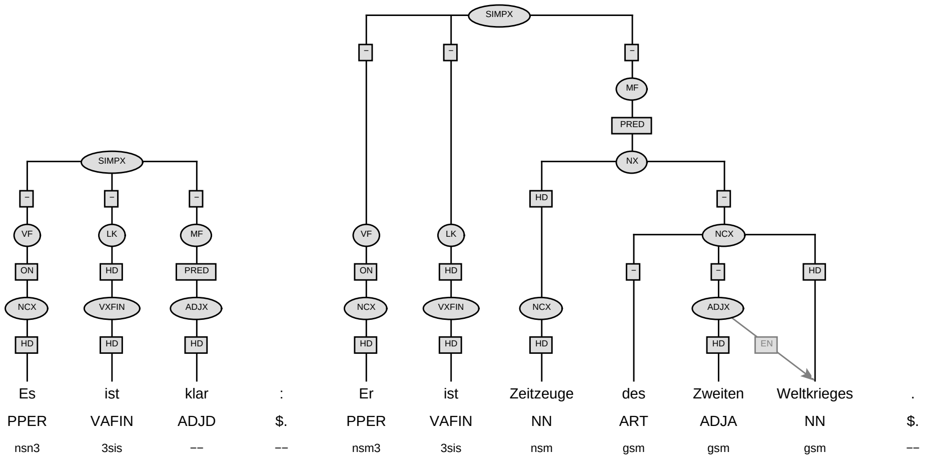


Figure 3: A tree containing a phrase internal named entity.

3. Complex names which are POS-tagged according to their distribution: titles, institutions, events, etc. (e.g. "Schlaflos in Seattle", "Zweiter Weltkrieg"). They are either assigned the additional node label EN-ADD or the secondary edge label EN.

The labels EN-ADD and EN are general markers of named entities, which have no syntactic function. Thus, they do not effect the syntactic structure if they are deleted. The internal structure of named entities is always governed by the general annotation rules, which allows recursive structure (named entities within named entities).

EN-ADD is inserted between two nodes to indicate that the node below represents a named entity. It is either directly attached to a phrase or a field. If this named entity has a pre- or postmodifier, its mother node is NX which represents the nominal status of the named entity.

The secondary edge label EN is used when the insertion of EN-ADD would cause a change of the syntactic structure. It gives information about the relation between two parts of a named entity within a complex phrase. The named entity is premodified, for instance, by an article and/or an attributive adjective which do not belong to the named entity itself (e.g. "vor den zweiten [Deutschen Existenzgründertagen]"), and may also be postmodified by an element which is part of the named entity (e.g. "das [Bundesinstitut für Arzneimittel]"). EN always points from the dependent part to the head noun of te named entity.

Figure 3 gives an example of a phrase internal named entity ("Zweiten Weltkrieges") in the sentences in example (3). The article ("des") is no part of the named entity itself.

(3) Es ist klar: Er ist Zeitzeuge des Zweiten Weltkrieges.
It is clear: He is contemporary witness of the Second World War.
'It is clear: He is a contemporary witness of World War II.'

Preliminary experiments have shown that the inclusion of named entity annotation improves parsing accuracy of statistical parsers trained on the TüBa-D/Z data.

4 Anaphoric and Coreference Relations

Due to its fine grained syntactic annotation, the TüBa-D/Z data are ideally suited as a basis for the identification of markables, i.e. the set of potential anaphoric and other contextually dependent expressions referring to a nominal or pronominal antecedent. The annotation of anaphoric and coreference relations is thus a natural extension to the existing annotation scheme. In this context, the potential markables are definite NPs, personal pronouns, relative, reflexive, and reciprocal pronouns, demonstrative, indefinite and possessive pronouns as well as possessive adjectives. Compared to other annotation efforts in this area where markables have to be chosen manually, the actual manual annotation in the case of TüBa-D/Z can be restricted to the selection of the appropriate linking relations between referentially dependent expressions and their nominal antecedents. The inventory of such relations is inspired by the annotation scheme first developed in the MATE project [2] and uses the following subset of relations: *coreferential*, *anaphoric*, *cataphoric*, *bound*, *part-of*, *instance*, and *expletive*. Following van Deemter and Kibble [14], we define a coreference relation to hold between two NPs just in case they refer to the same extra-linguistic referent in the real world. In the following example, a *coreference relation* exists between the noun phrases [1] and [2], and an *anaphoric relation* between the noun phrase [2] and the personal pronoun [3].

(4) [1 Der neue Vorsitzende der Gewerkschaft Erziehung und Wissenschaft]
The new chairman of the union Education and Science
heißt [2 Ulli Thöne]. [3 Er] wurde gestern mit 217 von 355
is called Ulli Thöne. He was yesterday with 217 out of 355
Stimmen gewählt.
votes elected.
'The new chairman of the union of educators and scholars is called Ulli Thöne. He was elected yesterday with 217 of 355 votes.'

Cataphoric relations hold between a preceding pronoun and its antecedent within the same sentence, even if this antecedent has already been mentioned within the preceding text. An example for a cataphoric relation is shown in (5).

(5) Vier Wochen sind [sie] nun schon in Berlin, [die 220 Albaner aus
Four weeks are they now already in Berlin, the 220 Albanians from
dem Kosovo].
the Kosovo.
’They have already been in Berlin for four weeks, the 200 Albanians from Kosovo.’

The relation *bound* holds between anaphoric expressions and quantified noun phrases as their antecedents (see example (6)).

(6) [Niemandem] fällt es schwer, das Bild vor [sich] zu sehen.
To nobody is it difficult, the picture in front of himself to see.
’Nobody has trouble imagining the picture.’

The *part-of relation* holds between coordinate NPs/plural pronouns and pronouns/definite NPs referring to one member of the plural expression.

(7) [Ein paar andere Fehler] hat er aber schon vorher gemacht. [Den
A few other errors has he however already before made. The
ersten] Ende des vergangenen Jahres.
first end of the previous year.
’He had however already made a few other mistakes. The first one at the end of the previous year.’

An *instance relation* exists between a preceding/following pronoun and its NP antecedent when the pronoun refers to a particular instantiation of the class identified by the NP.

(8) Die konservativen Kräfte warten ja nur darauf, ihm [Sätze] um
The conservative powers wait just only for that, him sentences around
die Ohren zu hauen wie [jenen von den 16 Mittelstrecklern],
the ears to hit like the one about the 16 middle-distance runners,
denen er in vier Wochen die Viererkette beibringe.
to whom he in four weeks the double full-back formation teaches.
’The conservative powers are just waiting to bombard him with sentences like the one about the 16 middle-distance runners who he is teaching the double full-back formation in four weeks.’

The impersonal third person sg. pronoun ES (IT) is marked as *expletive* only if it has no proper antecedent, which is the case for presentational ES in example (9), impersonal passive as in example (10) or ES as subject for verbs without an agent as in example (11).

(9) [1 Es] zeichnet sich die konkrete Möglichkeit ab.
It emerges the concrete possibility *verb part*.
'The concrete possibility emerges.'

(10) [Es] wird bis zum Morgen getanzt.
There is until the morning danced.
'People are dancing until morning.'

(11) [Es] steht schlecht um ihn.
It stands bad for him.
'He is in a bad way.'

The annotation of such relations is performed manually with the annotation tool MMAX [8]. Its graphical user interface allows for easy selection of the relevant markables and the accompanying relation between the contextually dependent expression and its antecedent. In a first step, the relevant markables receive an attribute value: coreferential, anaphoric, cataphoric, bound, part-of, instance, or expletive. Second, the relation between a contextually dependent expression and its antecedent is established, except for the attribute "expletive", which is not related to an antecedent in the text. MMAX distinguishes between two kinds of relations: a set relation is defined as a transitive undirected relation. A pointer relation, in contrast, is intransitive and directed. Expressions marked by the attribute "coreferential", "anaphoric", "cataphoric" or "bound" share a set relation with their antecedent. Expressions marked by the attribute "part-of" and "instance" share a pointer relation with their antecedent.

The resulting annotation is converted into the Annotate export format [1] and the XML format, in which the treebank is available[1].

[1] For licensing information please visit the webpage `http://www.sfs.uni-tuebingen.de/en_tuebadz.shtml`.

5 Conclusion

This paper presents three recent additions to the previous layers of annotation in the TüBa-D/Z, which significantly enhance the usability of the treebank for NLP applications. While each addition is independently motivated, it is important to note that the new information could be incorporated into the existing annotation scheme attesting to the flexibility and open architecture of the annotation scheme.

References

[1] Thorsten Brants. *The NeGra Export Format for Annotated Corpora*. Universität des Saarlandes, Computational Linguistics, Saarbrücken, Germany, 1997.

[2] Sarah Davies, Massimo Poesio, Florence Bruneseaux, and Laurent Romary. *Annotating Coreference in Dialogues: Proposal for a Scheme for MATE*. MATE, 1998.

[3] Erich Drach. *Grundgedanken der Deutschen Satzlehre*. Diesterweg, Frankfurt/M., 1937.

[4] Erhard W. Hinrichs, Julia Bartels, Yasuhiro Kawata, Valia Kordoni, and Heike Telljohann. The Tübingen treebanks for spoken German, English, and Japanese. In Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*, pages 550–574. Springer, Berlin, 2000.

[5] Erhard W. Hinrichs and Julia Trushkina. Getting a grip an morphological disambiguation. In *Proceedings of KONVENS 2002, 6. Konferenz zur Verarbeitung natürlicher Sprache*, pages 59–66, Saarbrücken, Germany, 2002.

[6] Tilman Höhle. Der Begriff "Mittelfeld", Anmerkungen über die Theorie der topologischen Felder. In *Akten des Siebten Internationalen Germanistenkongresses 1985*, pages 329–340, Göttingen, Germany, 1986.

[7] Claudia Kunze and Andreas Wagner. Integrating GermaNet into EuroWordNet, a multilingual lexical-semantic database. *Sprache und Datenverarbeitung. International Journal for Language Data Processing*, 23(2):5–19, 1999.

[8] Christoph Müller and Michael Strube. Multi-level annotation in MMAX. In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue*, Sapporo, Japan, 2003.

[9] Anne Schiller, Simone Teufel, and Christine Thielen. Guidelines für das Tagging deutscher Textkorpora mit STTS. Technical report, Universität Stuttgart and Universität Tübingen, September 1995.

[10] Rosmary Stegmann, Heike Telljohann, and Erhard W. Hinrichs. Stylebook for the German Treebank in VERBMOBIL. Technical Report 239, Verbmobil, 2000.

[11] Heike Telljohann, Erhard Hinrichs, and Sandra Kübler. 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)*, Lisbon, Portugal, 2004.

[12] Heike Telljohann, Erhard W. Hinrichs, and Sandra Kübler. *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)*. Seminar für Sprachwissenschaft, Universität Tübingen, Tübingen, Germany, 2003.

[13] Julia Trushkina and Erhard W. Hinrichs. A hybrid model for morphosyntactic annotation of German with a large tagset. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 238–246, Barcelona, Spain, 2004.

[14] Kees van Deemter and Rodger Kibble. On coreferring: Coreference in MUC and related annotation schemes. *Computational Linguistics*, 26(2):629–637, 2000.

Annotating and Querying a Treebank of Suboptimal Structures

Stephan Kepser, Ilona Steiner, and Wolfgang Sternefeld

SFB 441, University of Tübingen, Germany

{kepser,steiner}@sfs.uni-tuebingen.de,
wolfgang.sternefeld@uni-tuebingen.de

1 Introduction

Existing treebanks of written language, as e.g., TIGER [2], TüBa-D/Z [11], Penn Treebank [1] etc., usually consist of sentences that can be considered as grammatically well-formed. The SINBAD treebank we present here covers a completely new domain, namely suboptimal syntactic structures, i.e., sentences which are neither fully grammatical nor completely ungrammatical, but merely suboptimal.[1] The treebank consists of a collection of German sentences that are rated suboptimal or ungrammatical in the literature, as well as of sentences drawn from our own experimental work on graded grammaticality judgments. In the literature, these structures are usually compared with grammatical structures which express the same meaning, and for ease of comparison these were sometimes included in the treebank as well. With this data collection we provide access to negative evidence which does not occur in ordinary corpora of written or spoken language.

It is characteristic for suboptimal structures that these data are judged incoherently varying between different speakers and in different contexts. It is therefore important to provide a systematic collection of these judgments in order to allow researchers better access to past judgements on the phenomena they are interested in and thus contribute towards greater consistency, even in tricky cases. Since most work in syntactic theory is based on suboptimal or ungrammatical structures, the treebank aims at providing linguists with a data basis for their research. This requires a rich syntactic annotation with linguistically relevant concepts. The linguistic framework of the annotation is that of generative grammar in the sense that the trees are strictly binary branching and contain traces and empty categories. The

[1] Note that the term *suboptimal* is referred to grammaticality and not to mere processing considerations. Garden-path sentences, for example, are excluded from this domain.

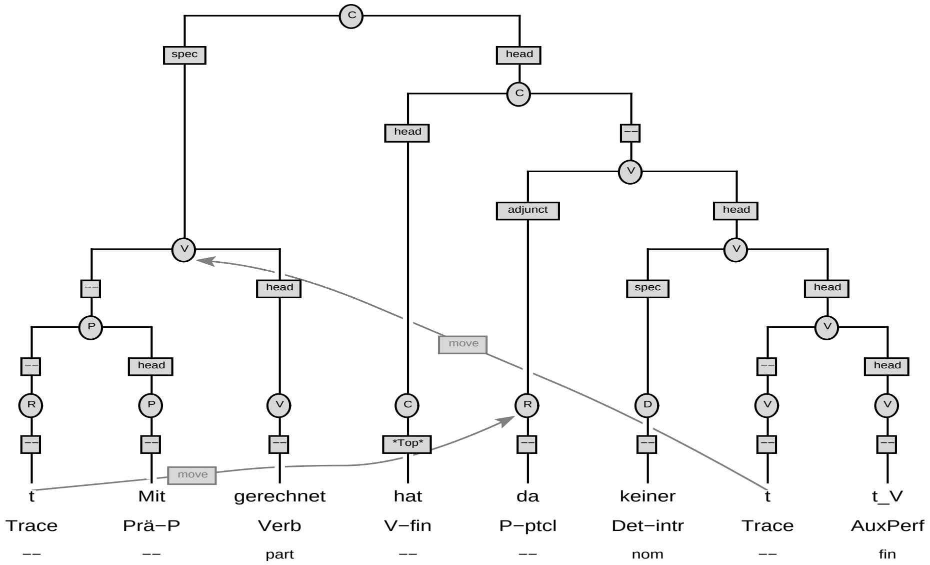


Figure 1: Remnant movement in German [5]

annotation scheme is inspired by the feature grammar which Sternefeld developed for German [9]. To our knowledge this is the first treebank following principles of generative grammar.

The new domain of suboptimal structures and the particular linguistic framework chosen raise additional research questions with respect to annotation schemes as well as querying these structures. In Section 2 we present the design principles chosen for our treebank, in Section 3 we focus on the how these structures can be queried effectively.

2 Syntactic Annotation of Suboptimal Structures

The treebank of suboptimal structures is work in progress and comprises ca. 1060 sentences at the moment. The intended size of the treebank is about 3000 sentences with the target being more a qualitative than a quantitative one. It has been annotated manually by one student assistant using the *Annotate* tool [6]. Fig. 1 shows a sample entry of our treebank: *Mit gerechnet hat da keiner* (lit. “With reckoned has it nobody”, meaning “Nobody expected that”). This sentence is rated suboptimal (‘?’) in the literature (taken from [5]).

The approach was to build up a modest basis of data, and then develop the analytical framework on the basis of this partial data set. The major part of this task has been completed, larger quantities of examples can be added, without the danger that they need to be recoded in an architectural redesign. To ensure accuracy and consistency of the annotations, the treebank has been checked in several proofreading sessions. In addition, the query tool fsq (see next section) has been used to eliminate errors in the annotation and to ensure consistency of the data.

2.1 Design Principles

Using a generative framework for the annotation is challenging, because it may well be that “a sentence has as many structures as there are theories” (Haider, [3]). Nonetheless, we tried to find a compromise between (a) naive expectations of a linguistically trained user (b) run of the mill assumptions in generative grammar (c) simplicity of structure, and (d) enhanced parsability. In accordance with these aims we attempted to minimize the number of different syntactic categories, to minimize occurrences of empty categories, to minimize inexplicitness of structure by strictly adhering to binary branching, and to minimize the role of X-bar theory by following minimalistic assumptions. As a result of these requirements, we maximized the analytical importance of structure.

2.2 Annotation Scheme

The treebank is annotated with Part-of-Speech tags (PoS tags), morphological information, syntactic categories (node labels), grammatical functions (edge labels) and additional contextual features (lexical edge labels). In addition, secondary edges are used for the annotation of movement and co-reference. The details of the annotation scheme are described in the SINBAD stylebook [10].

2.2.1 Node Labels

Node labels specify the major syntactic categories of constituents. Due to the richness of syntactic structure it is possible to reduce the number of node labels to a minimum of seven different syntactic categories:

A the category of adjectives and adverbials
C the category of complementizers and the position of the finite verb in main clauses
D the category of determiners, including
intransitive determiners like pronouns and proper names
N the category of common nouns including proper nouns
P the category of adpositions, i.e., pre- and postpositions
V the category of verbs
R a default category for anything that does not fit into the above categories

Categories like AP, CP, DP etc., which are primitives of traditional X-bar theory, are dispensed within our annotation, but can be defined with the help of edge labels, as will be shown further below. Note also that there is no Infl category in our annotation. Following the theory developed by Sternefeld [9], clauses are CPs, and the complement of C is a VP.

2.2.2 Part-of-Speech Tags and Morphological Labels

PoS tags subcategorize the seven node labels according to their morpho-syntactic lexical properties as illustrated in Table 1 below. We opted to develop our own PoS tag set for the following reasons. A considerable amount of information encoded in existing tag sets, such as the STTS [7], is already encoded in our annotation in a different way using edge labels, tree structure or morphological informations. We wanted to avoid the redundancy of restating that. Furthermore existing PoS tag sets do not adequately capture the linguistic intentions of the annotation; they thrive to be theory-neutral while our tag set is derived from the linguistic framework we use.

Subcategories of A		
Ad	adverb, predicative adjective	er fährt/ist *schnell*
A-infl	inflected adjective	ein *schneller* Fahrer
Adv	adverbial	*heute*, *schon*, *bald*
W-Pron	wh-pronoun	*wie* geht es dir?
Subcategories of C		
V-fin	the finite verb in C	Fritz *schläft* ein
C-fin	complementizer with finite clause	*dass* er kommt
C-zu	complementizer with infinite clause	*um* zu arbeiten, *anstatt*
Subcategories of D		
W-Pron	wh-pronoun	*wer*, *wessen*, *was*, *welcher*
Rel-Pron	relative pronoun	*dem*, *dessen*
Poss-Pron	possessive pronoun	*mein*, *dein*, *unser*
Refl-Pron	reflexive pronoun	*sich*
Rec-Pron	reciprocal pronoun	*einander*
Pers-Pron	personal pronoun	*ich*, *du*, *er*,. . . , *mich*, *dich*. . . , *meiner*, *mir* etc.
Prop-N	proper name	*Fritz*, *Anna*, *Fritzens* Mut, *Annas* Kleid
Det	transitive determiner	*d-er*, *jed-er*, *ein*, *kein*

Det-intr	intransitive determiner	das ist *meins*, da ist *keiner*, *den* kenne ich, *PRO*
Subcategories of N		
CN	common noun	*Haus*, *Wand*, *Eis*, *Gold*
PN	proper noun	der *Hans*, die *Schweiz*
Subcategories of P		
Prae-P	preposition	*in*, *an*, *auf*, *mit*, *ohne*, *von*
Post-P	postposition	*wegen*, *halber*
P-Adv	pronominal adverb	*damit*, *davon*
P+Det	preposition + determiner	*im*, *am*, *ins*
Subcategories of V		
AuxMod	modal auxiliary	*wollen*, *können*, *müssen*, *dürfen*, *sollen*
AuxPerf	temporal auxiliary	*haben*, *sein*
AuxFut	temporal auxiliary	*werden*
AuxPass	passive auxiliary	*werden*, *kriegen*, *bekommen*
AuxModPass	modal passive auxiliary	*sein*
A.c.I.	exceptional case-marking verb	*lassen*, *sehen*, *hören*, *fühlen*
Rais	raising verb (not one of above)	*scheinen*, *pflegen*, *haben + zu*
Cntr	control verb (not one of above)	*wünschen*, *möchten*, *versuchen*, *befehlen*
Verb	main verb (not one of above)	Fritz hat *geschlafen*
Subcategories of R		
Ptcl	particle	*wohl*, *ja*, *noch*
P-ptcl	stranded preposition particle	*da* (from *damit*, *daher*)
V-ptcl	verbal particle	wenn er *weg*läuft
W-ptcl	*was*-*für*-particle	*was* für Menschen
Neg	negative particle	*nicht*
Category-independent PoS tags		
Trace	trace	t
Conj	conjunction	*und*, *oder*, *(so)wie*

Table 1: The SINBAD PoS tagset

Morphological labels are those for case markings on determiners, nouns, and adjectives (nom, acc, dat, gen) and those for inflection on verbs (fin, inf, part (participle), to (to-infinitive)). Nouns and adjectives will only be labelled when having an explicit morphological case marking, i.e., a case affix (different from zero affixation). In contrast to this, determiners always bear a morphological label, even if it is a null determiner. Other morphological categories like person, number, and gender were not relevant in the hitherto recorded sentences, but could easily be added in future applications.

2.2.3 Edge Labels

We distinguish between lexical edge labels and syntactic edge labels. Lexical edge labels are the edge labels directly above the lexical layer and encode additional contextual information as *W* (the specifier of C contains a wh-item), *Rel* (the specifier of C contains a relative pronoun) and *TOP* (the specifier of C is a topicalized phrase).

Syntactic edge labels indicate head-complement or head-adjunct relations between two sister nodes. The node labels together with the syntactic edge labels constitute a minimal residue of X-bar theory. These are the syntactic edge labels:

adjunct	immediately dominates an adjunct
head	immediately dominates a head
rel-head	immediately dominates a relativized head
spec	immediately dominates a specifier
– –	immediately dominates a complement

Typical adjuncts are prenominal adjectives, relative clauses and adverbials. Typical specifiers are the SpecC position, prenominal genitive DPs and possessive pronouns, and the subject of a predicate; these will always be immediately dominated by the edge label spec. The head label is employed to encode a residue of X-bar theory. Any node which is not a head is a maximal projection. This way, categories like NP or CP can be dispensed with: A *maximal projection* NP can be defined as an N-node that is not immediately dominated by the edge label head.

2.2.4 Secondary Edge Labels

Secondary edges denote specific relations between nodes, represented as arrows. We identify four types of constructions or grammatical relations:

move	movement relating a trace to its antecedent
co-ind	co-indexing for the purpose of binding theory relating an anaphora to its antecedent

es-ko	*es*-correlative constructions relating the pronoun *es* to a coreferential, extraposed CP
w-w	*was*-*w*-constructions (partial movement) relating a partially moved wh-phrase to *was*

2.2.5 Null Elements

Although to some extend we avoid the use of empty categories, we still formally distinguish five types of empty lexical items:

pro	the subject of subjectless finite clauses
PRO	the empty subject of an infinitival CP
t_V	the trace of a verb-second movement
t	any other trace
0	any other empty category not mentioned above

pro only appears if there is no other way to satisfy some version of the extended projection principle, i.e., there is no nominative that could be argued to be the subject of a finite clause. In general, this is only the case in impersonal passive constructions. PRO is the subject of CPs headed by C-zu. The remaining zero categories represented by “0” are empty determiners, empty *wh*-operators, empty complementizers and empty conjunctions.

Traces are left by every category that has been moved to another position in the tree. Note, however, that we admit the following exception: In verb-second movement, we decided that the PoS tag of the moved verb in C is V-fin, the PoS tag of the trace is not Trace but the original one of the moved verb. The trace of V/2 itself is marked by t_V to distinguish it from other traces which are always connected with the element which has been moved by a secondary edge label. For perspicuity, we tried to reduce the role of movement to a minimum. For example, subjects may be directly generated in SpecC, without moving from within VP; this allows one to distinguish between genuine topicalizations and normal SVO order.

2.2.6 General Considerations

The annotation schema chosen for our treebank is completely different compared to those for existing German treebanks as TIGER, TüBa-D [8], Tüa-D/Z. These annotation schemes do not reflect a commitment to a particular syntactic theory. The syntactic structures are rather flat and simple and do not contain empty categories or traces. See, for example, the ‘flat clustering principle’ used in TüBa-D and TüBa-D/Z [8, 11] which keeps the number of hierarchy levels in a syntactic structure as small as possible. In the Penn Treebank, empty categories are annotated, but here again a relatively flat context-free notation is used without leaning towards a particular theoretical view.

The advantage of our annotation scheme is that the treebank contains much more information than ordinarily available. Linguistically relevant concepts such as c-command, extraction, pied piping, remnant movement, freezing, and many others are explicitly or implicitly encoded in terms of structure or secondary edges. These concepts are not necessary local and therefore cannot be encoded in other German treebanks; nonetheless they are absolutely crucial for any generative theory of language.

3 Querying Suboptimal Structures

In the treebank presented here, deep syntactic structures are used for the annotation and linguistic information is often encoded implicitly (e.g., the relation c-command). These characteristics pose a specific challenge for query tools and the power of their query languages. We therefore selected the query tool fsq [4] which allows the user to search treebanks for complex syntactic constructions and offers full first-order logic as query language.

3.1 The Query Language of fsq

The properties of a tree in the treebank are expressed as properties of nodes in the tree and relations between nodes. Properties of individual nodes are the annotation labels. That is to say, a nonterminal node has a major category and a grammatical function, which is the syntactic edge label described above. Terminal nodes have part-of-speech labels, lexical edge labels, and tokens and can bear additional morphological information.

Relations between nodes describe (part of) the structure of a tree. Hence, these relations comprise the mother-daughter-relation, also called immediate dominance, the dominance and proper dominance relation, which are the reflexive-transitive and the transitive closure of the mother-daughter-relation. The precedence relations are orthogonal to these, describing the left-to-right orientation in a tree. A node *x* precedes another node *y*, if the whole subtree rooted in *x* is to be found to the left of the whole subtree rooted in *y*. A node *x* immediately precedes *y*, if *x* precedes *y* and there is no node in-between, preceeding *y* and being preceeded by *x*. There can also be secondary relations between nodes, e.g., a move-relation. And one may express equality or disequality of two nodes.

Most of the above described properties of nodes and relations between nodes can be expressed in many existing treebank search engines. The query language of fsq is the full first-order logic over these properties and relations as atomic formulae. More explicitely, the properties and relations are formulae of fsq. The negation of a formula, the conjunction, disjunction and implication of formulae are again formulae. And existential or universal quantification of a node variable and a formula is again a formula. It is in particular the arbitrary quantification that provides the high expressive power of the query language. No other off-the-shelf query tool offers a comparable expressive power, which is often needed for the expression of linguistically important relations. A simple, but frequent example is the description of a complex structure in which a certain undesirable feature is absent. This requires universal quantification over all nodes in the complex structure, because *no* node is supposed to bear the feature.

3.2 C-Command and Remnant Movement

Let us explain the use of the query language by means of two examples that have strong linguistic motivations. The first example is that of *c-command*. This notion plays an important role in the binding theory. Roughly, a node c-commands her sister nodes and all the nodes that her sister nodes dominate. Formally, a node x c-commands another node y if there is a third node z that is the mother of x and that dominates y, i.e., $\exists z(z > x \wedge z >+ y) \wedge \neg x >> y$. The second conjunct excludes cases where x dominates y. The situation is actually a little bit more complicated if the node taking command is a terminal node. Due to the annotation scheme of SINBAD, the preterminal level is unary branching. In other words, the mother of a terminal node x is never the mother of any other node than x. To get to a properly branching node we have to go the the grandmother of a terminal node. Formally $(\neg \exists z\, x > z) \wedge \exists z, w\; w > x \wedge z > w \wedge z >+ y \wedge \neg w >> y$. To consider the terminal and the nonterminal case the disjunction of the two formulae has to be taken. But since the formula for the case of a terminal node has a higher quantifier depth, it should be used only in those circumstances where it is needed. Often linguists consider a c-command relation between nonterminal nodes, and in this situation, the simple formula stated first suffices.

Remnant movement describes the leftward movement of a complex structure out of which a smaller substructure is already moved. Consider Figure 1 as an example. Here, the complex VP [*Damit gerechnet*] is moved into the topic position of the sentence, which is the specifier of the CP. This complex VP contains the trace of the particle *da* which was moved out of the VP before the VP is moved. A necessary precondition for this type of construction is that the landing position of the smaller structure is c-commanded by the landing position of the large structure out of which it was moved. Due to the fact that movement is explicitely annotated in the treebank via a secondary edge with label move, it is simple to search for instances of remnant movement in the treebank. Remnant movement of node *x*, where *x* is the root of the complex structure that is moved, can be expressed by the following formula: $\exists y\, \mathsf{move}(y,x) \wedge \exists w,z\ \mathsf{move}(z,w) \wedge x >> z \wedge x$ c-commands *w*. The first conjunct expresses the existence of a move-secondary edge that ends in *x*. The second conjunct expresses the movement of the smaller substructure. It is moved from node *z* which is dominated by *x* to a landing position *w* that is c-commanded by *x*.

3.3 The Web Interface

The treebank is available on the web under the following URL: `http://barlach.sfb.uni-tuebingen.de/~a3/`. This site gives access to a structural search as well as to a keyword search to be described below.

The tree structure search is realized as a web interface to fsq. Part of fsq is a graphical user interface that systematically supports users in constructing queries. When composing a query most users think in a bottom-up fashion focusing first on the atomic constituents. This approach is supported by the user interface in the following way. An *Atomic* menu lets the user compose atomic formulae. He picks the relation of his choice, say, e.g., the dominance relation. He is successively asked for names of the variables one dominating the other. Thereafter, the syntactically correct formula is added to the list of formulae. The other atomic formulae can be constructed in a similar fashion.

In order to get more complex formulae, the user can choose operations from the *Complex* menu. It contains menu options for the boolean connectives and quantifiers. To compose, e.g., a conjunction, the user first chooses the formulae he wishes to conjoin by clicking on them in the list of formulae. Thereafter he just picks the *Conjunction* menu item and the conjunction of the formulae he chose is added to the list of formulae. In case of an existential or universal quantification, the user selects a formula from the list, and, e.g., the *Existential Quantification* menu item. He will be asked for the name of the variable to quantify over, and the existentially quantified formula is added to the list of formulae.

We transformed the graphical user interface of fsq into an applet and modified it for our purposes. With the help of the applet, queries can be composed, edited and submitted from a standard browser. In addition to the predicates and relations provided by fsq, we offer macros encoding such linguistic constructs as c-command, head relation, extraction, and remnant movement, which can be combined with other relations, thus forming complex queries. These are transmitted from the applet to a cgi-script, which starts the fsq engine and displays the retrieved sentences in HTML format.

Linguists however are not exclusively interested in searching a collection of suboptimal tree structures, they of course are also interested in additional information as the grammaticality rating of a given sentence, the reference source, etc. Therefore the treebank is embedded in a larger system that also comprises a MySQL database containing these informations. Accordingly, for each sentence retrieved by the tree structure search, the user can request the syntactic annotation as a tree, the source, the set of structurally similar sentences and their ratings as given by the author.

The database also contains an extensive description of each tree in the form of a set of keywords. The keywords are grouped into six areas of linguistic properties of a tree: wh-movement, topicalization, scrambling, binding, extraposition and dislocation, and complementation. For each area, there exists a fine grained list of potential features. As an alternative way to search the treebank we provide a web interface to this keyword database.

Keyword search is simple and may be more appealing to novel users of the treebank. But it provides access only to a proper subset of the structural properties of trees in the treebank. Every keyword search can also be performed by an fsq query. But there are interesting complex queries that cannot be expressed by keyword search.

4 Conclusion

We presented a treebank of suboptimal structures in German. The novelty of the present work is threefold. Our treebank is the first treebank for German that provides analyses of trees within the framework of generative grammar. It is also the first treebank to provide suboptimal sentences together with their grammaticality judgments. It is therefore of high importance for generative linguistics of German. To offer an open access to the treebank we subplanted the treebank with a very powerful query system that is accessible via the web. It is especially this accessibility that makes the treebank so useful for linguists. Future developments include an extension of the size of the treebank and an implementation of techniques to shorten query response times.

Acknowledgments

Kudos goes to Monika Toth, who annotated the trees in the treebank, and to Tanja Kiziak, who is a co-developer of the web interface of fsq. We would also like to thank Sam Featherston and Tanja Kiziak for interesting discussions and helpful comments. This research was funded by a grant from the German Research Council (DFG SFB-441).

References

[1] Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. Bracketing Guidelines for Treebank II style Penn Treebank Project. Technical report, University of Pennsylvania, 1995.

[2] Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. The TIGER Treebank. In Kiril Simov, editor, *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol, 2002.

[3] Hubert Haider. *Deutsche Syntax – Generativ*. Narr, Tübingen, 1993.

[4] Stephan Kepser. Finite Structure Query: A Tool for Querying Syntactically Annotated Corpora. In Ann Copestake and Jan Hajič, editors, *Proceedings EACL 2003*, pages 179–186, 2003.

[5] Gereon Müller. *Incomplete Category Fronting. A Derivational Approach to Remnant Movement in German*. Number 42 in Studies in Natural Language and Linguistic Theory. Kluwer, 1998.

[6] Oliver Plaehn and Thorsten Brants. Annotate – An Efficient Interactive Annotation Tool. In *Sixth Conference on Applied Natural Language Processing (ANLP-2000)*, 2000.

[7] Anne Schiller, Simone Teufel, and Christine Thielen. Guidelines für das Tagging deutscher Textcorpora mit STTS. Manuscript, Universities of Stuttgart and Tübingen, 1995.

[8] Rosmary Stegmann, Heike Telljohann, and Erhard Hinrichs. Stylebook for the German treebank in VERBMOBIL. Technical Report 239, SfS, University of Tübingen, 2000.

[9] Wolfgang Sternefeld. Syntax. Eine merkmalsbasierte generative Analyse des Deutschen. Book manuscript, 2004.

[10] Wolfgang Sternefeld. The SINBAD Stylebook – **S**ammlung **IN**teressanter **B**eispiele **A**us'm **D**eutschen. Technical report, SFB 441, University of Tübingen, 2004.

[11] Heike Telljohann, Erhard Hinrichs, and Sandra Kübler. Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z). Technical report, Seminar für Sprachwissenschaft, Universität Tübingen, 2003.

Annotations for Zero Pronoun Resolution in Korean Using the Penn Korean Treebank

Sun-Hee Lee, Donna K. Byron, & Whitney Gegg-Harrison

Computer Science and Engineering at the Ohio State University

1. The Problem of Null Elements

In topic prominent languages like Korean and Japanese, a repeated element has no surface realization, called a *zero pronoun*, in contexts where an explicit pronoun would be used in English. Compare example (1) and (2):

(1) a. John watched the children on the street.
b. Mary watched *them*, too.
(2) a. John-i kil-eyse aitul-ul po-ass-ta.
John-NOM street-on children-ACC see-PAST-END
'John watched the children on the street.'
b. Mary-to Ø po-ass-ta.
Mary-also OBJ see-PAST-END
'Mary also watched (zero=them).'

This property of Korean creates an issue for developers of Treebanks and other annotated language resources: when and how should these unrealized elements be explicitly introduced into the linguistic material being developed? Although this question could be asked when working with any language, in so-called pro-drop languages such as Korean, Japanese, Spanish and Portuguese, the problem is more immediate, because basic units of sentence structure, such as subjects of matrix clauses, are frequently unrealized.

Although researchers, especially the anaphora resolution community, have been eager to study the distribution of zero pronouns and their link to other discourse properties, there has historically been a lack of annotated material available to the wider research community that would allow us to investigate these questions. Researchers in the past worked mainly with small amounts of hand-constructed data rather than being able to do large-scale corpus analysis. This lack has been recently addressed by the release of the Penn Korean Treebank (Han et. al. [5]; henceforth PKT), which includes annotations indicating the position of zero pronouns. Our initial attempts to utilize the PKT as test data for a zero pronoun resolution algorithm have unfortunately revealed several problems in the construction of the trees and the positions of zero pronouns in the PKT. These problems need to be resolved before the PKT can fulfill its potential as a source of linguistic knowledge about zero pronouns, and they should be taken into consideration by other teams developing similar resources in other languages.

The work described here is primarily motivated by our efforts to develop an annotation scheme for zero pronouns in Korean that can be used to develop a gold standard for testing an anaphora resolution algorithm. Previous authors have pointed out that the antecedents of zero anaphors can often be determined by using various grammatical properties such as topicality, agreement, tense, and aspect as well as subcategorization information (Walker et al. [12]; Iida [8]; Hong [7], etc.). However, in order for these factors to be useful in developing anaphora resolution algorithms, they must be reliably and consistently annotated into the source data. In this paper, we will examine the necessity for a new annotation scheme that provides relevant semantic and discourse factors for reference resolution in the Korean Treebank. Our analysis can be extended to naturally occurring discourse that shows more complicated discourse properties and structures. The revised annotation scheme will be adopted for our new software handling zero anaphors in languages like Korean.

2. Identifying Positions of Zero Pronouns in the Penn Korean Treebank

Why mark zero pronouns in the PKT? Unrealized arguments are important for tracking the attentional state of a discourse in topic-oriented languages like Korean and Japanese. Within the framework of centering theory, e.g. Walker et al. [12], Iida [8]), Hong [7], etc. it has been shown that a salient entity recoverable by inference from the context is frequently omitted, and therefore interpreting these zero anaphors allows one to follow the center of the attentional state. Walker et al. [12] applied the centering model, developed for pronoun resolution in English, to zero anaphor resolution in Japanese. They argue that interpretation of a zero anaphor is determined by discourse factors. This suggests that identifying occurrences of zero anaphors and retrieving their antecedents are important in developing a computational model of discourse interpretations as well as syntactic and semantic analyses.

2.1 Obligatory vs. Optional Arguments

The first crucial step for zero anaphor resolution is to identify the exact positions zero pronouns. According to the guidelines of the PKT, only a missing obligatory argument should be annotated as an empty element. Missing optional arguments are not. Thus, missing subject or object elements were marked as zero pronouns while missing locative arguments, such as *hakkyo-ey* in (2), were not marked when they were omitted.

```
(2)  wuli-nun    eycey        hakkyo-ey     ka-ss-ta.
     we-Top      yesterday    school-to     go-Past-Decl
     'We went to school yesterday.'
```

However, the annotation method based on an obligatory vs. optional argument may result in the loss of crucial information needed at later stage of retrieving an antecedent of a zero element. For example, the locative argument, 제45 사단에 'the 45th division-in' has not been marked up as a zero anaphor in a tagged sentence of (3) B.

```
(3) A:  제     45    사단은 또   무엇으로 구성되어 있는가 ?
        the    45    division again with what composed   be
        'What is the 45th Division composed of ?'
        (S (NP-SBJ  제/XPF+45/NNU
                          사단/NNC+은/PAU)
        (VP (VP (ADVP  또/ADV)
                  (VP (NP-COMP  무엇/NPN+으로/PAD)
                       (VV  구성/NNC+되/XSV+어/EAU)))
                          있/VX+는가/EFN)
                               ?/SFN)
    B:   사단    지휘부가 있습니다 .
        division head-Nom    exist
        'The head division is (there).'
        (S (NP-SBJ  사단/NNC
                   지휘부/NNC+가/PCA)
           (ADJP  있/VJ+습니다/EFN)
                            ./SFN)
```

In the given discourse segments, the adjective *있다 issta* requires a locative argument which has been treated as an optional argument in the PKT. Thus, there is no information on the empty element in (3B). However, this sentence is uninterpretable unless the missing optional element is recovered.

Due to some unique properties of the PKT, we believe that the importance of missing optional arguments has been underestimated. The annotated Treebank consists of texts from military language training manuals with 54,000 words (Han et al. [5]), where most sentences are composed of a question and an answer between a military personnel and a captive. Thus, the discourse structure and conversational flow of the corpus are rather artificial and simpler than natural dialogs would be. In the Korean Treebank, occurrences of zero anaphors generally depend on the local contexts of the preceding utterance. However, in naturally occurring discourse, zero pronouns are licensed not only in local contexts but also in global contexts. In example 4, the last zero anaphor corresponding to a missing optional argument refers to ‘Jeju island’ that appears in the first utterance.

(4) A: kot Jejudo-ey-nun yuchaykkoch-i phil kesita.
soon Jeju island-in-Top yellow rape flowers-Nom bloom will
‘Soon, yellow rape flowers will bloom in Jeju island’
B: manhun kwankwangkayk-i Ø molyetul-kess-ci.
many tourists gether-Fut-Decl
‘Many tourists will flock (into there).’
A: Na-to kulehkey sayngkakhay.
I-also so think
‘I also think so.’
B: owel-ey-nun yehayngkayktul-i tewuk Ø pwumpinta.
May-in-Top travelers-Nom more crowd
‘In May, more travelers crowd (there).’

In order to maintain consistent annotations of zero anaphors and develop a reference resolution system, we argue that missing optional arguments need to be marked as zero pronouns. For this process, it is essential to check interannotator agreement and to use constant subcategorization frames of predicates. Dictionaries with specific subcategorization information can be used here, such as the Yonsei Korean dictionary, where different subcategorization frames are listed according to semantically disambiguated senses for each predicate.

2.2 Errors in the Penn Korean Treebank

We examined the current annotations of PKT and found some significant problems with respect to zero elements. For this study, we examined only 100 sentences from the Treebank. Those 100 sentences are composed of dialogue sequences and contain 133 occurrences of zeros. Among 133 zeros, 29 tokens should not have been marked as zero pronouns. In 15 cases, the sentence was marked as having a zero subject although subject NPs overtly appear in the same sentence. The problem arises due to an ambiguous case marking on nouns referring to collectives or groups. While the nominative case markers *ka* and *i* normally indicates subject nominals, groups or organizations appearing in subject position use the case marker *eyse* as in (5)[1], thus the case marker *eyse* is treated as a nominative marker in Korean grammar. In the 15 cases, subject nominals referring to a group or organization were mistakenly parsed as NP adverbials rather than as subjects.

(5) hakyo-eyse John-eykey sang-ul cwuessta.
school-NOM John-to prize-ACC awarded
'The school awarded a prize to John.'

This error significantly increases the number of zeroes in PKT, since it is composed of military training materials with frequent mention of group nominals such as *taytay* 'squadron', *sotay* 'platoon', *yentay* 'regiment', etc.

In addition, missing subjects have been assumed for some conventional fixed expressions that are similar to semi-idiomatic phrases. For example, the usages of *-ey tayhayse* 'regarding on', *-ey uyhayse* 'by' and *-ul wihayse* 'for the sake of' are conventional in that their meanings refer to something other than the compositional sum of their parts. In traditional Korean grammar, the verbs like *tayhata*, *uyhata* and *wihata* are classified as incomplete verbs that do not require a subject and have restricted inflectional forms. We found 14 examples of missing subjects in the subject positions of these incomplete predicates. Similar to subjectless incomplete verbs, it is well known that there exists a class of subjectless verbs in Korean which include modal predicates such as *sayngkita, moluta, toyta.*[2] However, empty subject positions have been assumed for these subjectless predicates in the parsed outputs of the PKT. Therefore, unnecessary zero elements appear in the Treebank and may lead to inaccurate conclusions about the distribution of zero pronouns in Korean.

3. Classification of Zero Pronouns

Once the positions of zero pronouns are identified, further information can be specified to indicate the antecedent, when there is one, and other properties of the reference. With respect to anaphor resolution, Han [6] proposed a classification of zeros and an annotation scheme based on Korean text corpora including the PKT. In this section, we will discuss some problems that we found in the current scheme and argue for a rather revised classification for annotating zeros.

[1] Refer to Yonsei Korean Dictionary, and Nam & Ko [10] etc.

[2] Kim [11] analyzes some modal verbs as subjectless verbs.

3.1. Han [6]'s Classification

In order to build an annotated corpus that can be used for developing a pronoun resolution system, Han [6] proposes a classification of zero elements in Korean as in Table 1.[3]

text dependent use	discourse anaphoric	Propositional
		NP anaphoric
text independent use	zero deictic and zero indexical	
	indefinite null pronoun	Generic (nonspecific)
		Specific
	general situational null pronoun	

Table 1: Han's Classification of Korean Zeros

Furthermore, she proposes the coding scheme for annotation as follows.

(i) NP-anaphoric elements are linked to their antecedents by the addition of a numeric index indicating the coreference class.

(ii) Other categories are marked with an alphabetic index:

-deictic speaker: i   -deictic hearer: y
-deictic speaker and hearer: w
-indefinite generic: g   -indefinite specific: s
-situational: x   -anaphoric propositional: p

One of advantages in Han's coding system is that it allows dual markings for zeros that can be interpreted as deictic and anaphoric at the same time. Thus, the following occurrence of a zero element can be marked as deictic and also anaphoric.

(6) *nai*$_{i5}$ nun kakkum ku secem-ey kanta. Ø$_{i5}$ hoksi ku-lul mannako sipheseita.
I$_{i5}$ occasionally stop by at the bookstore. It's because (I$_{i5}$) want to see him.

Han's system, however, introduces some unnecessary complexity in classifying zeros, which may complicate the annotation scheme and interannotator agreement. We will discuss relevant problems in the next section and will

[3] Han's classification is similar to Kameyama [9]'s classification of Japanese zero anaphors except that Han classifies discourse anaphoric zeros into propositional anaphoric zeros and NP anaphoric zeros while dividing indefinite personal zero anaphors into indefinite generic zeros and indefinite specific zeros.

propose more simplified version of zero classification.

3.2. Our Proposed Zero Anaphor Classification

We classify Korean zeros into three different classes as shown in Table 2; discourse anaphoric zeros, deictic and indexical zeros, and indefinite zeros.

Discourse Anaphoric Zeros	Individual Entities
	Eventualities
	Propositions
Deictic and Indexical Zeros	
Indefinite Zeros	

Table 2. Categories of Korean Zeros

The discourse anaphoric zeros take their reference from antecedents in the previous utterances in the given discourse. This class is the main one that anaphor resolution systems aim to handle. As for discourse anaphoric zeros, there are three subclasses. The first refers to individual domain entities, the second, eventualities, and the third, propositions. The first and the third subclasses correspond to Han [6]'s NP anaphoric zeros and propositional zeros. The zeros of individual entities refer to entities that were introduced into the discourse via noun phrases. The zeros of propositions refer to propositions introduced in the previous utterance. Relevant examples are provided in (7) and (8).

(7) A: *i* *taytay-uy* *yepi* *hochwul penho*-nun mwunka?
2 squadron-GEN provisional watchword-TOP what
'What is the watchword of the 2nd squadron?'
B: Ø "chwutong"-ipnimta
SUBJ chwutong-COP
'It is Chwutong.'
(Ø = 'the watchword of the 2nd squadron')
(8) A: 108 yentay cihwipwu-nun hyencay eti-ey wichihako issnun-ka?
108 regiment headquarter-TOP now where-at locate being-Q
'Where is the headquarter of the 108th regiment located?'
B: Ø$_1$ Ø$_2$ molukeyss-supnita.
SUBJ OBJ not know-END
'I don't know.'
(Ø$_1$ = 'B', Ø$_2$= 'Where the headquarter of the 10th regiment is located ')

While Han distinguish only two kinds of discourse anaphoric zeros, there exist another kind that does not belong to either of her two categories. Let us consider the following examples:

(9) A: aitul-i   sihem-ul   chi-ko   sipheha-ci   anha.
    children-NOM   exam-ACC   take-END   want-END   don't
    'Children don't want to take an exam.'
  B: na-to   Ø   silhe.
    I-also   hate
    'I also hate it.'   (Ø = the action of taking the exam)
(10) A: kyothong-i   wenhwalha-myen wuli-nun   nuc-ci   ahulkeya.
    traffic-NOM   smooth-if   we-TOP   late-END not
    'We won't be late if the traffic is smooth.'
  B: kulssey,   i   sikan-ey-nun   Ø   himtul-kel.
    well   this   hour-in-TOP   tough-will
    'Well, it will be tough at this time.'
    (Ø = 'the event that the traffic is smooth')

In (9) and (10), zeros do not refer to an individual or a proposition. They refer to eventualities, i.e. action and event (Asher [2]). Thus, we categorize non-entity referring zeros as zeros of eventualities.

The second class of zero anaphors includes deictic and indexical zeros that directly refer to entities that can be determined in the given spatiotemporal context, which generally include a speaker and an addressee. The third class includes indefinite zeros referring to general people, which corresponds to *they*, *one*, and *you* in English.

Our classification of zeros, however, does not contain categories of situational zeros that have been assumed both in Kameyama [9] and Han [6]. Han provides (11) as an example of situational zeros.

(11) Ø   pelsse   yel-si-ta
   SUBJ   already 10 o'clock-COP
   'It is 10 o'clock already.'

As mentioned in the previous section, Korean allows subjectless constructions. While a dummy subject, *it* or *there* is required for sentences referring to time, weather, or the situation in English, the subject position does not need to be filled in Korean.[4] Therefore, there should not be a zero element

[4] Han argues that the following example supports her category of situational zeros.
(i) cikum   sikan-i   yel-si-ta.
  now   time-NOM 10 o'clock-COP
  'Now, it is 10 o'clock.'

in the subject position of (11).

Another classification that we do not adopt is Han's distinction between indefinite specific zeros vs. indefinite generic zeros.[5] We instead categorize indefinite zeros as one category. This is because it is not clear that indefinite zero elements themselves are ambiguous. The indefiniteness of zeros is closely tied to the semantic interpretation of a sentence as Han also admits. For bare plurals in English, Carlson [4] derives generic interpretations by using a generalization operator, *Gn*, without assuming ambiguity of bare plurals. A similar account can be applied to Korean zeros. In addition, Amaral [1] shows that generic vs. specific interpretations of zeros in European Portuguese, a *pro*-drop language, can be constrained by semantic-pragmatic factors. With respect to empirical issues, it is difficult to distinguish interpretations of indefinite specific zeros vs. generic zeros, which will increase interannotator disagreement. Thus, we decide not to distinguish indefinite specific and generic zeros while leaving a more theoretical discussion as a separate study.

However, in Korean multiple subjects can be licensed based on various semantic relations between nominative NPs such as possessor-possessed, part-whole, class-member, etc. For example, the sentence (i) can be extended into (ii).

(ii) New York-i cikum-i sikan-i yel-si-ta.
New York-NOM now-NOM time-NOM 10 o'clock-END
'Now it is 10 o'clock in New York.'

Thus, the existence of *sikan-i* 'time-NOM' does not support the existence of a zero subject for the predicte *yel-si-ta*, which does not require that the subject position be filled.

[5] Examples for indefinite specific zeros and indefinite generic zeros have been provided by Han [6].

(i)Ø kyupu-nun kongsangwahakyenhwa-uy thul-ul pilie milo-uy
Cube- TOP SF movie-GEN frame-ACC adop labyrinth-GEN
talchwul-ilanun sinhwa-lul yenghwa-lo mantu-n cakpwum-i-ta.
escape-QUOTE mith-ACC movie-as make-REL work-COP-END
'Cube is a work that (someone) made the myth of an escape from a labyrinth
into a movie adopting a SF movie framework.'

(ii) paopap-namwu-nun Ø cachic nuckey son-ul ssu-myen kuttayn
paopap-tree-TOP SUBJ possibly late hand-ACC use-if finally
cengmal Ø Ø chechihal swu epskey toynta.
really SUBJ OBJ mangage way impossible become
'As for Baobab trees, they become impossible (for one) to manage (them) if
(one/he) treat them late.'

While the zero subject in (i) refers to someone specific, who is the movie maker, the zero subject in (ii) has an arbitrary reference. One notable thing is that in (i) the movie title, *Cube* and possibly the whole situation or script triggers a sort of bridging inference (Clark [3]) and restricts the reference of the zero subject. In contrast, there exist no clues for the referent of the zero subject in (ii).

4. Some Useful Features for Anaphor Annotations

Based on our coding system for Korean zeros, we now argue that more sophisticated information relating to semantic and discourse properties need to be added to the annotated corpora like the PKT. This will increase the applicability of the annotated corpora to both theoretical research on anaphors and computational modeling of anaphor resolution.

4.1. Topic Information

One crucial discourse factor that has an effect on determining an antecedent of a zero anaphor is the existence of topics. In Korean, a topic appears in a sentence initial position with the topic marker 는 *nun*. While the marker *nun* functions as a topic marker in a sentence initial position, it also works as an auxiliary postposition in a non-initial position of a sentence. The first is classified as a grammatical topic marker while the latter is a contrastive topic marker. According to the current annotation scheme of the PKT, the two kinds of topic marker *nun* are treated as the same auxiliary postposition, which is similar to other postpositions *man* 'only', *to* 'also', and *mace* 'even'. Structurally, a subject NP with a topic marker has been analyzed as the subject, while a topic marked object is treated as a scrambled argument out of its canonical position. Although grammatical topics are not independently annotated in the PKT, they are closely associated with interpretations of zero anaphors as in (12).

(12) 1. A: mwucenkiyoung chwukencci-nun pothong elmana olay kanun-ka?
radio batteries-TOP usually how long last-Q
'As for batteries of radios, how long do they last usually?'
2. B: kuken Ø elmana manhi ssununka-ey tallye iss-ci-yo.
that how much use-On depend exit-END-HON
'It depends on how much (we) use.'
3. B: pothong **Ø** han ilcwuil cengto kap-nita.
usually approximately a week about last-HON
'Usually, **(they)** last for about one week.'

The zero subject in (12.3) refers to the topicalized subject of (12.1). Walker et al. (1994) provides evidence that topic marked elements function as antecedents of zero anaphors in Japanese. Besides having similar morphosyntactic properties and sentence structure, Korean and Japanese also share the property that interpretations of zero anaphors are connected to grammatical topics. Therefore, we argue that the topic marker needs to be differentiated from other postpositions and that grammatical topics are to be differentiated from other grammatical arguments like subjects and objects.

4.2. Morphosyntactic Information of Speech Acts

To provide a potential source for zero anaphor resolution, verbal suffixes representing sentence types are useful to add to the annotated corpora. This is because they are associated with certain speech acts such as declaration, request, question, promise, etc. and information of a missing subject can be retrieved from verbal morphology. There exist five different types of verbal inflections respectively representing a different sentence type; declaratives, interrogatives, imperatives, propositives, and exclamatives.[6] Among them, the imperative verbal endings suggest that a missing subject tends to refer to the hearer while promising verbal endings imply that a missing subject is the speaker. For example, the missing subjects of the following examples are respectively interpreted as *you*, *I*, and *we* based on the verbal suffixes representing a particular speech act.

(13) a. Ø ca-ni. (Question)
sleep-Q
'Are (you) sleeping?'
b. Ø ca-llay. (Declaration)
seep-will
'(I) will sleep.'
c. Ø ca-ca. (Request)
sleep-let's
'Let's sleep.'

More specific classification of verbal endings could enhance the process of determining an antecedent of a zero anaphor subject. In the current annotations of the PKT, verbal suffixes are simply categorized together as final endings although the five subclasses according to sentence types have been recognized in the tagging guidelines. We propose to mark the five classes of verbal suffixes differently for developing an anaphor resolution system.

[6] Some examples of final suffixes matching each sentence type are listed as follows.
1. Declaratives: *-ta*, *-ney*, *-o*, *-pnita*, *-nunta*, *-ci*, etc.
2. Interrogatives: *-ni*, *-kka*, *-yo*, *-nunka*, *-pnikka*, *-e*, etc.
3. Imperatives: *-la*, *-psiyo*, *-ela*, *-key*, *-o*, etc.
4. Propositives: *-ca*, *-psita*, *-cakkwuna*, *-sey*, etc.
5. Exclamatives: *-kwuna*, *-ney*, *-kwun*, *-ela*, etc.

4.3. *Wh*-pronoun Information

Also useful for anaphor resolution is *wh*-element information. *Wh*-elements in Korean include *nwuka* 'who', *mwues* 'what', *encey* 'when', *etise* 'where', *way* 'why', *ettehkey* 'how', etc. Answering utterances for the *wh*-questions generally contain zero elements, whose antecedents can be found in the preceding questioning utterances. In general, a fragment directly related to a *wh*-element while non-*wh*-elements previously mentioned easily drop as shown in (14).

(14) A: John-i Min-ul mwe-la-ko mitko iss-ni?
John-NOM Min-ACC what-COP-COMP believe being-Q
'What does John believe Min to be?'
B: Ø Ø kyoswu-la-ko mitko iss-nuntey.
SUBJ professor-COP-COMP believe being-END
'(He) believes (her) to be a professor.'

Since question-answer pairs have similar argument structure, the referents of the subject and the object in (14) can be easily retrieved from the preceding question sentence. However, *wh*-elements are not distinctly tagged from other pronouns in the PKT.

Based on our classification of zero anaphors and relevant properties for anaphor resolution in Korean, an annotation example with a parsed syntactic structure of the Korean Treebank is given as follows (* indicates newly introduced annotations).

(16) A: 제45 사단은 어느 부대에 예속하는가 ?
'Which army does the 45th division belong?'
(S (NP-SBJ 제/XPF+45/NNU
사단/NNC+은/***TOP***)
(VP (NP-COMP 어느/DAN/***WH***
부대/NNC+에/PAD)
(VV 예속/NNC+하/XSV+는가/***Q***))
?/SFN)
B: 제6 군단-에 예속합니다.
'(It) belongs to the 6th corps.'
(S (NP-SBJ ***pro-individual***)
(VP (NP-COMP 제/XPF+6/NNU
군단/NNC+에/PAD)
(VV 예속/NNC+하/XSV+ㅂ니다/***DEC***))
./SFN)
A: 6 군단을 또 어떻게 부르는가?
'How do you call the 6th corps?'

```
    (S (NP-SBJ *pro-deictic/individual*)
       (VP (NP-OBJ 6/NNU
                   군단/NNC+을/PCA)
           (VP (ADVP  또/ADV)
           (ADVP  어떻게/*WH-ADV*)
           (VP  부르/VV+는가/*Q*)))
     ?/SFN)
  B:  모르겠습니다  .
     (S (NP-SBJ *pro-deictic*)
        (VP (NP-OBJ *pro-proposition*)
            모르/VV+겠/EPF+습니다/*DEC*)
        ./SFN)
```

5. Conclusion

In this paper, we examined a classification system of zeros in Korean and an annotation scheme applicable to zero anaphor resolution systems. We evaluated the current annotation system of the PKT and Han [6]'s proposal for an annotation scheme based on the Korean Treebank. Specific problems related to zero annotations and morphosyntactic analysis in the PKT have been discussed. We showed what kind of morphosyntactic information needs to be added to a newly developed annotation scheme. This newly developed annotation scheme will be adopted for our new software handling zero anaphors in languages like Korean.

References

[1] Amaral, P. 2004. Inferrables with pronominal subjects in European Portuguese: implications for theories of discourse anaphora. *Proceedings of the 5th Discourse Anaphora and Anaphor Resolution Colloquium*, 1-7.

[2] Asher, N. 1993. *Reference to Abstract Objects in Discourse.* Kluwer Academic Publishers.

[3] Clark, H. 1977. Bridging: In P.N. Johnson-Lard & P.C. Wason Eds. *Thinking: Readings in Cognitive Science*, 411-420. Cambridge:Cambridge University Press.

[4] Carlson, G. 1997. A unified analysis of the English bare plural. *Linguistics and Philosophy,* 1:3, 413-458.

[5] Han, C-H., N-R Han, and M. Palmer. 2002. Development and Evaluation of a Korean Treebank and its Application to NLP. *Proceedings of the 3rd International Conference on Language Resources and Evaluation* (LREC, 2002)

[6] Han, N-R. 2004. Korean null pronouns: classification and annotation. *Proceedingsof the ACL 2004 Workshop on Discourse Annotation*, 33-50.

[7] Hong M. 2000. Centering theory and argument deletion in spoken Korean. The Korean Journal of Cognitive Science, 11-1: 9-24.

[8] Iida, M. 1998. Discourse coherence and shifting centers in Japanese texts. In Walker, M.A., A.K. Joshi, and E.F. Prince, (eds.), *Centering theory in discourse*. Oxford University Press, Oxford: UK.

[9] Kameyama, M. 1985.*Zero Anaphora: the Case of Japanese*. Standford University Doctoral Dissertation.

[10] Kim, J-E. 1999. *A Study of Korean Modal Predicates* (*Wulimal Yangthey Yongen Yenkwu*). Thayhaksa. Seoul:Korea

[11]Nam, K-S., and Y-G. Ko. 2002. *Standard Korean Grammar* (*Phyocwun Kwuke Mwunpeplon*). Top Publishing Co.

[12]Walker, M.A., M. Iida, and S. Cotes. 1994. Japanese discourse and the process of centering. *Computational Linguistics* 20(2): 193-232

[13]Dictionary *Yonsei Korean Dictionary*. 1999. Dong-A Publishing Co.

Extracting Negative Polarity Items from a Partially Parsed Corpus [*]

Timm Lichte
Collaborative Research Centre 441
University of Tübingen

Manfred Sailer
University of Göttingen

1 Introduction

In this paper we will address a special group of lexical elements which show a particular affinity with negative contexts. Such elements, usually referred to as *negative polarity items* (NPI), have been widely studied in the linguistic literature since [4]. The classical example of an NPI is the English indefinite determiner *any*. As demonstrated in (1) a sentence containing *any* and negation is grammatical. Without the negation the sentence is ungrammatical. Following standard terminology we will refer to the negation as the *licenser* of the NPI. We will underline NPIs and print the licensers in bold face.[1]

(1) a. He has**n't** seen any students.
b. *He has seen any students.

Since we will be focussing on German an analogous German example is presented in (2). These sentences only differ in that sentence (a) contains a so-called *n-word* as licenser and in (b) there is no exponent of negation; thus the NPI *jemals* (ever) is not licensed.

(2) a. **Niemand** von uns war jemals im Jemen.
nobody of us was ever in Yemen
('None of us has ever been to Yemen.')

[*] We are grateful to Tylman Ule and Lothar Lemnitzer for comments and to Guthrun Love for her help with the English.

[1] There is a particular use of *any*, called *free-choice any*, which does not require a negative operator such as **not**. Nevertheless this use has a restricted distribution, i.e. it requires a context which is *nonveridical* ([14],see section 2).

b. *Einer von uns war jemals im Jemen.
One of us was ever in Yemen

The inventory of NPIs in English and Dutch has been documented fairly well. [3] presents about 250 Dutch NPIs. For German the state of documentation is less ideal. There are only two relatively extensive lists: [13] and [5], neither of which comes even close to the data collected by Hoeksema.

In this paper we will present a first step towards an automatic corpus-based compilation of a list of German NPI candidates. To our knowledge [11] was the first person to point out explicitly that the relation between an NPI and its licenser bares similarities to the relation between a collocate and its collocator. This idea, then pursued in [12], represents the basic motivating insight for this paper.

In Section 2 we will summarize the semantic literature on NPIs. These insights are applied to extract NPI candidates consisting of a single lexical element in Section 3, and in Section 4 to extract complex NPI candidates.

2 Linguistic Aspects

In this section we will present a summary of those aspects of NPIs which are directly related to our study.

Negation is understood as a logical operator which imports special entailment properties to the semantic value of an attached sentence. The literature on NPIs distinguishes several degrees of negativity based on their formal semantic properties.[2]

We will concentrate on downward entailing operators to identify negative contexts without differentiating operators of stronger negation. Downward entailing operators import a fairly weak degree of negativity and thereby include also the operators of stronger degrees. Most NPIs require at least a downward entailing context and the importance of downward entailingness for NPI licensing has been remarked on in [6]. Genuine downward entailing operators include words such as *höchstens* (at most), *kaum* (hardly) or *wenige* (few). A downward entailing context allows one to reason from sets to subsets as demonstrated in (3):

(3) **Few** congressmen eat vegetables.
$\|\text{spinach}\| \subseteq \|\text{vegetables}\|$
Few congressmen eat spinach.

An even weaker notion of negativity is that of *nonveridicality* ([1, 14]). Roughly

[2] See [12] for an introduction to the necessary formal semantic properties of negative contexts and NPIs with rich data.

put, if a statement is in the scope of a nonveridical operator, then the truth of the statement is not implied, but reasoning from sets to subsets is not possible in general. Nonveridical contexts are triggered by direct or indirect questions, free relatives, and also by adverbials such as *vielleicht* (perhaps). Since this category appears to be rather large, we will only include interrogatives in our considerations.

Although we will ignore the differences for the time being, it should be noted that NPIs can have different distributional patterns along the degrees of negativity, which make it possible to distinguish different subclasses of NPIs.

[15] mentions the modal verb *brauchen* (need) as an NPI that is compatible with downward entailing triggers, but excluded from questions.

(4) *__Wer__ braucht Brot zu kaufen__?__
who needs bread to buy

An NPI which can occur in all the above-mentioned contexts is *jemals* (ever). Note that it is still an NPI because it is excluded from sentences without a licenser, as demonstrated in (2).

Since we are only interested in finding NPIs we will skip the subclassification topic and concentrate on downward entailing contexts and interrogative constructions, although subclassification naturally follows acquisition.

3 Extraction of NPI Candidates

After having established the theoretical framework for our empirical study of German NPIs, we can now proceed to the actual corpus work. Some remarks on problematic corpus-related aspects of NPIs will be made at first in section 3.1. Section 3.2 discusses the corpus and the methods which we employed. Section 3.3 briefly goes into problematic cases of clause annotation. The extracted candidates will be presented in Section 3.4. In Section 3.5 we evaluate several quantitative measures for our data.

3.1 Remarks on Feasibility

Starting from English examples, [2] stresses several problematic aspects of automatic detection of NPIs. One of the most severe ones is polysemy, which “is rampant among polarity items”, i.e. NPIs “are nearly always the evil negative twin of some perfectly innocent nonpolarity item”. To account for various uses of NPIs, Hoeksema regards word-class tags and information about the structure of complements (“parsed corpus”) as prerequisites for a text-corpus on which polarity extraction methods are applied. TüPP-D/Z, the corpus we are using (see the following section), contains part-of-speech tagging and lemmatization, which is even more important, in order to abstract away from the many inflected forms a lexical item can have in German. However, complement structure is not available, which would be very helpful to account for certain NPIs, e.g. *brauchen* (need), whose negative polarity depends on the complement structure (see section 3.4).

On the other hand, there are “items of such low frequency that statistical methods will have trouble identifying them”([2], chap. 2). Because of the rare occurrence of many NPIs we need large collections of language data and therefore we use an automatically annotated corpus such as TüPP-D/Z. Therefore we have to accept, that the corpus is partially parsed and the annotation has its errors (see 3.3).

Another problematic aspect is determination of licensing contexts for each lemma. As Hoeksema shows for hidden licensers (such as conditional readings) and the class of negative predicates (such as *lack* or *impossible*), it is virtually impossible to detect all licensers of negativity. On the other hand, the determination of licenser scope can be considerably improved by using clause information of the TüPP-D/Z corpus. But still it is just an approximation towards correct scope modelling.

3.2 Methods

We use a part of the TüPP-D/Z corpus (*Tübingen Partially Parsed Corpus of Written German*; see [10]).[3] TüPP-D/Z is based on the electronic version of the German newspaper *die tageszeitung* (*taz*). It contains lemmatization, part-of-speech tagging, chunking and clause boundaries. The year 1998 of TüPP-D/Z consists of about 1.2 Mio sentences (1,170,618) which contain 733,098 different lemmatized forms.

The NPI extraction procedure is basically done in three steps: clause marking, lemmata counting and evaluation.

Based on the lemmatization and the part-of-speech assignments in TüPP-D/Z we classify the clauses according to the presence of an NPI licenser. Basically, we demand the licenser to impose downward entailingness or to form an interrogative construction. Thus the set of NPI licensers comprises lexical licensers (e.g. *nicht* (not), *niemals* (never), *kaum* (hardly), question mark) and structural licensers (e.g. superlative + relative clause).[4] In future research we want to add licensers with inherent negation (e.g. *bezweifeln* (to doubt)).[5] Subclause annotation plays a crucial role for structural and inherently negative licensers since they only license NPIs within the clausal complement. Unfortunately we can only model a subset of

[3] The homepage of TüPP-D/Z is `http://www.sfs.uni-tuebingen.de/tupp`.

[4] A full list of the triggers which we used in our study is given in the appendix.

[5] A full list of the triggers which we used in our study is given in the appendix.

structural licensers where the clausal complement immediately follows the structural licenser and we can be sure that both belong together. This limitation follows from the absence of an annotated complement structure, as mentioned in 3.1 .

We also use clause-structure annotation given in TüPP-D/Z to derive scope relations in a very general manner. It holds that a deeper embedded negative operator cannot license NPIs in a higher position. An example of such a configuration is given in (5-a). In this structural position *nicht* cannot license an NPI in the matrix clause (b).

(5) a. [Was Frauen droht, [die dem Aufruf **nicht** folgen]], blieb unklar.
'It remained unclear [what was going to happen to women [who do **not** follow the call]].'

b. *[Was . . . [. . . **nicht** folgen]] wurde jemals gesagt.
what not follow was ever said

On the other hand, a licenser of a clause is also valid for all its sub-clauses.

After clause marking we extract for each lemma in the corpus the number of total occurrences and the number of occurrences in clauses which contain a trigger. We restrict ourselves (i) to lemmata which are not lexical triggers for any of the contexts themselves and (ii) to lemmata which occur at least 40 times, because less frequent lemmata do not show a reliable occurrence pattern for polarity contexts. We have to concede that this is a purely heuristic threshold. The resulting data contain 15,791 lemmata.[6]

In order to derive lists of NPI candidates, we calculate the ratio of contextual and total occurrence for each lemma. We then calculate the mean μ and the standard deviation sd of the context ratio over all lemmata. That allows us to determine the z-value for each lemma on the basis of the following formula using x as the lemma's relative frequency in a given context.

(6) $z = \frac{x-\mu}{sd}$

A ranking based on the z-values is equivalent to a ranking based on the context ratio of occurrence. Nonetheless, the z-values veer away from actual ratios and indicate the difference of a context ratio from the mean in terms of multiples of the standard deviation. Furthermore, statistical significance levels can be integrated, given a normal distribution. The distribution curve of the context ratios, however, is not perfectly normal, but misses extreme z-values on the left.[7]

[6] 279,022 lemmata occurred less than 40 times.

[7] The z-values range from -2.65 to 8.29 and the mean is zero.

3.3 Problems of Annotation

Using an annotated clause-structure for modelling scopal relations requires, of course, correctly annotated clauses. Unfortunately clause annotation is a difficult task, especially in the case of highly nested or complex sentences, where the shallow parser used in TüPP-D/Z often reaches its limit. One can find instances of wrongly annotated clauses in the corpus, even when they are clearly marked, e.g. by initial *dass* (that) such as in (7).

(7) <**cl**> Dem Papier ist zu entnehmen </**cl**>, **dass** gestern [...] 1,8 Milliarden Mark gewonnen werden konnten . [taz, 01.04.1998]
’It can be seen from the paper, that [...] 1.8 billion Marks were gained, yesterday.’

Looking at the XML-structure of the corpus, the ideal case is the sentence-node containing one single clause-child, which comprises the matrix-clause and possibly further sub-clause structures. But this does not always correspond to reality: there are sentence-nodes without an immediate clause-child, unrecognised sub-clauses, annotated matrix-clauses and concurrently unrecognised sub-clauses (as in (7)), and unrecognised matrix-clauses next to annotated sub-clauses — every configuration seems to be possible.

In order to avoid wrong data due to wrong clause annotation, we concentrate on sentences with one clause-child. Here the parser can interpret all elements of the sentence in a (for the parser) satisfying way, therefore it is more likely that the clause-structure is correct. Doing this, 32% of the 1170618 sentences of the 1998 TüPP-D/Z year are ignored.[8]

Regarding the lemma rankings that result from using either the reduced or the whole corpus, we cannot find major differences for lemmata with high z-values, though the lemma ranking from the reduced corpus seems to be better. A comparison in terms of descriptive statistics, however, is problematic. The frequencies of the lemmata differ considerably according to the difference in corpus size.

3.4 Results: NPI candidates

We expect NPIs to have particularly high z-values. 595 (297) lemmata show a significantly high z-value at $p < 0.05$ ($p < 0.01$). Immediately the question arises of how to decide the quality of the ranking.

Quality could be expressed in terms of precision and recall: We start from an NPI list established by the lists of [13] and [5] and ask how many NPIs are retrieved

[8] An improved query gives us even a rate of 38%.

at least partly in the candidate list, which could consist of lemmata above a certain z-value. Whether or not complete NPIs are detected is not important, as long as parts of known NPIs are found.

However, the NPI list of Kürschner and Welte is not complete. We expect many NPIs not to be on the list, and our goal is also to find these NPIs. With this approach precision and recall are less trustworthy: lemmata not on the NPI list could be parts of NPIs not yet detected.

Therefore we will use the following qualitative evaluation, which is more flexible and more precise: We will consider the ranking position of the prototypic NPIs *jemals* and *brauchen*, and we examine a group of top ranked lemmata.

If we then examine the 25 highest z-scored lemmata[9], we get the following picture: introspectively, the list does not contain totally non-polar items. However, for several reasons not every item is an NPI by itself.

Firstly, in addition to NPIs such as *sonderlich* (particular) and *gar* ([not] at all), which are complete and non-polysemous, there are lemmata which clearly show negative polarity without being complete NPIs, i.e. they rarely occur as NPIs without certain lexical material surrounding them. An example of an incomplete non-polysemous NPI is *wahrhaben* from the multiword expression *wahrhaben wollen* (to want to admit).

Secondly, one finds several "pseudo-polarity items"([2]), that have a stylistically motivated affinity for negation, but still can occur outside negative contexts. And even here one can distinguish between stand alone (*finanzierbar* ('affordable')) and lexically dependent (*hinwegtäuschen können* (to obscure the fact)). Since the text type of the corpus influenced our data, we expect better results from a more balanced corpus. Nevertheless, pseudo-polarity is interesting as an early form of polarity sensitivity.

The NPIs *jemals* (ever) and *brauchen* (need), which we used in the previous examples, are not included in these 25 top-ranked lemmata. While *jemals* occurs at rank 49, which is satisfying, we have to go down to rank 525 for *brauchen*. This divergency is probably caused by the polysemous character of *brauchen*, which only requires a negative context, if it has a non-finite clausal complement, as shown in (8):

(8) a. Peter braucht das **nicht** zu essen.
Peter needs that not to eat

b. Peter braucht einen Kaffee.
Peter needs a coffee

[9] see the appendix for a detailed listing.

The grade of polysemy culminates in complex NPIs, where the elements on their own look totally innocent (*alle Tassen im Schrank haben* ('to have lost ones marbles')). To handle complex NPIs in general we need an enhancement of the current method, as proposed in section 4.

3.5 Further Collocation Measures

We want to briefly compare the ranking of our z-values of context ratios with the NPI candidates resulting from two commonly used measures of collocation: G^2, a derivative of Log-likelihood, and *Pearson's* χ^2 *test*. See [9] and [8] for the formal definitions of these measures. Although *mutual information* (MI) is not explicitely included it can easily be shown that the ranking by context ratios is the same as the ranking by MI.[10]

We computed the G^2 score for each lemma. The top ranked lemmata cannot be judged as promising NPI candidates. Except for *sondern* and *gar* ([not] at all) we cannot find a considerable affinity with negation. In fact the remaining lemmata occur at a very high frequency in the corpus; they are function words such as determiners (*die*, *der* (the)), copula (*sein* (to be)), particles (*auch* (also)), modal verbs (*können* (can)) or conjunctions (*dass* (that)).[11] From this we deduce that the G^2 score overemphasizes the general frequency of lemmata. With our rather heterogeneous frequencies, this causes a biased view on independence. [8] also state the weakness of the log-likelihood score against high frequencies in general.

To obtain a balance we can divide the G^2-value of a lemma by its general occurrence and use this modified G^2 as a ranking criterion. In doing this we get another list of promising NPI candidates. Closer inspection reveals that the ranking, even beyond the 30. position, is congruent with that of the z-value list. *jemals* has exactly the same position (49) on both lists. In fact, there are only 24 elements within the first 200 that appear on one, but not on the other list. Nevertheless there is at least one major difference when it comes to the position of *brauchen*, namely 928 with G^2 compared to 525 with context ratio.

When generating a ranking by χ^2 we once again end up with a disappointing

[10] Given a lemma w with frequence N_w , the frequence of negative contexts N_{neg} and furthermore $N_{w,neg}$ as the frequence of w occuring in a negative context, the formal definitions of context ratio and MI will then appear as follows:

$$\text{(i)} \quad \begin{array}{rl} \text{context ratio :=} & \frac{N_{w,neg}}{N_w} \\ \text{MI :=} & \frac{P(w\&neg)}{P(w)*P(neg)} = \frac{N_{w,neg}/N}{(N_w/N)*(N_{neg}/N)} = \frac{N_{w,neg}}{N_w} * \frac{N}{N_{neg}} \end{array}$$

$P(w\&neg)$ is the probability of the co-occurrence of w and a negative context. It is obvious that $\frac{N}{N_{neg}}$ has a constant value and hence is not substantial for the computation of the ranking.

[11] *der* (the) is the most frequent lemma in our corpus (590,322 occurrences).

list of lemmata that obviously gain a prominent position due to their high overall frequency. However, if we balance the χ^2-value by total frequency, we also get a promising candidate list. The χ^2 list, unlike the G^2 list, has the same elements even within the 200 top ranks, as the list by context ratio, as well as a congruency in the first 30 positions of the ranking. Again, *brauchen* makes the difference with position 674 in the χ^2 list.

To summarize this section, our use of context ratios is justifiable since we only target the candidate list by ranking, without being interested in the actual strength of association. The resulting ranking seems to be just as good as that from G^2 and χ^2, and even better as far as the position of *brauchen* is concerned. Moreover, G^2 and χ^2 are considerably more complex to compute, which might become important with large data sources.

4 Towards Complex NPIs

We have presented a method for extracting a list of NPI candidates automatically from a corpus. However, these NPI candidates are only single lemmata and we already mentioned in section 3.4 that many NPIs are more complex than that: they might be *multiword expressions* or they might consist of polysemous NPIs which require the presence of certain lexical material to develop negative affinity. We therefore want to propose an enhancement of the original method, in order to account for complex NPIs.

The outline of the enhancement looks as the following recursive procedure: the starting point is the list of lemmata and their context ratios. We do a collocation test for every lemma and ask for significantly co-occurring lemmata. Afterwards we test the distribution of a lemma and each of its collocates with respect to negative contexts. If there is a distribution pattern of lemma and collocate, which shows higher affinity to contexts of licensers, we then repeat the procedure again on the lemma and collocate pair, which is now handled the way we handled single lemmata. In doing this we get chains of lemmata as new NPI candidates, which cannot be expanded because they lack either collocates or an enlarged affinity for negation.

The advantage of using the whole lemma list is that we have the chance to detect complex NPIs such as *alle Tassen im Schrank haben*, where the elements, taken individually, behave very inconspicuously with respect to negative contexts, therefore being ranked far away from the usual NPI suspects[12]. The disadvantage is rather technical, but nevertheless meaningful to us: it takes a lot of time.

[12] *alle* (all) is the best ranked at position 1009.

As a collocation measure we integrated the G^2 score. The span of collocation testing was a clause as annotated in TüPP-D/Z. Here the question arises of which significance level to choose, as even a "strong" significance level at $p < 0.01$ (6.6) seems to be too weak ([7]). We took a G^2 score of 10 for the examples below.

Because of the computational efforts we will present only two isolated cases. Consider the verb *verkneifen* (manage without) at ranking position 84. We find a collocate *können* (can), which co-occurs 33 out of 58 times. The combination *verkneifen können* appears 32 times within a negative context, which makes a z-value of 7.959 and corresponds to the second best ranking ! *verkneifen können* does not appear in [5] and seems to be a NPI to the authors. Another example concerns *Kram* (stuff) at ranking position 101. From that we can generate a lemmata chain *Kram in passen der,* which occurs 16 out of 19 times in negative contexts, giving it a z-value of 6.563 and the 7th position of the ranking. The lemmata chain obviously corresponds to the multiword expression *in den Kram passen* ('to be welcome', literally: to fit in the stuff), also an complex NPI to the authors and not listed in [5].

This illustrates how complex NPIs can be obtained, even ones that were unnoticed so far. It also becomes obvious that we have to enlarge the corpus since the frequencies of the presented complex NPIs are quite low.

5 Conclusion

Our starting point was the insight from [12] that the relation between an NPI and its licenser is of a collocational nature. We extracted distributional profiles of lemmata in a partially parsed corpus of German, mainly with the aid of lemmatization, part-of-speech tags and clause structure annotation and with respect to negative contexts derived from the semantic literature on NPIs. We used these profiles to compile a list of NPI candidates. We showed that a simple quantitative ranking leads to promising candidates, and we compared this with other collocational measures. We also showed that we can extend our method naturally to complex NPIs.

References

[1] Anastasia Giannakidou. *Polarity Sensitivity as Nonveridical Dependency*. John Benjamins, Amsterdam, 1998.

[2] Jack Hoeksema. Corpus study of negative polarity items. Html version of a paper which appeared in the *IV-V Jornades de corpus linguistics 1996-1997*, Universitat Pompeu Fabre, Barcelona. URL:

http://odur.let.rug.nl/∼hoeksema/docs/barcelona.html, 1997.

[3] Jacob Hoeksema. De negatief-polaire uitdrukkingen van het Nederlands. inleiding en lexicon. Manuskript, Rijksuniversiteit Groningen, November 2002.

[4] Edward Klima. Negation in English. In Jerry A. Fodor and Jerrold Katz, editors, *The Structure of Language*, pages 246–323. Prentice Hall, Englewood Cliffs, New Jersey, 1964.

[5] Wilfried Kürschner. *Studien zur Negation im Deutschen*. Gunter Narr, Tübingen, 1983.

[6] William Ladusaw. *Polarity Sensitivity as Inherent Scope relations*. Garland Press, New York, 1980.

[7] Lothar Lemnitzer. *Akquisition komplexer Lexeme aus Textkorpora*. Niemeyer, Tübingen, 1997.

[8] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[9] P. Rayson and R. Garside. Comparing corpora using frequency profiling. In *Proceedings of the* Workshop on Comparing Corpora*, ACL, 1–8 October 2000, Hong Kong*, pages 1–6, 2000.

[10] Tylman Ule and Frank Henrik Müller. KaRoPars: Ein System zur linguistischen Annotation großer Text-Korpora des Deutschen. In Alexander Mehler and Henning Lobin, editors, *Automatische Textanalyse. Systeme und Methoden zur Annotation und Analyse natürlichsprachlicher Texte*. Westdeutscher Verlag, Opladen, 2004. to appear.

[11] Ton van der Wouden. Beperkingen op het optreden van lexicale elementen. *De Nieuwe Taalgids*, 85(6):513–538, 1992.

[12] Ton van der Wouden. *Negative Contexts. Collocation, Polarity and Multiple Negation*. Routledge, London, 1997.

[13] Werner Welte. *Negationslinguistik. Ansätze zur Beschreibung und Erklärung von Aspekten der Negation im Englischen*. Wilhelm Fink Verlag, München, 1978.

[14] Frans Zwarts. Nonveridical contexts. *Linguistic Analysis*, 25:286–312, 1995.

[15] Frans Zwarts. Three types of polarity. In Fritz Hamm and Erhard W. Hinrichs, editors, *Plurality and Quantification*, pages 177–237. Kluwer Academic Publishers, Dordrecht, 1997.

A Appendix

A.1 Triggers for Negative Contexts

We will only list the lemmata which were used as triggers. We only give the base form in nominative. The lemmatization subsumes the inflected forms.

Lexical Triggers:
nicht, nich, keinesfalls, keineswegs, ohne (as complementizer), weder
nie, niemals, nimmer, nirgendwo, nirgends, nirgendwohin, nirgendwoher, niemand, nichts, nix, kein, keinerlei, ohne (as preposition), bevor, wenn
höchstens, kaum, selten, wenig
ob, wer, was, welch, welcher, welche, wann, warum, weshalb, weswegen, wie, wieso, wieviel, wo, wobei, wodurch, wofür, woher, wohin, woran, worauf, woraufhin, woraus, worein, worin, worüber, worum, worunter, wovon, wovor, wozu, wonach, womit, ?
Comment: The interrogative pronouns are extracted by their part of speech tag (PWAT, PWAV, PWS).

Structural triggers:
- Universal quantifiers (*alle*,*jede*) + immediately following relative clause
- NPs containing superlatives + immediately following relative clause
- *zu* (too) + adjective + immediately following clause, introduced by *um* (in order to)

A.2 NPI candidates

#	Lemma	z-val.
1	verhehlen (to conceal)	8.29
2	hinwegtäuschen (to obscure the fact)	7,85
3	sonderlich (particular)	7,62
4	notwendigerweise (necessarily)	7,10
5	verwunderlich (amazing)	7,00
6	zimperlich (prissy)	6,63
7	hinnehmbar (tolerable)	6,56
8	Seltenheit (sparseness)	6,51
9	anbelangen (to concern)	6,38
10	antun (to do sth. to so.)	6,35
11	gar ([not] at all)	6,30
12	beileibe (by [no] means)	6,25
13	nützen (to be of use)	6,17
14	antasten (to touch)	6,15
15	Handhabe (handle)	6,15
16	scheren (to pay attention)	6,14
17	einwenden (to argue)	6,08
18	vorhaben (to intend)	6,06
19	durchsetzbar (enforceable)	6,03
20	abschätzen (to estimate)	5,96
21	verborgen (hidden)	5,80
22	Ahnung (anticipation)	5,79
23	wahrhaben (to disavow)	5,78
24	zurückschrecken (to stop at)	5,70
25	unsympathisch (unappealing)	5,69

An Empirical Investigation of the Effect of Lexical Rules on Parsing with a Treebank Grammar

Hiroko Nakanishi†, Yusuke Miyao†, Jun'ichi Tsujii†‡

University of Tokyo†
CREST, JST‡
E-mail: {n165, yusuke, tsujii}@is.s.u-tokyo.ac.jp

1 Introduction

Several approaches to grammar acquisition from treebanks have succeeded in obtaining lexicalized grammars with many lexical entries at low cost (Xia [1], Chen and Vijay-Shanker [2], Chiang [3], Hockenmaier and Steedman [4], Miyao et al. [5]). Although treebank grammars achieved significantly higher coverage in real-world texts than manually developed grammars, coverage still needs to be improved. Lexicalized grammars assign many lexical entries to a single word to deal with the various syntactic alternations. Since it is difficult to obtain all words in all syntactic alternations in the treebank, we need to modify the acquired lexicon to improve coverage.

We previously succeeded in improving the coverage of such grammars by using lexical rules to expand lexical entries acquired from treebanks [6]. The lexical entries are first reduced to their lexemes using the inverse lexical rules. After that, the lexemes are expanded to various kinds of lexical entries using the ordinary lexical rules. However, our method has not been evaluated in terms of statistical parsing. Considering that the number of lexical entries assigned to a word is increased significantly, the method possibly has negative side effects on parsing. Intuitively, the increase in lexical ambiguity makes disambiguation difficult. We can therefore expect that the accuracy of parsing decreases and the parsing time increases.

We examined the effect of using lexical rules to acquire an HPSG grammar from a treebank, in terms of accuracy and parsing time. Our experiments were based on an HPSG grammar extracted from the Penn Treebank (Marcus et al. [7]) and a probabilistic CKY-style parser [8] described in the following sections. Contrary to our expectation, the results showed recall significantly increased without

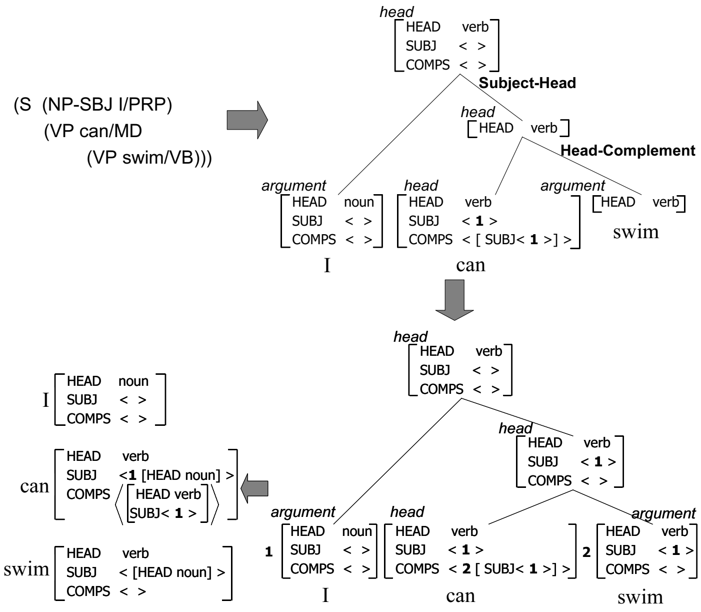


Figure 1: Grammar extraction from the Penn Treebank

decreasing precision. This indicates that lexicon expansion improves not only coverage but also the accuracy of statistical parsing when we use an appropriate probabilistic model and parsing strategy. The results also showed that parsing time is not affected by the lexicon expansion.

2 Background

2.1 Extracting HPSG from a treebank

We used an HPSG grammar extracted from the Penn Treebank using the method of Miyao et al. [5] as the original grammar. The original grammar consists of a few grammar rules (schemata) written by hand and many lexical entries acquired from the treebank. Figure 1 shows the grammar extraction process. A parse tree in the treebank is first converted into an HPSG-style derivation tree by enhancing the annotations in the treebank. For example, an auxiliary verb shares its subject with the main verb in HPSG, though it is not explicitly annotated in the Penn Treebank (Figure 1). In this case, an extra annotation is added so that “can” shares its subject with its complement. Lexical entries are then acquired by applying the schemata

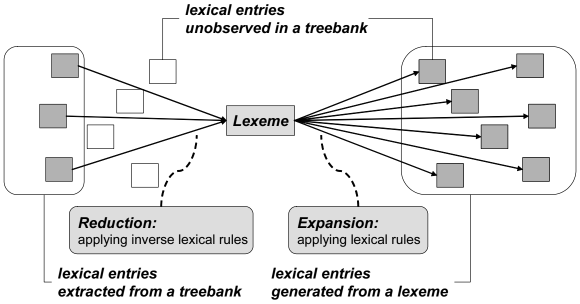


Figure 2: Lexicon expansion with lexical rules

inversely to the derivation tree.

Since each lexical entry represents detailed syntactic constraints, many lexical entries are assigned to one word in lexicalized grammars. For example, the lexical entries for "gave" in the following sentences are different in HPSG.

- He *gave* a book to Mary.
- He *gave* Mary a book.
- This is the book he *gave* to Mary.

Therefore, many lexical entries are needed to analyze real-world texts with lexicalized grammars.

In grammar acquisition, an acquired lexical entry corresponds to only a single syntactic alternation of a word. Even if the lexical entry for a verb in the present tense is acquired, lexical entries for the past tense, the base form, or other syntactic alternations are not acquired using the conventional method. As a result, a word is regarded as unknown when its surface form appears in a different syntactic alternation, limiting the coverage of the acquired treebank grammar.

2.2 Expanding a treebank grammar with lexical rules

Pollard and Sag [9] proposed that each lexical entry be generated by applying lexical rules to a *lexeme*. For example, *past_rule* generates a lexical entry for the past verb "had" from the lexeme for "have". Since lexical rules are common to all lexemes that satisfy the input conditions (e.g. the input of *past_rule* must be a verb in the base form), a small set of lexical rules are sufficient to generate various kinds of lexical entries.

We previously showed that lexical rules are useful for improving the coverage of a treebank grammar [6]. The process of lexicon expansion using lexical rules in grammar acquisition is divided into two steps: reduction and expansion (See Figure 2). The reduction phase reduces lexical entries having the same subcategorization frame into one lexeme by applying *inverse lexical rules*. For example, “gave” in “He gave his book to Mary” or “given” in “the book given to Mary” is reduced to one lexeme corresponding to the ditransitive verb “give”. Since one lexical entry corresponds to one lexeme, the reduction phase is deterministic. Once the lexemes are obtained, lexical entries unobserved in the treebank can be generated by simply applying lexical rules to the lexemes in the expansion phase.

We developed a set of lexical rules manually using one section in the Penn Treebank as a development set. While the lexical entries were designed to produce only linguistically valid lexical entries, the expanded lexicon includes lexical entries rarely observed in real texts. Such infrequent lexical entires may lead to superfluous ambiguities, possibly decreasing parsing accuracy. We will discuss this problem in Section 3.2.

The effect of lexical rules was evaluated by comparing the coverage of the three grammars.

- G_0 original grammar (without reduction or expansion)
- G_1 grammar with expansion
- G_2 grammar with reduction and expansion

The results showed that G_2 achieved a higher coverage than G_0 and G_1. Alternatively, the number of lexical entries assigned to one word significantly increased, indicating increase in lexical ambiguity. This is discussed in detail elsewhere [6].

3 Parsing with the grammar expanded by lexical rules

The results for coverage are insufficient because we intend to use the acquired grammar in statistical parsing. As mentioned before, the number of lexical entries assigned to a word significantly increases when lexical rules are used. The increase leads to more ambiguity than when using G_0.

We should consider two possible side effects here. First, parsing accuracy decreases because disambiguation becomes more difficult due to increase in lexical ambiguity. The expanded lexicon has several infrequent lexical entries, as mentioned in Section 2.2. If such lexical entries are included in the output parse trees, accuracy possibly decreases. Second, parsing time increases because the number of edges increases.

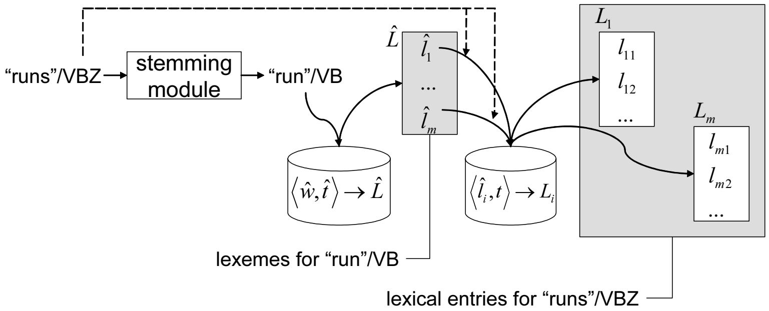


Figure 3: Looking up a word in the expanded lexicon

While these side effects are crucial despite the high coverage, parsing accuracy and time have not ye been investigated. To determine if the side effects actually occur, we evaluated changes in parsing accuracy and time among the three grammars. The HPSG parser we used was a probabilistic CKY-style parser with beam search for unification grammars developed by Tsuruoka et al. [8]. The following sections describe refinement of parsing with the grammar using lexical rules.

3.1 Offline expansion

Since the expansion phase makes looking up a word in the lexicon slower, we process this phase offline to save parsing time. We built two mappings. One is $\langle \hat{w}, \hat{t} \rangle \rightarrow \hat{L}$ mapping, where $\hat{w}$ is the base form of a word, $\hat{t}$ is the part-of-speech (POS) of $\hat{w}$, and $\hat{L}$ is the set of lexemes. The other is $\langle \hat{l}, t \rangle \rightarrow L$ mapping, where $\hat{l}$ is the lexeme, t is the POS of an inflected word, and L is the set of lexical entries. The first mapping is obtained through grammar acquisition and reduction with inverse lexical rules.

The second mapping is obtained offline. Since the basic idea of lexical rules is that lexical rules are not specific to particular words, we can build this mapping without considering words. We assume that the output POS of a lexical rule is determined when the input POS is given. Conversely, a lexical rule and its output lexical entry templates are determined given a lexeme and an output POS. As it is the same if plural lexical rules are applied to a lexeme in order, we applied as many rules as possible to $\hat{l}$, where the output POS is t, and registered the generated lexical entries with the key $\langle \hat{l}, t \rangle$.

When looking up a word in the expanded lexicon, the two mappings are used in order, as shown in Figure 3. To look up lexical entries for "runs"/VBZ, first the base form of the word and the corresponding POS, "run"/VB, is obtained by an

Table 1: Lexical entry templates assigned to "calculates"/VBZ

	l_1 (noun complement)	l_2 (wh-clause complement)
occurrence in treebank	0	1
probability	0.163	0.021
features (weights)	$\langle$VBZ, $l_1\rangle$(679)	$\langle$VBZ, $l_2\rangle$(85.8) $\langle$"calculates", VBZ, $l_2\rangle$(1.00)

external stemming module. Second, the first mapping is checked for $\langle$"run",VB$\rangle$ to obtain a set of lexemes $\hat{L}_{\langle''\mathrm{run}'',\mathrm{VB}\rangle}$. Thirdly, the second mapping is consulted with $\langle\hat{l}_i$,VBZ$\rangle$ for each $\hat{l}_i$ in $\hat{L}_{\langle''\mathrm{run}'',\mathrm{VB}\rangle}$ to obtain expanded lexical entries L_i. The lexical entries for "runs"/VBZ are obtained as $\bigcup_{\hat{l}_i\in\hat{L}} L_i$.

3.2 Probability estimation for unobserved lexical entries

To parse texts with G_1 or G_2, we have to estimate the probability distribution for lexical entries unobserved in the treebank. A study (Miyao and Tsujii [10]) using a probabilistic model on HPSG assigned a conditional probability $P(l|w,t)$ to each word, where l is the lexical entry, w is the word, and t is the POS. $P(l|w,t)$ was calculated using the maximum entropy (ME) estimation (Berger et al. [11]). The set of ME features consisted of $\langle$surface form, POS, lexical entry template[1]$\rangle$ tuples and $\langle$POS, lexical entry template$\rangle$ pairs. We can simply adopt this model for G_1 and G_2 because reasonable probabilities are assigned to unobserved lexical entries due to the smoothing effect of ME estimation.

$P(l|w,t)$ can be high even if l is not associated with w in the treebank. Table 1 shows two of the lexical entry templates assigned to "calculates"/VBZ and their related features. Both templates have one subject and one complement, while l_1 has a noun complement and l_2 has a wh-clause complement[2]. Although l_1 is not associated with "calculates" in the treebank, it has a higher probability than l_2 because it is frequently associated with other verbs, so that feature $\langle$VBZ, $l_1\rangle$ is heavily weighted.

In contrast, an unobserved lexical entry has quite a low probability when its lexical entry template is not observed in the treebank. Therefore, even if lexical

[1] A lexical entry template is a lexical entry whose word-specific features (e.g. PHON) are abstracted.

[2] The sentence where l_2 appeared is "And it *calculates* how often the words appear in the story compared with how often they appear in the entire data base".

Table 2: Size of grammars

	G_0	G_1	G_2
No. of words	11797	11797	9312
No. of lexical entry templates	1146	2362	2356
No. of ME features	72645	72645	72658

Table 3: Coverage for test set

	G_0 (%)	G_1 (%)	G_2 (%)
lexical coverage (all, 51003 words)	98.31	98.59	98.67
sentential coverage (all, 2329 sentences)	73.12	76.86	78.40
sentential coverage ($<$ 40 words, 2186 sentences)	74.75	78.32	79.73
No. of lexical entries assigned to one word (all)	26.32	36.91	36.90

rules generate infrequent lexical entries, as mentioned in Section 2.2, they are assigned low probabilities and considered to have little effect on statistical parsing.

4 Experiments

We compared the three types of grammars mentioned in Section 2.2. Tables 2 and 3 show the size and coverage of the grammars[3]. All the grammars were acquired from the Penn Treebank, Section 02-21. Section 22 was used for developing the lexical rules, and Section 23 was used as the test set. G_1 and G_2 had about twice as many lexical entry templates as G_0 because new templates were generated using the lexical rules (See Table 2). G_2 had fewer words than the other grammars because inflected words were reduced to their base form in G_2 using the inverse lexical rules. In G_2, inflected words were looked up with the base forms in the lexicon. The model size (No. of ME features) differed little among the grammars. The lexical coverage was measured by examining if the acquired lexicon had exactly the same lexical entries as those extracted from the test set using the method of Miyao et al. [5]. The sentential coverage was measured by examining if the lex-

[3] The results are different from the ones in our previous work because we added some lexical rules. Since the emphasis of this paper is on empirical evaluation, we omit detailed description of the rules and generated lexical entries.

Table 4: Parsing results for the test set ($<$ 40 words)

	G_0 (%)	G_2 (%)	difference (%) (p-value)
LP/LR	84.08 / 81.51	84.25 / 83.42	0.17 / 1.91 (1.4e-1 / 1.0e-4)
UP/UR	86.88 / 84.23	87.01 / 86.16	0.13 / 1.93 (1.6e-1 / 1.0e-4)
parsing time (ms/sentence)	1398	1259	-

	G_1 (%)	G_2 (%)	difference (%) (p-value)
LP/LR	84.06 / 81.76	84.25 / 83.42	0.19 / 1.66 (8.2e-2 / 1.0e-4)
UP/UR	86.95 / 84.57	87.01 / 86.16	0.06 / 1.59 (2.8e-1 / 1.0e-4)
parsing time (ms/sentence)	1483	1259	-

icon had all lexical entries extracted from a sentence in the test set. As previously described [6], G_2 had higher sentential coverage than G_0 and G_1.

Table 4 shows the labeled precision/recall (LP/LR) and the unlabeled precision/recall (UP/UR) for the sentences within 40 words in the test set. We compared G_2 with G_0, then G_2 with G_1 separately. Here precision/recall represents accuracy of the predicate-argument relations between words. “Unlabeled” means that predicate and argument words were correctly related, while “labeled” means that the position of the argument (e.g. the logical subject should be ARG1) was also correct. This measure is similar to that reported elsewhere (Hockenmaier [12], Miyao and Tsujii [10]). We conducted *stratified shuffling tests* (Cohen [13]) to determine if the differences between the results were statistically significant in Table 4. The rightmost column represents the differences and their p-values.

In the stratified shuffling test, the null hypothesis is that two models are the same. If the two models are the same, a difference between the results should be observed even when the samples are randomly shuffled between models. Hence, the null hypothesis is tested by performing n shuffles and counting the number n_g when the difference between the shuffled results is greater than the original difference. The likelihood that the null hypothesis is correct (p-value) is computed as $(n_g + 1)/(n + 1)$. In this case, we performed $10,000$ shuffles through each test.

G_2 achieved the best performance in all the measures as shown in Table 4. Although precision was expected to decrease, there was no significant difference between the precision of G_0 and G_2. Additionally, the increase in recall was signif-

Table 5: Parsing results for newly covered sentences

	G_0 (%)	G_2 (%)	difference (%) (p-value)
	originally covered sentences		
No. of sentences	1634		
LP/LR	87.90 / 87.95	87.62 / 87.28	-0.28 / -0.67 (2.7e-3 / 8.9e-3)
UP/UR	89.62 / 89.67	89.42 / 89.08	-0.20 / -0.59 (2.9e-2 / 7.8e-3)
	newly covered sentences		
No. of sentences	127		
LP/LR	75.55 / 64.62	82.51 / 82.51	6.96 / 17.89 (1.0e-4 / 1.0e-4)
UP/UR	81.08 / 69.35	85.90 / 85.90	4.82 / 16.55 (1.0e-4 / 1.0e-4)

	G_1 (%)	G_2 (%)	difference (%) (p-value)
	originally covered sentences		
No. of sentences	1712		
LP/LR	87.20 / 86.56	87.14 / 87.35	-0.64 / 0.79 (3.1e-1 / 1.6e-3)
UP/UR	89.17 / 88.52	89.08 / 89.30	-0.09 / 0.78 (1.7e-1 / 1.8e-3)
	newly covered sentences		
No. of sentences	41		
LP/LR	76.50 / 73.72	85.96 / 85.68	9.46 / 11.96 (1.0e-4 / 1.0e-4)
UP/UR	82.04 / 79.06	88.75 / 88.46	6.73 / 9.40 (3.0e-4 / 3.0e-4)

icant, showing $n_g = 0$. These results indicate the positive effect of lexical rules on parsing accuracy. Considering that G_1 is a grammar with lexical rules and without inverse lexical rules, we can examine the effect of inverse lexical rules by comparing G_1 with G_2. If there is no significant difference between G_1 and G_2, ordinary lexical rules are sufficient to obtain high accuracy. However, the difference between G_1 and G_2 was also significant in recall ($n_g = 0$ again), which indicates the positive effect of inverse lexical rules. The difference in precision was not significant as with G_0 and G_2.

In addition, the parsing time did not increase even when additional lexical entries were assigned to one word. This is probably because the HPSG parser used conducted a beam search. Since the edges with low probability are pruned in an early step of the beam search, the number of lexical entries has little effect on the parsing time.

We conducted another experiment to see how G_2 achieved high recall. To compare the results of G_0 and G_2, we extracted the following two types of sentences and evaluated the parsing accuracy for each type.

- originally covered sentences: sentences covered by G_0
- newly covered sentences: sentences uncovered by G_0 and covered by G_2

The same experiment was done for G_1 and G_2. Table 5 shows the results and p-values for the differences between the two grammars. G_0 was higher than G_2 in the originally covered sentences with significant differences. Although G_2 had better recall than G_1, there was no significant difference between the precision of G_2 and G_1. However, G_2 clearly outperformed the other grammar for the newly covered sentences. Contrary to the results in Table 4, both precision and recall showed significant increases for these sentences.

We can determine the reason for the increases by focusing on the "G_0" column. The accuracy for the covered sentences exceeded that for the uncovered sentences. This is because the lexicon did not have correct lexical entries for the uncovered sentences, so that the correct predicate-argument relations were difficult to output. Recall decreased when no parse tree was output due to the lack of lexical entries. Precision decreased when a parse tree was built using an incorrect lexical entry, which degrades the surrounding constituents. Hence, it is reasonable that G_2 achieved higher accuracy for the newly covered sentences. G_2 had higher overall accuracy than the other grammars since its advantage for the newly covered sentences was greater than its disadvantage for the originally covered sentences.

5 Related work

Chen [14] extracted an LTAG grammar from the Penn Treebank and improved the coverage of the grammar by mapping *tree families*, a subcategorization frame in a hand-crafted English grammar XTAG (XTAG Research Group [15]), onto the extracted treebank grammar. In XTAG, *elementary trees*, which correspond to lexical entries in HPSG, are classified into tree families by hand. If elementary tree t is assigned to word w in the extracted grammar, all elementary trees in T, the tree family to which t belongs, are assigned to w. Chen improved the coverage of the grammar using this method, and also evaluated the accuracy of supertagging. To assign non-zero probabilities to elementary trees added by tree families, several smoothing methods were proposed in Chen's work. Based on the experimental results, smoothing using distributional similarity was better than other methods using tree families or POS.

There are two major differences between Chen's methods and ours. The first and most crucial difference is that our method does not require a hand-crafted grammar like XTAG. We cannot necessarily assume the existence of huge resources, and available resources are not always consistent with our target grammar. Since we assume that lexical rules are independent of grammar formalisms, we can apply our method to other lexicalized grammars without extra resources.

The second difference is that we can use the same probabilistic model whether we use lexical rules or not. The smoothing methods correspond to our strategy for probability estimation described in Section 3.2. Since the ME models we used worked well for assigning non-zero probabilities to unobserved lexical entries, our probabilistic model did not require refinement. Note that our model originally included smoothing by POS, but this was found to work worse than the other smoothing methods in Chen's work. This indicates that our model has the possibility for improvement by using other smoothing techniques, which can be our future work.

6 Conclusion

We examined the effectiveness of incorporating lexical rules into corpus-oriented grammar development for the first time. Although this could have had a negative effect on parsing in terms of accuracy and time, experimental results showed that higher recall was achieved without losing precision and that parsing time did not increase. These results are strongly dependent on the probabilistic model and the parsing strategy. Since both are compatible with lexicalized grammars other than HPSG, applying our method to other grammars such as LTAG is the basis of our future work.

References

[1] Fei Xia. Extracting tree adjoining grammars from bracketed corpora. In *Proceedings of the fifth NLPRS*, 1999.

[2] John Chen and K. Vijay-Shanker. Automated extraction of TAGs from the Penn Treebank. In *Proceedings of the sixth IWPT*, 2000.

[3] David Chiang. Statistical parsing with an automatically-extracted tree adjoining grammar. In *Proceedings of the 38th ACL*, pp. 456–463, 2000.

[4] Julia Hockenmaier and Mark Steedman. Acquiring compact lexicalized grammars from a cleaner treebank. In *Proceedings of the third LREC*, 2002.

[5] Yusuke Miyao, Takashi Ninomiya, and Jun'ichi Tsujii. Corpus-oriented grammar development for acquiring a Head-Driven Phrase Structure Grammar from the Penn Treebank. In *Proceedings of the first IJCNLP*, 2004.

[6] Hiroko Nakanishi, Yusuke Miyao, and Jun'ichi Tsujii. Using inverse lexical rules to acquire a wide-coverage lexicalized grammar. In *Proceedings of the first IJCNLP Workshop on Beyond Shallow Analyses*, 2004.

[7] Mitchell P. Marcus, Beatric Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 1993.

[8] Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Tsujii. Towards efficient probabilistic HPSG parsing: integrating semantic and syntactic preference to guide the parsing. In *Proceedings of the first IJCNLP Workshop on Beyond Shallow Analyses*, 2004.

[9] C. Pollard and Ivan A. Sag. Head-driven Phrase Structure Grammar, 1994.

[10] Yusuke Miyao and Jun'ichi Tsujii. Probabilistic modeling of argument structures including non-local dependencies. In *Proceedings of RANLP 2003*, pp. 285–291, 2003.

[11] Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, Vol. 22, No. 1, pp. 39–71, 1996.

[12] Julia Hockenmaier. Parsing with generative models of predicate-argument structure. In *Proceedings of the 41st ACL*, pp. 359–366, 2003.

[13] Paul R. Cohen. *Empirical Methods for Artificial Intelligence*. MIT Press, 1995.

[14] John Chen. *Towards Efficient Statistical Parsing Using Lexicalized Grammatical Information*. PhD thesis, University of Delaware, 2001.

[15] The XTAG Research Group. A Lexicalized Tree Adjoining Grammar for English. Technical report, IRCS 95-03, 1995.

A Large-Scale Japanese CFG Derived from a Syntactically Annotated Corpus and Its Evaluation

Tomoya Noro   Taiichi Hashimoto
Takenobu Tokunaga   Hozumi Tanaka

Tokyo Institute of Technology
Graduate School of Information Science and Engineering
{noro,taiichi,take,tanaka}@cl.cs.titech.ac.jp

1 Introduction

Although large-scale grammars are prerequisite for parsing a great variety of sentences, it is difficult to build such grammars by hand. Yet, it is possible to build a context-free grammar (CFG) by deriving it from a syntactically annotated corpus. Many such corpora have been built recently to obtain statistical information concerning corpus-based NLP technologies. For English, it is well known that a CFG derived from the Penn Treebank corpus (tree-bank grammar) can parse sentences with high accuracy and coverage although the method for deriving a CFG is very simple [1]. Actually, there have been quite a few studies concerning this kind of grammars. For Japanese, however, CFGs cannot be derived using the Charniak's method since there is no large-scale syntactically annotated corpus such as the Penn Treebank corpus [1]. Therefore such corpus needs to be developed to enable derivation of a large-scale CFG.

However, even if a large-scale, syntactically annotated corpus were already available, a CFG derived from it can be unsatisfactory, in as it creates a great number of possible parses (in average more than 10^{12}, according to our preliminary experiment). Too many parse results do not only reduce the parsing accuracy and parsing speed, but also require larger memory to parse and store long sentences. Although Charniak has removed some CFG rules (e.g. rules occurring only once

[1] The EDR Japanese corpus [4] is one of the large-scale Japanese corpora. However, unlike the Penn Treebank corpus, it is a bracketed corpus (nonterminal symbols are not assigned to each intermediate node). Shirai et al. proposed a method to derive a CFG from the EDR corpus, guessing nonterminal symbols to be assigned automatically to each intermediate node using some heuristics [17].

in the corpus) to avoid such problems, this is not enough, as the rules that occur more than once may also increase ambiguity.

Since the sentences of a normal, syntactically annotated corpus have “semantically correct” structure, the derived grammar creates many parse results, representing a different possible reading, i.e. meaning. A syntactic parser does not deal with semantics. Hence, it is difficult to deal with ambiguity of that sort. On the other hand, if the parser creates many different parses, it becomes difficult to disambiguate the results, even if semantic analysis is carried out after the syntactic parsing. We assume that syntactic analysis based on a large-scale CFG is followed by semantic analysis. Since the parse results are sent to the subsequent semantic processing, the number of parse results should be as small as possible. Therefore, it is necessary to build a CFG that minimizes the ambiguity during the syntactic parsing.

We attempt to build such a CFG from a syntactically annotated corpus, by using the following method: (1) derive a CFG from an existing syntactically annotated corpus, (2) analyze causes of ambiguity, (3) create a policy for modifying the corpus, (4) modify the corpus according to the policy and derive again a CFG from it, (5) repeat steps (2), (3) and (4) until all problems are solved. While repeating the steps (2) - (4) is labor-intensive and time-consuming, it is very important to do so in order to build an adequate, large-scale CFG for syntactic parsing.

In this paper, we propose a method for building such a large-scale Japanese CFG, under the assumption that the parse results will be subsequently sent to the semantic processing module. We also provide an experimental evaluation of the obtained CFG showing reduction in the number of parse results (reduced ambiguity) created by the CFG and the improved parsing accuracy. Several methods for tree transformation have been proposed for other languages [10, 16]. Although our work is similar, the difference is that we consider parsing ambiguity as well as parsing accuracy. Note that the CFG described in this paper does not perform any semantic analysis, it deals with syntax only. While our syntactic structures might look a bit odd from a semantic point of view, they are useful for keeping ambiguity low during syntactic parsing.

2 Causes of Ambiguity

To decrease the ambiguity (i.e the number of parse results), we start by analyzing main causes. There are four main causes of ambiguity:

Human Errors: Human annotators sometimes make mistakes when annotating syntactic structure of a sentence. If there are mistakes in the corpus, the derived CFG is likely to produce an incorrect structure.

Inconsistency: There may be contradiction concerning the structure since large-scale corpora are usually built incrementally and by several annotators. A CFG derived from an inconsistent corpus can yield many parse results with inconsistent structures.

Lack of Syntactic Information: Some important syntactic information might be lost during the CFG derivation since CFG rules generally represent only structures of subtrees of depth one (relation between a parent node and some child node). Yet, in case of Japanese, a verb phrase can be an adnominal phrase, continuous clause, or subordinate clause. In order to decide which one to choose, one has to consider verb conjugation or particles (postpositions) at the end of the phrase. In a sentence like "*boushi wo kabutteiru hito wo mita* (I saw the person wearing a hat)", the verb phrase "*boushi wo kabutteiru* (wearing a hat)" could be an adnominal phrase, because the conjugation of the verb "*kabutteiru* (wear)" is an adnominal form. If no information concerning verb conjugation can be assigned at intermediate nodes of the subtree covering the verb phrase, it is not clear whether the verb phrase is an adnominal phrase or continuous clause.

Need for Semantic Information: Semantic information is necessary for disambiguation in some cases (e.g. PP attachment problem for English). In the case of a phrase like "*kare no me no iro*", one cannot decide whether the adnominal phrase "*kare no* (his)" should be attached to the noun "*me* (eyes)" (the phrase meaning "color of his eyes"), or to the noun "*iro* (color)" (the phrase meaning "his color of eyes") by relying solely on syntactic information.

Since the first and second causes are types of annotation errors, they need to be corrected manually as soon as they are found[2]. On the other hand, since the third and fourth causes are not errors, they can be handled by modifying the structures in the syntactically annotated corpus and by deriving the CFG from this newly-annotated corpus.

3 Policy for Modifying the Corpus and the CFG

In order to avoid the third cause of ambiguity, information should be added to each intermediate node in the structure, where necessary. On the other hand, some am-

[2] Although this kind of error can be automatically corrected (or detected) in some methods [2, 3], not all of them can be corrected. They should be manually corrected at the end. Furthermore, Japanese has another problem that English does not have: there are potential errors in word segmentation since words are not separated by spaces.

biguity due to the fourth cause should be left to the subsequent semantic processing since it is difficult to reduce the ambiguity without recourse to semantic information during syntactic parsing. This can be achieved by representing the ambiguous cases as the same structure.

We have considered modification for verb conjugation, compound noun structure, adverbial and adnominal phrase attachment and conjunctive structure. In this section, we describe their modification briefly. The details are given in [14].

3.1 Verb Conjugation

As mentioned in the previous section, information of verb conjugation or particles (postpositions) at the end of the verb phrase is important to judge whether the phrase should be adverbial phrase or adnominal phrase. We add the information to each intermediate node related to the verb (cf. "SPLIT-VP" in [10] and "Verb Form" in [16]).

3.2 Compound Noun Structure

In general, it is difficult to disambiguate structure of compound noun without any semantic information. Shirai et al. modify their CFG to produce a right linear binary branching tree for compound nouns during the parse [17][3]. We modify the structure in the same way: structure ambiguity of compound noun is represented as the same structure regardless of the meaning or word-formation.

3.3 Adverbial and Adnominal Phrase Attachment

Semantic information is necessary to disambiguate adverbial and adnominal phrase attachment. However, it is meaningless to represent all of the ambiguity as the same structure regardless of the meaning, since it means no decision about phrase attachment is made during syntactic parsing and it makes the subsequent semantic processing difficult. Some of the ambiguity should be represented as the different structure (i.e. the ambiguity is unresolved during syntactic parsing). We represent structure ambiguity of adnominal phrase attachment as the same structure regardless of the meaning while we distinguish structure ambiguity of adverbial phrase attachment by meaning. In case of a phrase like "*watashi no chichi no hon* (my father's book)", the structure is same whether the adnominal phrase "*watashi no*

[3] Instead of the term "compound noun", Shirai et al. use the term "compound word", meaning by that term any constituent covering an identical part-of-speech (POS) sequence (e.g. a noun sequence). Our term "compound noun" refers to the fact that the constituent under study acts as a noun and consists of nouns, suffixes, prefixes, etc. (there is no need for an identical POS sequence.)

(my)" attaches to the noun "*chichi* (father)" or the noun "*hon* (book)". On the other hand, in case of a sentence like "*kare ga umi wo egaita e wo katta*", we distinguish the structure according to whether the adverbial phrase "*kare ga* (he)" attaches to the verb "*egaita* (paint)" (it means "I bought a picture of a sea painted by him") or the verb "*katta* (buy)" (it means "he bought a picture of a sea").

Since we believe that a different algorithm should be used to disambiguate adverbial phrase attachment and adnominal phrase attachment in Japanese, we have decided to deal with them separately. This means that the ambiguity concerning whether a phrase is an adverbial phrase or adnominal phrase remains during syntactic parsing. However, this increase of ambiguity is not very big. Actually, in Japanese it is relatively easy to discriminate between an adverbial and adnominal phrase [4]. We have also decided to annotate a corpus as described above since adverbial phrase attachment can be disambiguated in some cases using syntactic information (e.g. particles, punctuation).

3.4 Conjunctive Structure

In general, parsing accuracy of the sentences containing conjunctive structures is significantly worse than that of sentences without such structures. Our preliminary experiments show that the sentence accuracy of such sentences is only about half of the rest [5]. Coping with conjunctive structures is important for improving overall accuracy.

Since semantic information is necessary for analysis of conjunctive structures, it is difficult to disambiguate these structures in syntactic parsing. Kurohashi et al. propose a method that first detects conjunctive structures in a sentence, then analyzes the dependency structure of the sentence in order to disambiguate them [9]. Contrary to their method, our CFG does not specify conjunctive structures during syntactic parsing, as they are assumed to be analyzed during the subsequent semantic processing (similar to "Coordinated Categories" in [16]).

4 Evaluation

To evaluate the efficiency of the CFG modified according to our policy, we consider two aspects, both of which are important: the number of parse results created by the derived CFG, and the accuracy of the parsing achieved when using the CFG.

[4] There are cases where this discrimination is not so easy. For instance, the adverb "*hobo*" can be an adverbial phrase in the case of a sentence like "*hobo owatta* (it has almost been finished)" while it can be an adnominal phrase in the case of a sentence like "*hobo zen'in ga kita* (nearly everyone has come)", however, these cases are quite limited in number.

[5] The definition of sentence accuracy is described later.

As mentioned earlier, it is important to decrease the number of parse results, as this speeds up the processing while reducing memory load. It goes without saying that it is more important to increase the accuracy of the parsing rather than to speed up the process. We evaluated on the EDR Japanese corpus [4] (we refer to this corpus as "EDR corpus") and the RWC corpus [5].

4.1 Evaluation on the EDR corpus

The EDR corpus is a bracketed corpus with only skeletal structures recorded for each sentences. The intermediate nodes of the structure are not assigned with non-terminal symbols. We extracted 8,911 sentences (on average 20.01 morphemes in a sentence) from it and manually annotated "semantically correct" structure of each sentence (we refer to this corpus as "EDR original corpus"). Then we modified the structure according to the policy described above by an annotation tool [15] (we refer to this corpus as "EDR modified corpus").

We followed bracket structure in the EDR corpus to annotate "EDR original corpus". Since POS system of the EDR corpus is so coarse (only 15 POS tags), we assigned fine-grained POS tags from the EDR Japanese word dictionary. Each word in this dictionary has left and right adjacency attribute (i.e. information about what kind of POS tag can precede or follow the word) and surface case information for verbs and adjectives (i.e. information about what kind of case the verb or adjective takes) [6] as well as POS tag. We combined them and used as the POS tag set for the both EDR original corpus and EDR modified corpus.

CFGs are derived from the original and modified corpus (we refer to these two CFGs as "EDR original CFG" and "EDR modified CFG" respectively), and used to parse POS sequences of sentences in the corpus by MSLR parser [18] [7]. The number of rules in two CFGs and the number of parse results are shown in Table 1. The number of parse results decreased by 10^7 order, while the number of CFG rules increased by 255 [8].

Next, we ranked parse results by training the parser according to the probabilistic generalized LR (PGLR) model [6] using 10-fold cross-validation (CFGs were derived from the training data only). We examined three kinds of evaluation metrics:

$$\text{Coverage} = 1 - \frac{\text{\# sentences failed in parsing}}{\text{\# all sentences}}$$

[6] In case of English, it is similar to information about whether the verb is intransitive, transitive or ditransitive.

[7] Although MSLR parser integrates morphological and syntactic analysis of unsegmented sentences, it can perform only syntactic parsing by giving POS sequences as inputs.

[8] The number of terminal symbols does not change because we have not modified any POS tags under our policy.

Table 1: The number of CFG rules and the number of parse results (EDR original CFG vs EDR modified CFG)

	# CFG rules	# non-terminals	# terminals	# parse results
EDR (original)	1,694	249	600	1.868×10^{12}
EDR (modified)	1,949	279	600	9.355×10^{5}

Table 2: Coverage and recall (EDR original CFG vs EDR modified CFG)

	Coverage	Recall
EDR (original)	98.51%	96.63%
EDR (modified)	97.32%	95.88%

$$\text{Recall} = \frac{\text{\# sentences parsed correctly}}{\text{\# all sentences}}$$

$$\text{Sentence Accuracy} = \frac{\text{\# sentences parsed correctly in the top-}n\text{ parse results}}{\text{\# all sentences}}$$

“Sentences failed in parsing” means no result can be created in parsing the sentences. “Sentences parsed correctly” means the sentences in which all constituents are labeled correctly (i.e. exact match) in all the parse results, and “Sentences parsed correctly in the top-n parse results” means the sentences in which all constituents are labeled correctly in the top-n parse results ranked by PGLR model. Since the parse results are re-analyzed using semantic information in the subsequent processing, the structure of the parse result must match the correct structure exactly. That is why we use this evaluation metric rather than labeled precision and labeled recall, which are commonly used in evaluation of parsing.

Results are shown in Table 2 and Figure 1. Coverage and recall decreased by around 1%. Despite the decrease of coverage and recall, sentence accuracy increased about 8% under assumption that the top-100 parse results are re-analyzed in the subsequent processing. On the other hand, only the top-10 parse results are enough for the EDR modified CFG to overcome the accuracy among top-100 parse results using the EDR original CFG.

Some readers might take it for granted that sentence accuracy increases if the EDR modified corpus is used as a gold-standard because certain difficult decisions are not made in annotation and left to the subsequent processing. To test the accuracy if the EDR original corpus is used as a gold-standard, we randomly selected 100 sentences from the EDR modified corpus and examined dependency accuracy (the percentage of correct dependency relations out of all dependency relations) of the top parse results ranked by PGLR model (the EDR original corpus is used as a gold-standard). Since phrase structure is annotated in the corpus and the

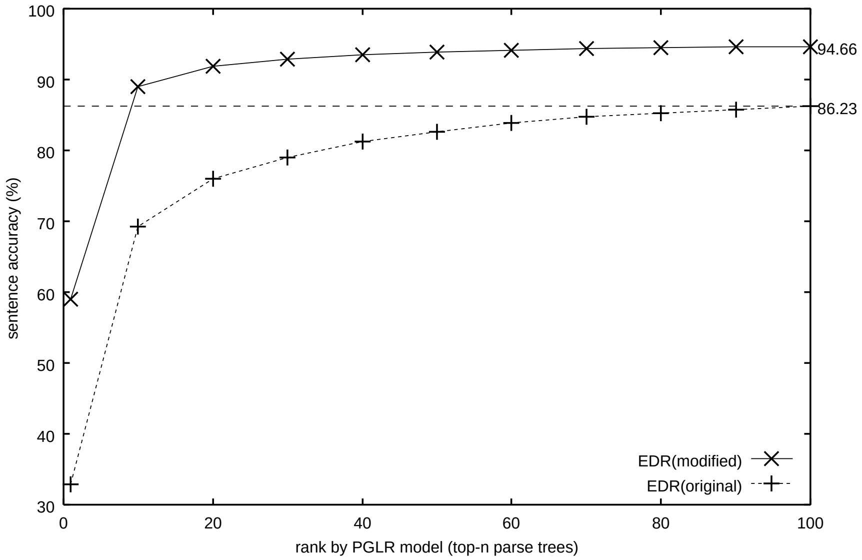


Figure 1: Sentence accuracy (EDR original CFG vs EDR modified CFG)

EDR modified CFG does not create dependency structures but phrase structures, we converted the parse results and structures in the EDR original corpus to dependency structures. Since the CFG does not determine adnominal phrase attachment, we assume that every ambiguous adnominal phrase attaches to the nearest noun. Whether the relation between two units is conjunctive or not is distinguished in this evaluation. 96 sentences were correctly segmented into Japanese phrasal units (*bunsetsu*), and dependency accuracy was 89.23%, which rivals the state-of-the-art dependency analysis using support vector machine, maximum entropy, etc [7, 8, 19] [9] although no semantic information is considered yet. We expect that the accuracy will increase as soon as semantic information is incorporated in the subsequent processing. The method of incorporating semantic information is left for future research.

4.2 Evaluation on the RWC corpus

There is a problem with using the CFG derived from the EDR corpus: there is no morphological analyzer based on the POS system used in this corpus. Thus we evaluated on the RWC corpus, a tagged corpus whose POS system is based on the Japanese morphological analyzer, ChaSen [12]. We extracted 16,421 sentences (on

[9] We cannot compare their model with ours absolute equity because they use different corpus and carry out their experiment under different conditions.

average 21.71 morphemes in each sentence) from it and we annotated the "modified corpus" only without annotating the "original corpus" according to [11]. We refer the CFG derived from the corpus as "RWC CFG".

The POS system of the RWC corpus (i.e. the POS system of ChaSen) is not sufficient for syntactic parsing. For instance, particles (postpositions) should be classified by word [17]. Some word sequences such as phrases which act as auxiliary verbs should be merged to reduce unnecessary ambiguity. We convert POS tags in the RWC corpus automatically before annotating. The main changes in POS tags are follows:

1. Numeral sequences are merged (ChaSen splits numeral sequences along characters).

2. Case particles are classified by case.

3. Verb endings for past tense (e.g. "*ta*"), gerund (e.g. "*te*") and others (e.g. "*tara*", "*tari*") [10] are merged with the previous verb [11].

4. Sequences of alphabet (i.e. roman) characters are labeled as common noun.

5. Word sequences which act as auxiliary verbs (e.g. "*noda*") are merged.

6. Suffixes for changing nouns to verbs (e.g. "*suru*") are separated from other verbs.

7. Symbols which are usually used at the end of sentences (e.g. question mark) are separated from other symbols [17].

8. Adverbs which are also used as noun modifiers are separated from other adverbs (similar to "Adverbial Classification" in [16]).

9. The latter verbs of verb sequences in [13] are labeled as auxiliary verb. For instance, a verb sequence "*fuki kesu* (blow out)" consists of two verbs "*fuku* (blow)" and "*kesu* (put out)", and the latter verb "*kesu*" is labeled as auxiliary verb.

We evaluated on the corpus and the CFG derived from it in the same way as we did for the EDR corpus. Results are shown in Table 3, Table 4 and Figure 2. The number of parse results was 9.599×10^4, coverage and recall were 98.38% and 97.18% respectively, and sentence accuracy among top-100 parse results was 95.76%. These results are comparable to the evaluation on the EDR corpus[11].

[10] This type of verb ending is called "*ta*-series ending" [11].

[11] We have not examined dependency accuracy, since we did not annotate the "original corpus".

Table 3: The number of CFG rules and the number of parse results (EDR modified CFG vs RWC CFG)

	# CFG rules	# non-terminals	# terminals	# parse results
EDR (modified)	1,949	279	600	9.355×10^5
RWC	2,556	290	391	9.599×10^4

Table 4: Coverage and recall (EDR modified CFG vs RWC CFG)

	Coverage	Recall
EDR (modified)	97.32%	95.88%
RWC	98.38%	97.18%

5 Conclusion

Although a large-scale CFG can be derived from a syntactically annotated corpus, in general, such CFGs create a large number of parse results. The principal cause is due to the fact that such CFGs are not built so as to sufficiently limit the ambiguity. We show that a practical large-scale CFG for syntactic parsing can be built by investigating the cause of increased ambiguity and modifying a corpus and consequently a CFG to remove the cause of such ambiguity.

Since we assume that the parse results created by our CFG are re-analyzed in the subsequent processing, we have to provide a method for re-analysis of the parse results. Our policy for annotating a corpus has been considered with several types of ambiguity: structure of compound noun, adnominal phrase attachment, adverbial phrase attachment and conjunctive structure. We are planning to provide each method individually and integrate them into one processing.

References

[1] Charniak, Eugene (1996) Tree-bank Grammars. In *the 13th National Conference on Artificial Intelligence*, pp. 1031–1036.

[2] Dickinson, Markus and Meurers, W. Detmar (2003) Detecting Errors in Part-of-Speech Annotation. In *the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*.

[3] Dickinson, Markus and Meurers, W. Detmar (2003) Detecting Inconsistencies in Treebanks. In *the 2nd Workshop on Treebank and Linguistic Theories (TLT 2003)*.

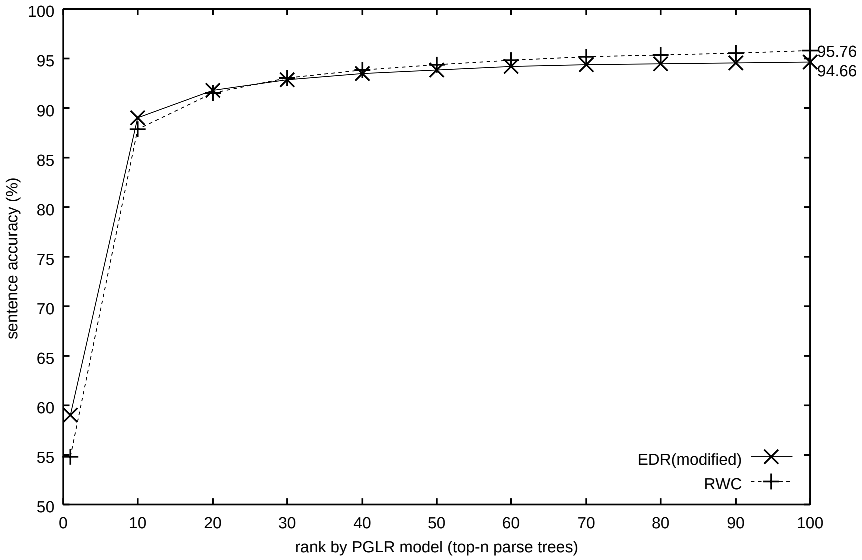


Figure 2: Sentence accuracy (EDR modified CFG vs RWC CFG)

[4] EDR (1994) EDR Electronic Dictionary User's Manual, 2.1 edition. In Japanese.

[5] Hashida, Koichi, Isahara, Hitoshi, Tokunaga, Takenobu, Hashimoto, Minako, Ogino, Shiho and Kashino, Wakako (1998) The RWC Text Databases. In *the 1st International Conference on Language Resource and Evaluation (LREC 1998)*, pp. 457–461.

[6] Inui, Kentaro, Sornlertamvanich, Virach, Tanaka, Hozumi and Tokunaga, Takenobu (2000) Probabilistic GLR parsing. In Bunt, Harry and Nijholt, Anton (eds) *Advances in Probabilistic and Other Parsing Technologies*, pp. 85–104. Kluwer Academic Publishers.

[7] Kanayama, Hiroshi, Torisawa, Kentaro, Mitsuishi, Yutaka and Tsujii, Jun'ichi (2000) A Hybrid Japanese Parser with Hand Crafted Grammar and Statistics. In *the 18th International Conference on Computational Linguistics (COLING 2000)*, pp. 411–417.

[8] Kudo, Taku and Matsumoto, Yuji (2002) Japanese Dependency Analysis Using Cascaded Chunking. In *Conference on Computational Natural Language Learning (CoNLL 2002)*.

[9] Kurohashi, Sadao and Nagao, Makoto (1994) A Syntactic Analysis Method of Long Japanese Sentences Based on the Detection of Conjunctive Structures. *Computational Linguistics*, 20(4), pp. 507–534.

[10] Klein, Dan and Manning, Christopher D. (2003) Accurate Unlexicalized Parsing. In *the 41st Annual Meeting of Association for Computational Linguistics (ACL 2003)*, pp. 423–430.

[11] Masuoka, Takashi and Takubo, Yukinori (1992) *Kiso Nihongo Bunpou (Foundation of Japanese Grammar)*, Kurosio Shuppan. In Japanese.

[12] Matsumoto, Yuji, Kitauchi, Akira, Yamashita, Tatsuo, Hirano, Yoshitaka, Matsuda, Hiroshi, Takaoka, Kazuma and Asahara, Masayuki (2000) *Japanese Morphological Analysis System ChaSen version 2.2.1 Manual*. Nara Institute of Science and Technology.

[13] Nomura, Masaaki and Ishii, Masahiko (1987) *Fukugoudoushi Shiryoushuu (Collection of Data about Compound Verbs)*, National Institute for Japanese Language. In Japanese.

[14] Noro, Tomoya, Hashimoto, Taiichi, Tokunaga, Takenobu and Tanaka, Hozumi (2004) Building a Large-Scale Japanese CFG for Syntactic Parsing. In *the 4th Workshop on Asian Language Resources (ALR 2004)*, pp. 71–78.

[15] Okazaki, Atsushi, Shirai, Kiyoaki, Tokunaga, Takenobu and Tanaka, Hozumi (2001) A Syntactic Annotation Tool with User Navigation. In *the 15th Annual Conference of Japanese Society for Artificial Intelligence*. In Japanese.

[16] Schiehlen, Michael (2004) Annotation Strategies for Probabilistic Parsing in German. In *the 20th International Conference on Computational Linguistics (COLING 2004)*, pp. 390–396.

[17] Shirai, Kiyoaki, Tokunaga, Takenobu and Tanaka, Hozumi (1995) Automatic Extraction of Japanese Grammar from a Bracketed Corpus. In *Natural Language Processing Pacific Rim Symposium*, pp. 211–216.

[18] Shirai, Kiyoaki, Ueki, Masahiro, Hashimoto, Taiichi, Tokunaga, Takenobu and Tanaka, Hozumi (2000) MSLR Parser – Tools for Natural Language Analysis. *Journal of Natural Language Processing*, 7(5), pp. 93–112. In Japanese.

[19] Uchimoto, Kiyotaka, Murata, Masaki, Sekine, Satoshi and Isahara, Hitoshi (2000) Dependency Model Using Posterior Context. In *the 6th International Workshop on Parsing Technologies (IWPT 2000)*.

Automatic Node Insertion for Treebank Deepening

Yvonne Samuelsson and Martin Volk
Stockholm University
Department of Linguistics
106 91 Stockholm
Sweden
yvonnesamuelsson@yahoo.se and volk@ling.su.se

1 Background

Creating a treebank is a time-consuming task. The Part-of-Speech tagging of the sentences is fast and automatic, with only minor corrections afterwards, since there are good taggers out there today. However, parsing is still a task that needs to be done semi-automatically, where the human annotator has to make many decisions manually.

We are working on a German-Swedish parallel treebank, where the data consists of the first chapter of Jostein Gaarder's novel Sophie's World (the Norwegian original is [4]). The German treebank contains 225 sentences and the Swedish one 216 sentences. The initiative for using this text comes from the Nordic Treebank Network[1], which has an ongoing project to syntactically annotate the first chapter of this book in the Nordic languages. This text was chosen since it has been translated into a vast number of languages and since it includes interesting linguistic properties such as direct speech.

For the annotation of the German part we used the treebank editor Annotate[2]. It includes Thorsten Brants' statistical Part-of-Speech Tagger and Chunker. The PoS tagger is trained with the STTS (Stuttgart-Tübingen TagSet [8]) for German. The chunker follows the NEGRA/TIGER annotation guidelines [7, 2], which gives a rather flat phrase structure tree. This means for instance no unary nodes, no "unnecessary" NPs (noun phrases) within PPs (prepositional phrases) and no finite VPs (verb phrases).

[1] The Nordic Treebank Network is headed by Joakim Nivre. See www.masda.vxu.se/∼nivre/research/nt.html

[2] Annotate has been developed at the University of Saarbrücken. See www.coli.uni-sb.de/sfb378/negra-corpus/annotate.html

Using a flat tree structure for manual treebank annotation has two big advantages for the human annotator:

1. the annotator needs to make fewer decisions, and
2. the annotator has a better overview of the trees.

This comes at the prize of the trees not being complete from a linguistic point of view. One could ask why an NP that consists of only one daughter is not marked, or why an NP that is part of a PP is not marked, while the same NP outside a PP is explicitly annotated. These restrictions also have practical consequences: If certain phrases (e.g. NPs within PPs) are not explicitly marked, then they can only indirectly be searched for in corpus linguistics studies.

In addition to the linguistic drawbacks of the flat syntax trees, they are also problematic for node alignment in a parallel treebank. Our goal is to align subsentential units (such as phrases and clauses) so that we get fine-grained correspondences between languages. Our alignment focuses on meaning, rather than sentence structure. This means that sentences can have alignment on a higer level of the tree (for instance if the S-node carries the same meaning in both languages), without nescessarily having alignment on lower levels (for instance an NP without correspondence). We prefer to have “deep trees” to be able to draw the alignment between the German sentences and the parallel Swedish sentences on as many levels as possible; in fact, the more detailed the sentence structure is, the more expressive is our alignment.

2 Building treebanks with automatic node insertion

We first annotated the German sentences semi-automatically, in the flat manner, according to the TIGER guidelines ([3] and [1]) and then automatically deepened the flat syntax trees. This was achieved by a Perl-program, which automatically inserts nodes to create the deeper structure. However, these insertions must be totally un-ambiguous, so that no errors are introduced.

2.1 The node insertion program

The input for this program is a tree description in TIGER-XML [5], an interface format which can be created and used by the treebank tool TIGERSearch[3]. The output is a deepened TIGER-XML tree. Our deepening program can be called with or without the _i-flag. When the flag is used, the program adds the marker

[3] See also www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/doc/html/TigerXML.html.

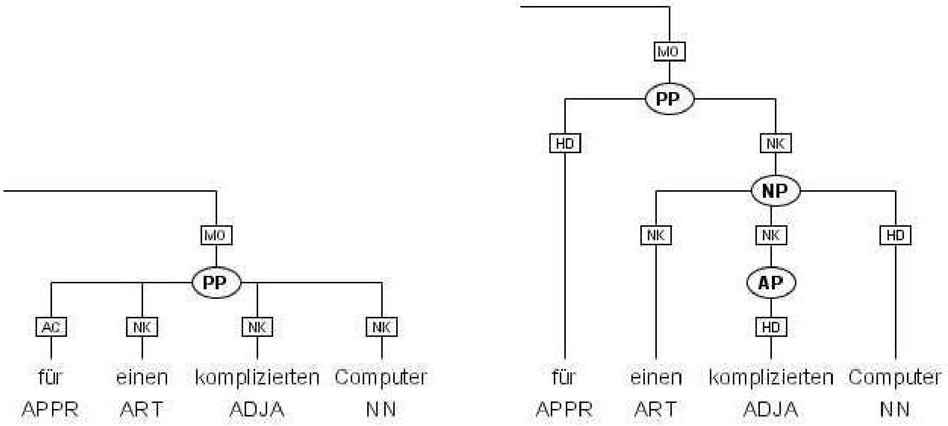


Figure 1: Part of a German tree, annotated in the flat manner (to the left) and after automatic node insertion (to the right).

_i to every automatically inserted node, enabling checks of the insertions in a tool like TIGERSearch. Figure 1 shows an example tree before and after the automatic insertion of an adjective phrase node and a noun phrase node.

There are basically two sets of rules; rules for the insertion of unary nodes and rules for handling other nodes. The first set of rules insert adjective phrases (APs), adverbial phrases (AVPs), noun phrases (NPs) and verb phrases (VPs). One simple rule is the one which inserts an AP if there is an NP with a direct adjective child (ADJA). More complex rules are e.g. the rules for handling NPs. The main point is that an NP that is the child of a PP should be marked in the same way as any other NP. These rules are listed in table 1.

We have three rules for unary branching nodes, which state that

1. If we have an NP with an AG (genitive attribute), APP (apposition) or GL (prenominal genitive) child, then this child is annotated as NP.
2. If there is an S or CNP (coordinated nominal phrase) with a direct MPN (multi-word proper noun) child, then this child is annotated as NP.
3. If there is an S, VP or CNP with a nominal (noun, pronoun or the like) child, then this child is annotated as NP.

Another rule states that unless there is only an AP, AVP or CNP together with

Unary branching nodes:

The format of the rules: ’flat structure’ ⟹ ’deep structure’,
with nodes in parentheses, edge labels in brackets; X standing for any label.

if NP with a direct genitive or apposition child, e.g. *Sofies Mutter*,
then insert an NP:
(NP) →[AG|APP|GL]→ NE ⟹
(NP) →[AG|APP|GL]→ (NP) →[HD]→ NE

if S or CNP with a direct MPN child, then insert an NP:
(S|CNP) →[X]→ MPN ⟹
(S|CNP) →[X]→ (NP) →[HD]→ MPN

if S or VP or CNP with a direct noun or pronoun child, then insert an NP:
(S|VP|CNP) →[X]→
(NN|NE|PPER|PDS|PRF|PPOSS|PIS|PRELS|PWS|TRUNC) ⟹
(S|VP|CNP) →[X]→ (NP) →[HD]→
(NN|NE|PPER|PDS|PRF|PPOSS|PIS|PRELS|PWS|TRUNC)

Other nodes:

All children of a PP except for the preposition are marked as NP,
unless there is only an AP (e.g. *seit längerem*) or an AVP (e.g. *bis morgen*)
or a CNP in the NP.

A coordinated noun phrase (CNP) does not get an NP mother
if it is the child of an S or a PP. But if the CNP has siblings
(typically modifiers) that belong to the same NP, then it gets an NP mother.

Table 1: Rules for insertion of NP nodes

the preposition in a PP, everything but the preposition should be made into an NP. This binds the parts of an NP inside the PP together (like in figure 1). Finally the program checks that every NP has a head (this is especially important since the automatic alignment program created for [6] is based on the fact that every node has a head).

The node insertion rules should all be reliable because they are un-ambiguous, as long as the manual part of the annotation is correct according to the guidelines. However, there are some problems with the program. Cardinal numbers are still not handled since they can be of different types, e.g. adjective-like in *25 Computer* and noun-like in *im Jahre 2000*. Therefore they are not un-ambiguous and cannot easily be handled automatically. Adverbs in adjective phrases are still not handled (they should have their own AVP) and there are several STTS-tags that do not have their own node label, e.g. PTKNEG for the negation particle. Several of them could probably be made into adverbial phrases (AVPs).

2.2 Creating the German treebank

The main gain of the program is of course to speed up the work in creating treebanks. This worked very well for the German trees. For the 225 German sentences in the first chapter of Sofies Welt, with 3146 tokens, we thus semi-automatically annotated 1426[4] nodes with 4570 edges. The automatic node insertion resulted in a total of 2278 nodes with 5422 edges. This means an increase of almost 60% with regards to the nodes. 548 of the inserted nodes were NPs (of which 420 are unary), 143 were APs (all unary), 160 AVPs (all unary) and 1 VP. Within PPs 189 NPs were inserted and 45 APs.

The semi-manual annotation of the flat structured German sentences took about 5 hours. This means almost 5 nodes per minute, which in turn would mean that we saved almost 3 hours of annotation time due to the automatic node insertion. This is not entirely true, since not all nodes are equally problematic. An annotator needs more time to make a decision for a problematic node and many of the nodes that are automatically inserted are easy to create during semi-manual annotation. But it gives a hint about the possible time gain in the annotation process, when creating large treebanks.

2.3 Creating the Swedish treebank

When aiming for a parallel treebank it is advantageous to handle the annotation of the Swedish sentences similar to the annotation of the German sentences. Tra-

[4] These numbers are taken from TIGERSearch, which also includes a top node for each sentence, with edges to the punctuation.

ditionally, however, a different PoS tagset has been used for Swedish, called the SUC-tagset[5]. We trained Brants' TnT-tagger with the SUC-tagset for automatic PoS tagging. Unfortunately there is no constituent structure treebank for Swedish that could be used for training a chunker with resulting structures corresponding to the German sentences. Therefore we mapped the SUC-tags into the German STTS to be able to re-use the German chunker in Annotate for Swedish. This works nicely. A small experiment, where the children were manually selected, shows that the German chunker suggests 89% correct node labels and 93% correct edge labels for Swedish ([9] and [6]).

Still Swedish annotation takes more time than German annotation (over 10 hours), mainly due to the fact that the NEGRA annotation guidelines are written for German and there is no appropriate Swedish annotation manual. We had to adapt the guidelines to Swedish as we went along. Even though the Annotate tool mostly suggests the correct node and edge labels, there are still a number of difficult cases for the annotator to decide.

One example is the difference in Swedish between prepositions and verb particles. In spoken language, prepositions are not stressed, while verb particles are. In some cases the word is a preposition, not a verb particle, but it still is closer to the verb and therefore behaves "strangely". One example is the relative clause

(1) [*hår*] *som varken gelé eller spray bet på*

([*hair*] *which neither gel nor spray would work on)*

In this relative clause the pronoun *som* is put in the beginning and is thus separated from the stranded preposition *på*. The annotator has to establish that this is indeed a preposition and then decide whether the pronoun should be in the prepositional phrase (with crossing branches) or not.

After annotation, the Swedish trees are automatically deepened in the same manner as the German trees. Since Swedish and German are similar languages, there are only minor differences between the insertion programs. For instance, pre-noun genitives in German are always assumed to be proper names (NE) but in Swedish they could also be regular nouns (NN) (e.g. *vid världens ände* (literally 'at the world's end')).

One difference between Swedish and German is that a PP in Swedish can consist of a preposition and a sentence or verb phrase. A phrase like

(2) [*göra*] *min av att svara*

([*make*] *as if to answer, literally 'make an expression of answering')*

[5] SUC, the Stockholm-Umeå Corpus, is a 1 million word representative Swedish corpus which is annotated with Part-of-Speech tags, morphological tags, lemmas and name classes. All of SUC is manually checked.

should contain this structure: [PP av[VP att svara]]. This means that we have to add the possibility of having an S or VP in the PP for Swedish.

Some problems with the node insertion for Swedish actually resulted from errors in the mapping from STTS-tags back into SUC-tags. For instance, the STTS-tag KOKOM (comparison particle, without sentence) is translated into PR (preposition). According to the German annotation guidelines the KOKOM is part of the NP, while of course a preposition should have a PP mother node. Since the translation of the STTS-tags back into SUC-tags is done after the node insertion, this creates erroneous tree structures. A look through the SUC-database showed that the correct translation of KOKOM should have been KN (conjunction).

We also experimented with a rule-based Swedish chunker, to pre-process the sentences before loading them into Annotate for the semi-manual parsing. The chunker produces deep structures for NPs and PPs, which makes part of our later deepening obsolete. But it turned out that this type of deep pre-chunking thwarts the advantages of the flat tree annotation. In fact it leads to more decisions and worse overview for the human annotator. This is mostly due to the fact that our chunker builds all the nodes it can find, which gives trees of “uneven depth”. The solution would be a pre-chunker, which computes only flat and “safe” structures so that the human annotator can concentrate on deciding ambiguous attachments.

3 How flat can a tree be?

The successful automatic node insertion gave rise to the question of whether the flat annotation according to the NEGRA guidelines could be made even flatter. In other words, is it possible and feasible to define a minimal set of human annotation decisions with a maximum number of automatically inserted nodes and labels? A minimal form for the manual annotation, where the rest of the nodes are automatically inserted later, could save a lot of time. The deepening of course still has to be totally safe, i.e. un-ambiguous.

The problem is defining a minimal form that is still linguistically plausible. If the form is too minimalistic and skeletal, it will be hard for the human annotator to still maintain an overview and to see what is correct in the annotation. The advantages of the flat annotation should not be renounced.

It is rather complex to determine the nodes that are always un-ambiguous. There are mainly two difficulties. The first one is that a node has to be manually inserted (i.e. a manual decision has to be made) if any of its edges are ambiguous. This means for instance that we cannot automatically insert a missing top node S (sentence), since the distinction between subject and object is often ambiguous (for the computer).

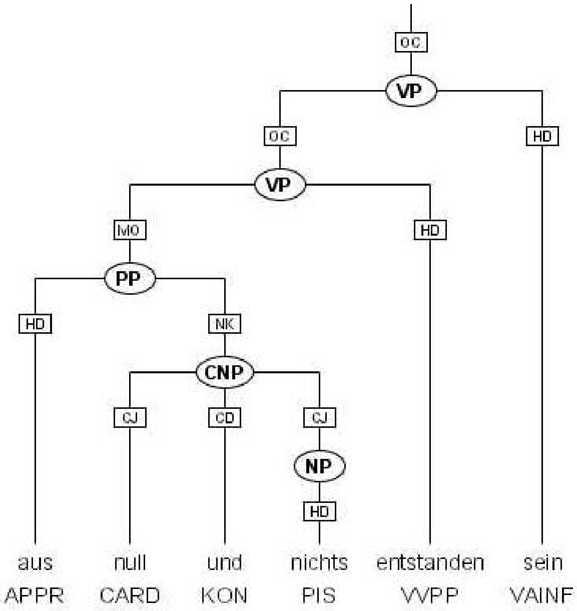


Figure 2: Example of a nested VP which could be automatically inserted.

The second difficulty in finding the un-ambiguous nodes lies in language differences. One example of this is the node VZ (*zu*-marked infinitive) which in German un-ambiguously groups the infinitive marker immediately followed by an infinitive verb. But the Swedish equivalent (marked with *att*) is ambiguous. For instance the marker and the verb do not have to be adjacent, there might be an intermediary adverb. Also the word *att* might be something other than an infinitive marker.

One example of nodes that could be left out in the manual annotation for both German and Swedish (to be automatically inserted afterwards) is nested VPs in verb chains. According to the NEGRA guidelines we need nested VPs for every verb in a chain (see figure 2). But it would facilitate the manual annotation if all verbs in such a chain could be entered into one VP (i.e. as sister nodes on the same level) in the flat tree structure, and the nesting could be done automatically afterwards. This nesting is unambiguous in both German and Swedish since the order of the verbs within the verb group is fixed (with few exceptions like the German *Oberfeldumstellung* which need to be handled manually).

4 Conclusions

In creating a parallel treebank we investigated how much of the manual labour involved can be carried out automatically. We built flat phrase structure trees according to the NEGRA guidelines. Then we had a program insert un-ambiguous nodes to get a deeper and more detailed structure. This insertion step rendered about 60% more nodes.

Our ultimate goal is to make the tree structure even flatter. A minimal tree is a tree where all and only the ambiguous attachments have been decided by the human annotator. We found that some more nodes could be automatically inserted (in addition to the ones that are already left out according to the NEGRA guidelines). This means that creating treebanks in the future can be made into a less time consuming task.

References

[1] S. Albert, J. Anderssen, R. Bader, S. Becker, T. Bracht, S. Brants, T. Brants, V. Demberg, S. Dipper, P. Eisenberg, S. Hansen, H. Hirschmann, J. Janitzek, C. Kirstein, R. Langner, L. Michelbacher, O. Plaehn, C. Preis, M. Pussel, M. Rower, B. Schrader, A. Schwartz, G. Smith, and H. Uszkoreit. TIGER Annotationsschema. July 2003.

[2] Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, pages 24–41, Sozopol, Bulgaria, 2002.

[3] Thorsten Brants, Stefanie Dipper, Peter Eisenberg, Sabine Kramp, Cordula Preis, Marcus Pussel, Anne Schwartz, George Smith, and Hans Uszkoreit. TIGER Annotationsschema. May 2000.

[4] Jostein Gaarder. *Sofies verden: Roman om filosofiens historie*. Aschehoug, 1991.

[5] Esther König and Wolfgang Lezius. The TIGER language - a description language for syntax graphs. Part 1: User's guidelines. Technical report, IMS, 2002.

[6] Yvonne Samuelsson. Parallel Phrases - Going automatic. Further experiments towards a German-Swedish parallel treebank. Master's thesis, Stockholm University, http://ling16.ling.su.se:8080/PubDB/doc_repository/
samuelssonautomatic2004.pdf, 2004.

[7] W. Skut, B. Krenn, T. Brants, and H. Uszkoreit. An annotation scheme for free word order languages. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 88–95, Washington, DC, 1997.

[8] Christine Thielen, Anne Schiller, Simone Teufel, and Christine Stöckert. Guidelines für das Tagging Deutscher Textkorpora mit STTS. Technical report, IMS and SfS, 1999.

[9] Martin Volk and Yvonne Samuelsson. Bootstrapping parallel treebanks. In *Proceedings of the 5th International Workshop on linguistically Interpreted Corpora (COLING 2004)*, pages 63–69, Geneva, Switzerland, August 2004.

Steps towards a GENIA Dependency Treebank

Gerold Schneider, Fabio Rinaldi, Kaarel Kaljurand and Michael Hess
Institute of Computational Linguistics, University of Zurich
{gschneid,rinaldi,kalju,hess}@ifi.unizh.ch

1 Introduction

In this paper we describe on-going work aimed at creating a dependency-based annotated treebank for the BioMedical domain. Our starting point is the GENIA corpus [14], which is a corpus of 2000 MEDLINE abstracts, which has been manually annotated for various biological entities, according to the GENIA Ontology.[1]

There is an exponential growth of published research in this sector, which makes it difficult even for the experts to follow the recent developments. This creates the need for tools that can automatically process the research literature and extract only relevant information, such as interactions between genes and proteins. In order for these tools to be developed, annotated resources, such as corpora and Treebanks are of fundamental importance. Such resources will support the development of practical domain-specific information extraction tools.

For an information extraction application extracting relations between genes and proteins [19] the dependency based parser Pro3Gres [20, 21] has been used. Pro3Gres is an open, modular and highly parameterized system. The module interaction can be seen in fig. 1. Pro3Gres is fast and robust, it parses the entire GENIA in under 3 hours. Although its performance is competitive, a considerable effort will have to go into correcting it to achieve a nearly error-free treebank.[2]

The creators of GENIA are currently planning to release a version of GENIA enriched with syntactic annotations based on a HPSG analysis of the corpus [22]. Our work can be considered parallel and complementary to theirs. We intend to compare and coordinate our results with the HPSG parsing based GENIA Treebank that is becoming available from the GENIA project. We also plan to make a dependency analysis widely available for research activities.

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/topics/Corpus/genia-ontology.html

[2] Whether this task can be be completed will depend on future funding.

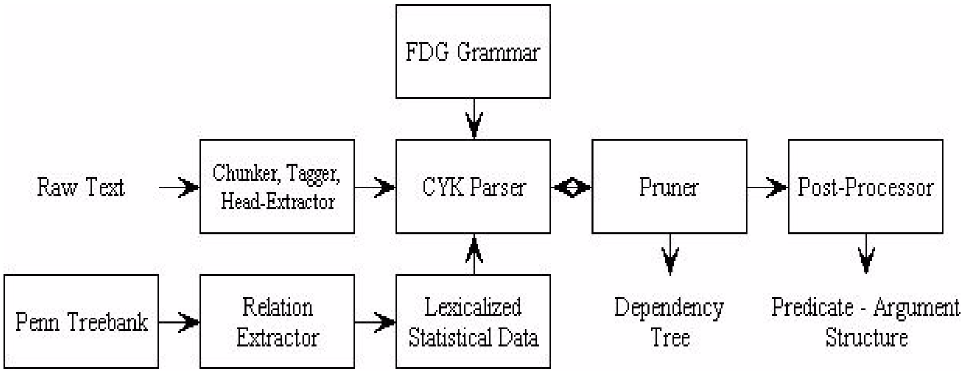


Figure 1: Pro3Gres flowchart

LTPos-tagged Parsing	Subject	Object	noun-PP	verb-PP
Precision	91.5	90.3	70.5	72.5
Recall	80.6	83.4	64.0	86.4

Table 1: Percentage results of evaluating the LTPos tagger based parser output on Carroll's test corpus on subject, object and PP-attachment relations

In this paper we show that the annotation effort can be reduced by using a high-performance parser. For this purpose, we test settings that allow us to optimize on recall and on precision respectively. The errors of the Pro3Gres parser are analyzed in detail. After an evaluation on a general corpus and on the GENIA corpus, we describe methods to minimize the annotator's task: high recall and high precision parsing. A practical evaluation discusses the impact of errors for obtaining domain knowledge and we conduct an analysis of remaining errors.

2 General Evaluation

[15] suggests evaluating on the linguistically meaningful level of syntactic relations. For the first evaluation, a hand-compiled gold standard following this suggestion is used [5]. It contains the grammatical relations of 500 random sentences from the Susanne corpus. Results are in table 1. The mapping between our and Carroll's annotation is discussed in [20].

A detailed analysis breaks down the errors into classes, in table 3 for PP-attachment, and in table 4 for subject precision. The analysis identifies mistagging and mischunking as generally important error sources in addition to parsing, but also that differing grammar assumptions are a problem. For the evaluation on the Carroll corpus, a mapping to our relation types was necessary [20]. Mapping one annotation scheme to another is non-trivial and can only lead to indicative results

MaxEnt-tagged Parsing	Subject	Object	noun-PP	verb-PP
Precision	92.4	89.5	72.9	70.3
Recall	81.1	83.9	64.5	84.4

Table 2: Percentage results of evaluating Charniak's MaxEnt tagger based parser output on Carroll's test corpus on subject, object and PP-attachment relations

	Attachment Error	Head Extraction Error	Chunking or Tagging	compl/prep Error	Grammar Mistake or incompl. Parse	Grammar Assumption
Noun-PP Prec.	22	1	8	0	3	3
Verb-PP Prec.	12	1	5	1	1	2
Noun-PP Recall	25	1	14	0	12	5
Verb-PP Arg. Recall	2	0	1	0	0	0
Total	61 51%	3 3%	28 24%	1 1%	16 13%	10 12%

Table 3: Error Classification of PP-Attachment errors from the first 100 Carroll corpus sentences

[11]. An obvious response to the big influence of tagging and chunking errors was to try a different tagger. Instead of LTPos [16] a Maximum Entropy tagger has been tested [6]. The results are in table 2. The comparison yields no clear result, but has prompted us to do the first two high-precision experiments in section 4.

For the second evaluation, 100 random GENIA sentences have been manually annotated and compared to the output of the parser[3]. We keep the manual annotation for multi-word biological terms as chunked input to the parser[4]. The results are in table 5.

3 High Recall Parsing

The annotation task is greatly facilitated if the annotator, instead of being asked to annotate every sentence manually, can choose from a (relatively short) ranked list of analyses. [3] have shown that parser-assisted annotation (in their case an interactive scenario with a shallow parser [2]) greatly increases annotation speed. Table 6 shows the increase in recall in relation to the length of the list of analyses.

[3] This is a small set. Average sentence length is 17.9 chunks, compared to 17.0 in the whole GENIA, so we can assume that it is fairly representative

[4] see [19] for the impact of keeping this information

	Spurious Error	Chunking or Tagging	Control	Parsing Error	Rel. Pronoun Resolution	Grammar Mistake or incompl. Parse	Grammar Assumption
Subject Precision	8	22	9	15	4	9	9

Table 4: Error Classification of Subject Precision errors of all Carroll corpus sentences

Percentages on GENIA	Subject	Object	noun-PP	verb-PP	subord. clause
Precision	90	93	85	82	68
Recall	87	91	82	84	73

Table 5: Evaluation of 100 sentences of the GENIA corpus, using multi-word term boundary information

	Carroll				GENIA			
High Recall	Subject	Object	noun-PP	verb-PP	Subject	Object	noun-PP	verb-PP
1 analysis	80.8	83.4	64.9	86.4	86.6	91.1	81.6	83.3
max. 2 analyses	81.4	83.6	70.4	89.9	87.7	91.1	85.4	83.3
max. 4 analyses	81.6	84.1	73.9	90.4	90.3	91.1	91.8	86.2
max. 8 analyses	81.8	84.1	75.2	91.4	91.3	91.1	93.7	86.2
max. 16 analyses	81.9	84.4	75.4	91.4	91.8	91.1	94.2	86.2

Table 6: Percentage results of recall among first N-ranked analyses

Lists longer than 16 readings of a sentence (which convey 4 2-way ambiguous relations) are thought to be prohibitively long for manual scanning.

The $subj$, obj and the two PP-relations together average above 90% recall in GENIA, which means that less than one in ten of these relations need to be added manually by the annotator. Generally, recall in GENIA is higher. This is due to the following reasons:

1. As we have annotated our test corpus with the Pro3Gres scheme, there are no spurious mapping errors
2. We can profit from the fact that GENIA contains near-perfect tagging and multi-word term (MWT) information
3. We have written an unsupervised learning module and applied it to GENIA. Based on the fact that sentence-initial <NP PP*> sequences are almost always unambiguous [8], it learns which nouns are allowed to be modified by several PPs and restricts noun modification by several PPs accordingly. This especially explains the very high noun-PP-attachment recall.

4 High Precision Parsing

In order to keep the necessity for intervention of a human annotator during corpus annotation to a minimum, it is desirable to recognize a maximum number of unproblematic relations. An alternative annotation scenario is thus to report the highest ranked parse and to point out to the human annotator the few difficult and highly ambiguous relations in a given analysis. Parsing methods that optimize precision while reducing recall up to an acceptable point are required. A related study

Experiment 1	Subject	Object	noun-PP	verb-PP
Precision	92.2	95.4	85.6	71.6
Recall	31.5	30.7	23.2	27.8

Table 7: Percentage results of Experiment 1: keeping only sentences with identical tags from two taggers, on Carroll's test corpus on subject, object and PP-attachment relations

Experiment 2	Subject	Object	noun-PP	verb-PP
Precision	94.1	93.0	73.3	75.4
Recall	76.4	78.8	60.5	80.3

Table 8: Percentage results of Experiment2: keeping only agreeing relations arising from parsing with two taggers, on Carroll's test corpus on subject, object and PP-attachment relations

on this subject is [4]. This field of research may eventually lead to the automatized detection of potential parsing errors. It is also important for building up knowledge databases automatically, where recall deficiencies are often compensated by natural language redundancy, but asserting wrong knowledge arising from low precision poses a serious problem.

Experiment 1: Tagger Agreement Different taggers often make different mistakes. In a simple experiment, only sentences where both taggers deliver identical tags are used. Precision increases, but the large cost of decrease in recall is unacceptable, as shown in table 7.

Experiment 2: Grammatical Relations Agreement when using different Taggers In order to minimize the loss in recall in the previous experiment, the output of each tagger is used as input to the LTChunk chunker and the Pro3Gres parser. Only grammatical relations that are different due to the tagging differences are discarded. The increase in precision is similar to experiment 1 (noun PP-attachment is slightly worse) while the decrease in recall is much more moderate, as table 8 shows.

Experiment 3: Parsing Alternatives Agreement In this experiment, the relation intersection between the 2 top ranked analyses is kept. This amounts to discarding only the most ambiguous relation of any given sentence. The decrease in recall (table 9) is higher than in experiment 2. Mainly the PP-attachment relations profit, which are often the most ambiguous relations, and which are more affected by attachment ambiguities than other relations.

Experiment 3		Subject	Object	noun-PP	verb-PP	subord. S
Carroll	Precision	92.6	90.1	76.6	76.7	68.2
	Recall	76.8	63.6	53.7	67.2	n/a
GENIA	Precision	91.1	93.4	87.0	84.2	65.2
	Recall	78.1	65.8	68.0	70.5	60.4

Table 9: Percentage results of Experiment 3: discarding the most ambiguous relation in each sentence, for subject, object, PP-attachment and subordinate sentence relations

```
Interaction_NN of_IN nuclear_JJ extracts_NNS from_IN various_JJ cell_NN lines_NNS and_CC tissue_NN
with_IN the_DT MNP_NN site_NN leads_VBZ to_TO the_DT formation_NN of_IN fast-migrating_JJ
protein-DNA_JJ complexes_NNS with_IN similar_JJ but_CC distinct_JJ electrophoretic_JJ mobilities_NNS

prep('extract#3', 'of#2', _, '(<-)').               prep('line#5', 'from#4', _, '(<-)').
conj('tissue#7', 'and#6', _, '(<-)').               conj('line#5', 'tissue#7', 'and#6', '(->)').
prep('site#9', 'with#8', _, '(<-)').                modpp('line#5', 'site#9', 'with#8', '(->)').
modpp('extract#3', 'line#5', 'from#4', '(->)').     modpp('interaction#1', 'extract#3', 'of#2', '(->)').
subj('lead#10', 'interaction#1', _, '(<-)').        prep('formation#12', 'to#11', _, '(<-)').
prep('complex#14', 'of#13', _, '(<-)').             modpp('formation#12', 'complex#14', 'of#13', '(->)').
pobj('lead#10', 'formation#12', 'to#11', '(->)').   prep('mobility#16', 'with#15', _, '(<-)').
pobj('lead#10', 'mobility#16', 'with#15', '(->)').
```

Figure 2: A sample sentence with its top-ranked grammatical relation annotation

Experiment 4: Trust Short Distances Relation spanning short distances are intuitively thought to be easier for the parser to find. Experiment 4 discards all relations that are longer than a certain threshold. Length is measured in chunks. The experiment has been conducted at several distances for the Carroll test corpus (table 10) and for the 100 manually annotated GENIA sentences (table 11).

The results reveal interesting differences between different relation types. For $subj$, longer distances are almost as reliable. obj relations are almost exclusively very short. Subordinate clause relations are difficult and mostly very long, about 20% spanning at least 5 chunks. For envisaged applications, e.g. protein interaction relations, sentence subordination is less important. PP-attachment relations very strongly depend on distance. This is largely due to the fact that many PP-attachments across longer distances[5] are in competition with intervening other PPs and thus exponentially lower the baseline[6].

When comparing the two evaluation corpora and genres a major difference is PP-attachments. The complexity of medical language partly stems from very complex nouns with embedded PPs (see e.g. fig. 2). The noun-PP-attachment per sentence ratio is 2.1 in our GENIA 100 test corpus and 1.6 in Carroll. The fact that the performance on GENIA is better than on Carroll can largely be explained by our remarks in section 3.

[5] observe that "longer distances" does not entail a long-distance dependency traditionally expressed by coindexing or movement, although a considerable portion of the "longer distances" here are long-distance dependencies, for example fronted PPs attaching to the verb

[6] [7] describe for PP attachment that a sequence <verb-NP-PP*> with n PPs has C_{n+1} analyses, where C_{n+1} is the $(n+1)$'th Catalan number. The Catalan number C_n is defined as $\frac{1}{n+1}\binom{2n}{n}$

Experiment 4 on Carroll		Subject	Object	noun-PP	verb-PP	subord. S
Distance 1-2	Precision	94.3	90.5	76.0	85.7	74.1
	Recall	70.5	83.9	52.3	69.7	n/a
Distance 1-3	Precision	92.7	90.3	74.0	77.5	74.7
	Recall	75.5	84.1	59.2	78.3	n/a
Distance 1-4	Precision	92.2	90.0	73.5	75.2	70.8
	Recall	76.8	84.4	61.7	81.3	n/a
Distance 1-5	Precision	92.3	89.8	73.3	74.2	69.1
	Recall	78.6	84.4	62.5	82.3	n/a
Distance > 5	Precision	96.0	null	0.0	37.4	55.0
	Recall	5.4	null	0.0	2.0	n/a

Table 10: Percentage results of Experiment 4: discarding relations that span long distances, on Carroll's test corpus

Experiment 4 on GENIA		Subject	Object	noun-PP	verb-PP	subord. S
Distance 1-2	Precision	92.3	92.9	88.1	95.5	75.0
	Recall	57.1	91.1	79.0	64.7	14.5
Distance 1-3	Precision	89.5	92.9	87.2	87.6	84.0
	Recall	64.8	91.1	79.0	74.1	39.6
Distance 1-4	Precision	90.2	92.9	86.8	87.5	79.3
	Recall	69.4	91.1	79.5	77.7	43.8
Distance 1-5	Precision	90.9	92.9	85.6	85.6	71.8
	Recall	74.5	91.1	80.0	79.1	54.2
Distance > 5	Precision	89.3	null	0.0	41.7	57.1
	Recall	2.0	null	0.0	3.6	18.7

Table 11: Percentage results of Experiment 4: discarding relations that span long distances, on GENIA corpus relations

Experiment 5: Cut low probability parsing decisions In a first attempt, experiments with an increased probability cutoff at parse time were conducted. However, they had the effect of greatly increasing the amount of non-full parses, thus returning many local analyses that the syntactic parsing context would have disambiguated. Precision remained comparable, while recall dropped. In a second approach, the parsing algorithm remains unchanged, but only relations whose probability is above a certain threshold are reported. Here we profit from the fact that the Pro3Gres probabilities express decision probabilities at each given ambiguous point as suggested by [10]. In addition to offering a psycholinguistically plausible model this has the advantage that points of uncertain decisions and high entropy can be directly pinpointed. These experiments have been made on the highly ambiguous PP-attachment relations, see table 12.

Below threshold values of about 0.5 there is a reasonable trade-off in gained precision for lost recall. With higher thresholds, precision stagnates while recall drops off.

Experiment 5		Carroll		GENIA	
		noun-PP	verb-PP	noun-PP	verb-PP
Threshold 0.3	Precision	73.7	71.0	84.5	81.6
	Recall	64.2	84.8	79.0	82.7
Threshold 0.4	Precision	74.4	71.3	85.3	81.3
	Recall	63.6	84.3	78.1	80.5
Threshold 0.5	Precision	76.0	72.6	86.2	79.8
	Recall	61.3	81.3	72.4	71.2
Threshold 0.6	Precision	76.6	72.8	87.4	82.3
	Recall	56.2	73.2	70.4	59.7
Threshold 0.7	Precision	77.0	72.5	87.6	81.5
	Recall	52.6	66.1	68.6	51.8
Threshold 0.8	Precision	77.0	73.0	88.1	80.3
	Recall	51.2	63.1	64.8	45.3
Threshold 0.9	Precision	77.1	73.6	88.2	79.7
	Recall	50.9	62.1	64.8	43.9

Table 12: Percentage results of Experiment 5: discarding low-probability relations, on Carroll's and the GENIA test corpus

Experiments	Carroll				GENIA			
3,4,5 combined	ubject	Object	noun-PP	verb-PP	Subject	Object	noun-PP	verb-PP
Precision	92.6	90.1	78.9	80.5	92.4	93.5	87.9	88.1
Recall	75.0	63.4	51.2	67.2	67.3	67.0	66.7	65.5

Table 13: Percentage results of Experiments 3, 4 and 5 combined at threshold 0.4 and distances 1 to 5

Combinations Most of the above high-precision experiments can be combined in various ways. E.g. combinations of experiment 3, 4 and 5 are reported in tables 13 with threshold 0.4 and distances 1 to 5. This sample combination on the GENIA annotation task allows us to reach about 9 out of 10 precision at 2 out of 3 recall for all reported relations.

5 Practical Evaluation

Our interest lies in the discovery of domain specific relations, such as "Protein *activates* Gene". Most of the NLP techniques applied to the domain of molecular biology focus on the discovery of Entities, such as Genes and Proteins, (see for instance [1]). However there are also interesting applications aiming at detecting syntactic and semantic relations among those entities. Examples of systems aiming at detecting relations are the following:

- [9] identifies possible drug-interaction relations between proteins and chemicals using a "bag of words" approach applied to the sentence level.

- [17] reports on extraction of protein-protein interactions based on a combination of syntactic patterns.
- [12] describes a system (GENIES) which extracts and structures information about cellular pathways from the biological literature.
- [18] processes titles and abstracts of Medline articles focusing on relation identification (in particular the `inhibit` relation)
- [13] uses a template-based Information Extraction approach, focusing on the roles of specific amino acid residues in protein molecules

In order to discover domain specific relations we believe that an accurate detection of predicate/argument relations is essential. We have asked domain experts to evaluate the quality of the extracted relations, so far focusing on triples of the form (predicate - subject - object).[7]

A first evaluation was based on assigning a simple key code to each record: 'P' for positive (biologically relevant and correct, 53 cases), 'Y' for acceptable (biologically relevant but not completely correct, 102 cases) and 'N' (not biologically relevant or seriously wrong, 14 cases). This result was considered as encouraging as it showed 91.7% of relevant records.

On closer inspection of the expert results, we identified a number of 'typical cases', which we then asked the expert to evaluate in detail. In this second evaluation the expert had to evaluate each argument separately and mark it according to the following codes:

- [Y] the argument is correct and informative
- [N] the argument is completely wrong
- [Pr] the argument is correct, but it is a pronoun, and it would need to be resolved to be significant (e.g. "This protein").
- [A+] the argument is "too large" (which implies that a prepositional phrase has been erroneously attached to it)
- [A-] the argument is "too small" (which implies that an attachment has been omitted)

Despite parsing errors – some of which we are now correcting in the parser – the results can be considered satisfactory, as they show 86.4% and 58.6% correct results in the detection of subjects and objects (respectively). If all loose cases are considered as positive (excluding only the 'N' cases), these results increase to 93.5% and 99.4% (respectively).

[7] This evaluation has been performed in collaboration with Biovista (`http://www.biovista.com/`)

	Y	N	Pr	A+	A-
Subject	146	11	4	6	2
Object	99	1	4	59	6

Table 14: Distribution of GENIA parsing errors in the application-oriented evaluation

Recall Error Classification on GENIA High Recall Parsing									
	Adjective Trans.	Incompl. Grammar	Chunking Error	Tagging Error	Incompl. Parse	LDD Resol.	Annotation Problem	Attachment Error	Conjunction Error
Subject Recall	1	2	1	2	0	6	2	1	1
Object Recall	1	1	2	0	0	0	1	1	1
N-PP Recall	0	2	2	0	3	0	2	3	0
V-PP Recall	1	5	2	4	2	1	2	2	0
Total	3	10	7	6	5	7	7	7	2

Table 15: Analysis of recall errors on max. 16 GENIA high recall parsing

Let us consider a realistic annotation scenario using the high recall parsing method from section 3 with the annotator selecting the best of top 16 analyses. Over subject, object and PP-attachment relations, recall is 564/618 = 91.3%. 54 errors stemming from 34 sentences remain. Table 15 breaks down these errors into classes.

Bearing in mind that the annotation problem errors are spurious errors, that long-distance dependencies (LDDs) are often left underspecified by statistical parsers, and that the parser is affected by tagging and chunking mistakes, actual high recall parsing performance for the evaluated relations can be confirmed to reach 95%.

Pro3Gres is a modular system. Tagging and chunking are external processes whose output can be confirmed or corrected by the user. We are now investigating ways to integrate annotator feedback at an interactive and especially at the post-parsing stage. The latter triggers re-parsing erroneous sentences using the annotator's safe corrections.

6 Conclusion

We have shown that the annotation effort for building a syntactically analyzed corpus can be reduced by using a high-performance deep-linguistic parser and that it is possible to pin-point places of high entropy, to optimize on recall or on precision, respectively, to distinguish between more and less reliable relations.

We have shown that Pro3Gres can do full, deep-linguistic parsing of BioMedical texts at competitive speed and accuracy. The parser's errors have been analyzed in detail. We plan to compare and coordinate our grammatical relations output to the GENIA Treebank that is becoming available from the GENIA project.

References

[1] Sophia Ananiadou and Jun'ichi Tsujii, editors. *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, 2003.

[2] Thorsten Brants. Cascaded markov models. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*, pages 118–125, Bergen, Norway, 1999. University of Bergen.

[3] Thorsten Brants and Oliver Plaehn. Interactive corpus annotation. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2000),* Athens, Greece, 2000.

[4] John Carroll and Ted Briscoe. High precision extraction of grammatical relations. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, Taipei, Taiwan, 2002.

[5] John Carroll, Guido Minnen, and Ted Briscoe. Corpus annotation for parser evaluation. In *Proceedings of the EACL-99 Post-Conference Workshop on Linguistically Interpreted Corpora*, Bergen, Norway, 1999.

[6] Eugene Charniak. A maximum-entropy-inspired parser. In *Proceedings of the North American Chapter of the ACL*, pages 132–139, 2000.

[7] K. Church and R. Patil. Coping with synactic ambiguits or how to put the block in the box on the table. *American Journal of Computational Linguistics*, 8(3-4):139–149, 1982.

[8] Michael Collins and James Brooks. Prepositional attachment through a backed-off model. In *Proceedings of the Third Workshop on Very Large Corpora*, Cambridge, MA, 1995.

[9] M. Craven and J. Kumlien. Constructing biological knowledge bases by extracting information from text sources. *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB-99)*, 1999.

[10] Matthew Crocker and Thorsten Brants. Wide coverage probabilistic sentence processing. *Journal of Psycholinguistic Research*, 29(6):647–669, 2000.

[11] Richard Crouch, Ronald M. Kaplan, Tracy H. King, and Stefan Riezler. A comparison of evaluation metrics for broad-coverage stochastic parsers. In *Beyond PARSEVAL workshop at 3rd Int. Conference on Language Resources an Evaluation (LREC'02)*, Las Palmas, 2002.

[12] C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky. GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, 17(1):S74–S82, 2001.

[13] R. Gaizauskas, G. Demetriou, P. J. Artymiuk, and Willett P. Protein Structures and Information Extraction from Biological Texts: The PASTA System. *Bioinformatics*, 19:135–143, 2003.

[14] J.D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(1):180–182, 2003.

[15] Dekang Lin. A dependency-based method for evaluating broad-coverage parsers. In *Proceedings of IJCAI-95*, Montreal, 1995.

[16] Andrei Mikheev. Automatic rule induction for unknown word guessing. *Computational Linguistics*, 23(3):405–423, 1997.

[17] T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi. Automated extraction of information on protein-protein interactions from the biological literature. *Bioinformatics*, 17(2):155–161, 2001.

[18] J. Pustejovsky, J. Castaño, J. Zhang, B. Cochran, and M. Kotecki. Robust relational parsing over biomedical literature: Extracting inhibit relations. In *Pacific Symposium on Biocomputing*, 2002.

[19] Fabio Rinaldi, Gerold Schneider, Kaarel Kaljurand, James Dowdall, Christos Andronis, Andreas Persidis, and Ourania Konstanti. Mining relations in the genia corpus. In Tobias Scheffer, editor, *Accepted for publication in: Second European Workshop on Data Mining and Text Mining for Bioinformatics*. ECML/PKDD, September 2004.

[20] Gerold Schneider. Extracting and using trace-free Functional Dependencies from the Penn Treebank to reduce parsing complexity. In *Proceedings of Treebanks and Linguistic Theories (TLT) 2003*, Växjö, Sweden, 2003.

[21] Gerold Schneider, James Dowdall, and Fabio Rinaldi. A robust and deep-linguistic theory applied to large-scale parsing. In *Coling 2004 Workshop on Robust Methods in the Analysis of Natural Language Data (ROMAND 2004)*, Geneva, Switzerland, August 2004, 2004.

[22] Akame Yakushiji, Yuka Tateisi, Yusuke Myao, and Jun'ichi Tsujii. Building the GENIA Dependency Grammar Treebank of BioMedical documents. In *Proceedings of ACL04, poster session*, 2004.

Paraphrasing Treebanks for Stochastic Realization Ranking

Erik Velldal[♣], Stephan Oepen[♣♠], Dan Flickinger[♠]

[♣] Department of Linguistics, University of Oslo (Norway)
[♠] Center for the Study of Language and Information, Stanford (USA)

1 Introduction

This paper[1] describes a novel approach to the task of *realization ranking*, i.e. the choice among competing paraphrases for a given input semantics, as produced by a generation system. We also introduce a notion of *symmetric treebanks*, which we define as the combination of (a) a set of pairings of surface forms and associated semantics *plus* (b) the sets of alternative analyses for the surface form and sets of alternate realizations of the semantics. For inclusion of alternate analyses and realizations in the symmetric treebank, we propose to make the underlying linguistic theory *explicit* and *operational*, viz. in the form of a broad-coverage computational grammar. Extending earlier work on grammar-based treebanks in the Redwoods (Oepen et al. [13]) paradigm, we present a fully automated procedure to produce a symmetric treebank from existing resources. To evaluate the utility of an initial (albeit smallish) such 'expanded' treebank, we report on experimental results for training stochastic discriminative models for the realization ranking task.

Our work is set within the context of a Norwegian–English machine translation project (LOGON; Oepen et al. [11]). The LOGON system builds on a relatively conventional semantic transfer architecture—based on Minimal Recursion Semantics (MRS; Copestake et al. [5])—and quite generally aims to combine a 'deep' linguistic backbone with stochastic processes for ambiguity management and improved robustness. In this paper we focus on the isolated subtask of ranking the output of the target language generator.

[1] We would like to thank Mark Johnson (Brown University) and Rob Malouf (San Diego State University) for many fruitful discussions and their comments on earlier drafts and ideas presented in this paper.

remember that dogs must be on a leash
remember dogs must be on a leash
on a leash remember that dogs must be
on a leash remember dogs must be
a leash remember that dogs must be on
a leash remember dogs must be on
dogs remember must be on a leash

if you come with the morning boat you can start the trip the same day.
if you come with the morning boat the trip you can start the same day.
if you come with the morning boat the same day you can start the trip.
the trip you can start the same day if you come with the morning boat.
you can start the trip the same day if you come with the morning boat.
the same day you can start the trip if you come with the morning boat.

Figure 1: Example sets of generator outputs using the LinGO ERG. Unless the input semantics is specified for aspects of information structure (e.g. requresting foregrounding of a specific entity), paraphrases will include all grammatically legitimate topicalizations. Other sources of generator ambiguity include, for example, the optionality of complementizers and relative pronouns, permutation of (intersective) modifiers, and lexical and orthographic alternations.

For target language realization, LOGON uses the LinGO English Resource Grammar (ERG; Flickinger [6]) and LKB generator, a lexically-driven chart generator that accepts MRS-style input semantics (Carroll et al. [2]). Over a representative LOGON data set, the generator already produces an average of 45 English realizations per input MRS; see Figure 1 for an example. As we expect to move to generation from packed, ambiguous transfer outputs, the degree of generator ambiguity will further increase. It is therefore essential for end-to-end MT to have a scalable means of ranking generator outputs and ultimately selecting one (or a few) preferred realizations.

In this paper we explore the use of discriminative log-linear models, or maximum entropy models, for ranking the realizations and propose an extended and symmetric notion of treebanks for the supervised learning task. This means that we treat the optimality relation encoded in each treebanked $\langle \mathit{utterance}, \mathit{analysis} \rangle$ pair as being *bidirectional*, and use the underlying grammar to generate all of their possible paraphrases. This provides us with all the admissible realizations for a set of input semantics, each accompanied with an indication of the preferred candidate(s).[2]

The next section further elaborates on the problem of realization ranking as well as the issue of symmetrizing and extending the treebank data. In section 4 we describe the log-linear models that are trained using structural features of the paraphrase data. We also compare the performance of the log-linear models to that of a simple n-gram language model, as well as to a hybrid model that combines the two. The experiments using the language models are described in section 3.

2 Bidirectionality of Treebank Data

Our perspective on the task of realization ranking is given by recognizing its similarity to the task of *parse selection*. As described below, selecting among the analyses delivered by a parser can be seen as the inverse of task of the realization ranking. The results reported by Oepen et al. [13] on the construction of the HPSG Redwoods treebanks and associated parse selection models provides us with a starting point in this respect, both in terms of the methodology used and available data sets.

When training a model for the task of parse selection (i.e. choosing among competing analyses of a token utterance), the distribution that one is typically interested in is the conditional probability of an analysis given a string. Moreover, this typically requires training data that consists of all possible analyses for a set of strings, where the goal is to estimate a distribution that for each string maximizes the probability of the preferred analysis over all the other competing candidates.

For the task of realization ranking (i.e. choosing among multiple paraphrases of a meaning representation input to the generator), on the other hand, we are interested in a different distribution. In order to select the best realization(s) we need a model that gives us the probability of a string given its semantics.

A treebank is traditionally conceived as a set of utterances (typically strings) paired with their optimal or correct analyses. In this paper we take the optimality relation that these pairs encode to be *bidirectional* or *symmetric*, in the sense that the original utterance can also be treated as an optimal realization of the corresponding semantic analysis (i.e. ‘meaning’). The remaining part of this section looks at how we can exploit this bidirectionality or symmetry of the recorded $\langle \textit{utterance}, \textit{analysis} \rangle$ pairs to construct all the possible paraphrases for the treebanked items. This will provide us with the necessary training data to learn the

[2] The utility of this kind of resource is by no means restricted to our MT setting, but should prove relevant for other applications that generates from semantic representations. Furthermore, the ability to generate paraphrases of a given input seems potentially beneficial to other tasks too, as, for example, question-answering (QA) and summarization systems.

discriminative models described in section 4.

The Redwoods treebank[3] is a collection of HPSG analyses derived from the LinGO ERG for various domain corpora (e.g. transcribed scheduling dialogues, ecommerce email, and lately tourism text), with manual annotation to identify the intended parse(s) for each utterance. Since (a) the Redwoods treebank provides a full HPSG sign for each input item and (b) the ERG analyses incorporate an MRS-style semantic component into HPSG, we have the option of using the semantics associated with each preferred analysis for subsequent generation.

Note that the Redwoods approach to treebanking—viz. the construction of the treebank by virtue of selecting among the analyses provided by a broad-coverage computational grammar—already includes alternate ('competing' but dis-preferred) analyses for a token utterance as first class data. While this extension to a conventional conception of treebanks as only providing the 'optimal' $\langle \textit{utterance}, \textit{analysis} \rangle$ pairs clearly benefits stochastic parse selection research, it would seem possible in theory at least that statistical parsing work—aiming to induce a grammar from the treebanked analyses, rather than using the ERG—could derive value from the additional data. Likewise, viewing a treebank as a repository of linguistic information, the availability of dis-preferred analyses might turn out useful to researchers in (formal) grammar or linguistic students. The proposal of the current paper is to further augment the treebank by the inverse correspondence: with regard to the explicit linguistic model underlying the treebank (i.e. the grammar used to build it), the paraphrase step aims to provide the mirror image of the dis-preferred analyses, this time making alternative but sub-optimal realizations first class data to be included in the treebank.

The actual procedure is straightforward. Given a Redwoods treebank, for each analysis that has been manually marked as the correct reading, we exhaustively generate all possible realizations for its semantics. In other words, for each string (and its hand-annotated intended meaning) in the original treebank, all semantically equivalent paraphrases admitted by the grammar are generated.

The next step is to automatically identify and mark the preferred realization(s). This is done by matching the *yields* of the generated trees against the original *strings* in the parse treebank, where all derivations yielding this preferred surface string are counted as equally good candidates. We now have a data set that includes all possible paraphrases for each treebanked semantic representation, with the best realization(s) marked. Note that, the *grammaticality* of all the candidates is guaranteed by the generator with respect to the input grammar. Furthermore, the fact

[3] See '`http://redwoods.stanford.edu/`' for further information on the Redwoods initiative and access to the data available to date. The Redwoods treebank is available under an open-source license and currently comprises some 15,000 annotated utterances.

Aggregate	**items** $\sharp$	**words** ϕ	**readings** ϕ
$100 \leq$ *readings*	19	19.7	422.9
$50 \leq$ *readings* < 100	17	17.8	71.7
$10 \leq$ *readings* < 50	72	13.7	22.6
$1 <$ *readings* < 10	153	10.4	4.8
Total	**261**	**12.4**	**44.5**

Table 1: Some core metrics for the symmetric treebank data used in our initial experiments, broken down by degrees of ambiguity in generation. The columns are, from left to right, the subdivision of the data according to the number of realizations, total number of items scored (excluding items with only one realization), average string length, and average structural ambiguity.

that the Redwoods approach provides a treebank that is built on a grammar (and not the other way around) means that our data can be dynamically updated to reflect developments and improvements as the grammar is revised, i.e. as the grammar evolves there is a semi-automated procedure to (re-)synchronize the treebank with a new set of analyses provided by the grammar (see Oepen et al. [12] for details).

As mentioned initially, the strategy described here for utilizing treebanks comes with the underlying assumption that, without introducing too much distortion, the original string associated with a given reference analysis (in the 'parse treebank'), can also reasonably be taken to be an optimal way of expressing the corresponding meaning. After all, by the fact that a sentence is observed to occur naturally in our semantically annotated corpus (i.e. treebank), we are in some sense already making the assumption that a presumably rational and competent language user chose this very utterance to express the given semantics. Granting real language users some authority when it comes to formulating expressions that in an effective and natural-sounding way convey the meaning that they want to communicate, this does not seem like a too radical proposition.

2.1 Data and Evaluation

In the following we report on a preliminary investigation into the utility of such a set of paraphrases in a symmetric treebank for some 300 sentences from the LOGON domain—hiking instructions in Norway. The (relatively small) symmetric treebank data that we had available for these initial experiments is summarized in Table 1. Although the total number of items in the treebank is above 300, when reporting results for our realization ranking experiments we exclude items that are unambiguous in generation and hence do not present a realization ranking problem.

Before we in section 4 go into the details of using log-linear models trained using structural features of the paraphrased treebank data, section 3 reports on some experiments that take a purely surface-oriented approach to the ranking task. The simple n-gram model presented here will not only serve as a baseline for experimental results obtained for the initial log-linear model, but will also be incorporated as one of the features in a final combined model.

All models are evaluated according to two different measures; exact match accuracy and the similarity-based BLEU score (Papineni et al. [15]). The exact match measure simply counts the number of top-ranked sentences, according to some model, that exactly matches a corresponding "gold" or reference sentence. In other words, after a model has been applied to all possible paraphrases in the symmetric treebank, we count the number of times that the model assigns the best score to (one of) the string(s) marked as preferred in the symmetric treebank. The similarity-based and less rigid BLEU measure has gained a well-established role as an evaluation metric in MT, and is modeled after the *word error rate* measure used in speech recognition. The score is computed as a weighted average of the n-gram precision of the selected candidate realization with respect to the reference, for all $1 \leq n \leq 4$. When evaluating a models performance on the test data, we report the averaged BLEU scores over all realizations ranked best by the model, which has a constant range in $[0, 1]$. Although it may be hard to intuitively interpret this precision-based measure in isolation, it at least offers an alternative view when comparing the relative performance of the various models that we now turn to describe. For more information on the BLEU scoring metric, see Papineni et al. [15].

3 N-Gram Language Models

As a first shot at ranking the generator outputs, we order the English target strings with respect to the probabilities assigned by a simple n-gram language model.

An n-gram model relies on the Markov assumption that the probability of a given word only depends on the $n - 1$ words preceding it, and so the probability of a sequence $(w_1, \ldots, w_k)$ is computed as

$$(1) \qquad p_n(w_1, \ldots, w_k) = \prod_{i=1}^{k} p(w_i | w_{i-n}, \ldots, w_{i-1})$$

In MT applications, the idea of choosing the most fluent string as the best translation is a commonly used technique (see, among others, Langkilde and Knight [8], and Callison-Burch and Flournoy [1]). Using the CMU-SLM Toolkit (Clarkson and Rosenfeld [4]), various n-gram language models were trained on a plain (unannotated) text version of the British National Corpus (BNC), containing approximately 100 million words.

For the experiments reported in this paper, we use 4-gram model trained with Witten-Bell discounting, a vocabulary of 65,000 words, sentence boundary context cues, and using back-off to lower-order models for unobserved n-grams. When applying the language model to the ranking task we obtain close to fifty per cent exact match accuracy (see Table 2). This result improves significantly over a seventeen per cent random choice baseline. The same holds for the similarity-based evaluation in Table 3.

We also tried ranking the realizations by their cross-entropy or perplexity with respect to the language model. Of course, this setting is quite far from the typical model evaluation setting in which perplexity scores are computed (as an asymptotic approximation) over an entire test corpus for assessing the quality of a model. On the sentence level the usual approximation to the perplexity essentially indicates the average log probability of the words:

(2) $$2^{-\frac{1}{k}\log p_n(w_1,\ldots,w_k)}$$

Using these scores, however, gave somewhat inferior performance compared to using the (negative log) probabilities directly. Furthermore, we also saw that increasing the value of n, as well as increasing the vocabulary size, always lead to better performance in our ranking task, although at the expense of larger models.

The basic underlying assumption of the approach described in this section, is that the best realization of the input semantics corresponds to the most fluent string. This implies that we rank outputs as isolated strings rather than as realizations of a given semantic representation. Another obvious limitation inherent to the simple n-gram approach described here is the fact that it cannot capture long-range dependencies.

4 Log-Linear Models

Taking inspiration from contemporary parse selection work, we here describe conditional log-linear models that take into account structural features of competing realizations for a given input MRS. The family of maximum entropy models or log-linear models provides a general framework that allows one to combine disparate and overlapping sources of information in a single model without making unwarranted independence assumptions. A model is given in terms of *specified feature functions* describing the data points, and an associated set of *learned weights* that determine the contribution or importance of each feature. Each event—in our case a realization $r \in \Omega$—is mapped to a feature vector $f(r) \in \Re^d$, and a vector of weights $\lambda \in \Re^d$ is then fitted to optimize some objective function. A conditional log-linear model for the probability of a realization r given the semantics s, has the form

$$p_\lambda(r|s) = \frac{1}{Z(s)} \exp(\lambda \cdot f(r)) \tag{3}$$

where $Z(s)$ is a normalization factor defined as

$$Z(s) = \sum_{r' \in Y(s)} \exp(\lambda \cdot f(r')) \tag{4}$$

When computing the so-called partition function $Z(s)$ as in equation (4) above, $Y(s)$ gives the set of all possible realizations of s. The weight vector λ is chosen as to maximize the (log of) a penalized likelihood function as in

$$\hat{\lambda} = \arg\max_{\lambda} \log L(\lambda) - \frac{\sum_{i=1}^{d} \lambda_i^2}{2\sigma^2} \tag{5}$$

where $L(\lambda)$ is the pseudo-likelihood of the training data (as described by Johnson et al. [7]), computed as

$$L(\lambda) = \prod_{i=1}^{N} p_\lambda(r_i|s_i) \tag{6}$$

In accordance with current best practice, the second term of the objective function in (5) defines a zero mean Gaussian prior on the weight parameters (Chen and Rosenfeld [3], Johnson et al. [7], Malouf and van Noord [10]). By promoting less extreme parameter values this penalty term can reduce the tendency of log-linear models to over-fit the training data. In addition to improving accuracy, this kind of smoothing tends to also reduce the number of iterations needed for convergence during estimation (Malouf and van Noord [10]). We empirically determined a suitable value for the variance σ^2 which is uniformly set to 100 for the results reported here. Note that the value of the variance parameter determines the relative contribution of the prior and the likelihood function, and thereby the degree of smoothing (Malouf and van Noord [10]).

For the parse selection task, Toutanova and Manning [16] train a discriminative log-linear model with features defined over ERG *derivation trees*, where labels identify specific *construction types* and fine-grained *lexical classes*. For our own initial experiments with the realization ranking we define the feature set in the same way (the basic PCFG-S model of Toutanova and Manning [16]), using the `estimate` open-source package (Malouf [9]) for parameter estimation (using the

Aggregate	**random**	**n-gram**	**MaxEnt**	**combined**
	%	%	%	%
$100 \leq$ *readings*	0.4	10.5	21.1	31.6
$50 \leq$ *readings* < 100	1.5	17.7	22.1	29.4
$10 \leq$ *readings* < 50	5.4	38.2	31.6	37.5
$1 <$ *readings* < 10	27.0	62.8	68.2	75.8
Total	**17.2**	**49.2**	**51.7**	**59.0**

Table 2: Realization ranking accuracies for a random-choice baseline model, 4-gram language model, simple conditional model, and combination of the two. The columns are, from left to right, the subdivision of the data according to degrees of ambiguity, followed by exact match accuracies for the four models.

Aggregate	**random**	**n-gram**	**MaxEnt**	**combined**
	%	%	%	%
$100 \leq$ *readings*	0.5869	0.7209	0.7023	0.7534
$50 \leq$ *readings* < 100	0.5956	0.7751	0.7939	0.7790
$10 \leq$ *readings* < 50	0.6418	0.7985	0.8293	0.8172
$1 <$ *readings* < 10	0.7382	0.8792	0.9178	0.9316
Total	**0.6913**	**0.8386**	**0.8696**	**0.8771**

Table 3: Realization similarity measures for a random-choice baseline model, 4-gram language model, simple conditional model, and combination of the two. The columns are, from left to right, the subdivision of the data according to degrees of ambiguity, followed by averaged BLEU scores of the realization(s) ranked best by each of the four models.

limited-memory variable metric). With only around 300 training sentences in our current generation treebank and ten-fold cross validation (which tends to underestimate model performance), this simplest of log-linear models performs competitively to the language model trained on the BNC (see Tables 2 and 3).

As a third model we augmented the log-linear model with an extra feature corresponding to the sentence probabilities of the language model (described in section 3). The value of the $d + 1$'th feature is the (negative log) probability of the string as given by the n-gram model p_n, i.e. $f_{d+1}(r) = -log\ p_n(y(r))$, where $y(r)$ is the yield of r and $n = 4$ as before. Unsurprisingly, the combined model significantly outperforms both the previously described models.

5 Discussion and Outlook

For the relatively coherent LOGON domain at least, a tiny training set of some 300 automatically 'annotated' paraphrases combined with a discriminative baseline model originally proposed for the parse selection task outperforms a language model trained on all of the BNC. Our results suggest that this use of domain-specific treebanks—and the underlying assumption of relative 'naturalness' of the original, corpus-attested realizations—provide a good handle on ranking generator outputs, and that structural, linguistic information as is available to the log-linear model is of central importance for this task. We are currently extending the size of the available treebank for the LOGON domain (to some 1,500 utterances initially and ultimately to at least 5,000 annotated items) and expect that the larger training set—combined with more systematic experimentation with discriminative models using larger and more specialized feature sets—should allow us to improve exact match accuracy significantly, ideally to around eighty per cent exact match as are the currently best available parse selection results.

Additionally, we plan to formalize a notion of graded acceptability of competing realizations (based on string similarity metrics, e.g. BLEU or string kernels) and refine both model training and evaluation in this respect. Unlike in parse selection—where distinct system outputs typically have distinct semantics—in realization ranking there is more of a graded continuum of more or less natural verbalizations (given available information). All outputs are guaranteed by the grammar to be semantically equivalent and grammatically well-formed. This means that the kind properties we aim at capturing with the discriminative model rather are soft constraints that govern the graded degree preference among the competing paraphrases. The approach described by Osborne [14] and Malouf and van Noord [10] for scoring the training instances (parses in both cases) according to some measure of preference, and defining the empirical distributions based on these weights, seems like a well-suited approach for dealing with generation outputs too, where the notion of correctness may be inherently fleeting.

To investigate the degree of domain-specificity in stochastic models derived from Redwoods-style symmetric treebanks, we plan to automatically paraphrase additional segments of the available Redwoods treebank and perform cross-domain realization ranking experiments.

References

[1] Chris Callison-Burch and Raymond S. Flournoy. A program for automatically selecting the best output from multiple machine translation engines. In *Proceedings of the MT Summit*, Santiago, Spain, September 2001.

[2] John Carroll, Ann Copestake, Daniel Flickinger, and Victor Poznanski. An efficient chart generator for (semi-)lexicalist grammars. In *Proceedings of the 7th European Workshop on Natural Language Generation*, pages 86 – 95, Toulouse, France, 1999.

[3] Stanley F. Chen and Ronald Rosenfeld. A Gaussian prior for smoothing maximum entropy models, 1999. Technical Report CMUCS-CS-99-108.

[4] Philip Clarkson and Roni Rosenfeld. Statistical language modeling using the CMU-Cambridge Toolkit. In *Proceedings of ESCA Eurospeech*, 1997.

[5] Ann Copestake, Dan Flickinger, Rob Malouf, Susanne Riehemann, and Ivan Sag. Translation using minimal recursion semantics. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, Leuven, Belgium, 1995.

[6] Dan Flickinger. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6 (1) (Special Issue on Efficient Processing with HPSG):15 – 28, 2000.

[7] Mark Johnson, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. Estimators for stochastic 'unification-based' grammars. In *Proceedings of the 37th Meeting of the Association for Computational Linguistics*, pages 535 – 541, College Park, MD, 1999.

[8] Irene Langkilde and Kevin Knight. The practical value of n-grams in generation. In *International Natural Language Generation Workshop*, 1998.

[9] Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th Conference on Natural Language Learning*, Taipei, Taiwan, 2002.

[10] Robert Malouf and Gertjan van Noord. Wide coverage parsing with stochastic attribute value grammars. In *Proceedings of the IJCNLP workshop Beyond Shallow Analysis*, Hainan, China, 2004.

[11] Stephan Oepen, Helge Dyvik, Jan Tore Lønning, Erik Velldal, Dorothee Beermann, John Carroll, Dan Flickinger, Lars Hellan, Janne Bondi Johannessen, Paul Meurer, Torbjørn Nordgård, and Victoria Rosén. Som å kappete med trollet? Towards MRS-based Norwegian – English Machine Translation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*, Baltimore, MD, 2004.

[12] Stephan Oepen, Dan Flickinger, and Francis Bond. Towards holistic grammar engineering and testing. Grafting treebank maintenance into the grammar revision cycle. In *Proceedings of the IJCNLP workshop Beyond Shallow Analysis*, Hainan, China, 2004.

[13] Stephan Oepen, Kristina Toutanova, Stuart Shieber, Chris Manning, Dan Flickinger, and Thorsten Brants. The LinGO Redwoods treebank. Motivation and preliminary applications. In *Proceedings of the 19th International Conference on Computational Linguistics*, Taipei, Taiwan, 2002.

[14] Miles Osborne. Estimation of stochastic attribute-value grammars using an informative sample. In *Proceedings of the 18th International Conference on Computational Linguistics*, Saarbrücken, Germany, 2000.

[15] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu. A method for automatic evaluation of Machine Translation. In *Proceedings of the 40th Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA, 2002.

[16] Kristina Toutanova and Christopher D. Manning. Feature selection for a rich HPSG grammar using decision trees. In *Proceedings of the 6th Conference on Natural Language Learning*, Taipei, Taiwan, 2002.

An Annotation of What Is Not There: Empty Arguments and Cross-Clausal Reference in Spoken and Written Tibetan Texts

Bettina Zeisler

SFB 441, Universität Tübingen
zeis@uni-tuebingen.de

1 Introduction: purpose of the annotation

Like in many other Asian languages, speakers of Tibetan languages avoid to express information that is already given or can be derived by the hearer from the context. Practically, the verb is the only obligatory constituent of a clause, while all nominal constituents can be deleted, as in the second clause of (1).

(1) *khoŋ gñiskas de žiŋpola sabon btabste ø ø rmos* | (Francke [8]: 1)
The two sowed the seeds on the field and ploughed [it].

In particular, there is no syntactic restriction for the deletion of arguments. While the deleted argument in (2) could only be a subject (agent) in a syntactically accusative language and only a patient (direct object) in a syntactically ergative language, in Tibetan this sentence

(2) *žiŋbatpas yokpo kolte ø žiŋ rhmos.* (Field data 1996)
The farmer hired a servant and ø ploughed the field.

can be interpreted in three ways: either the farmer or the worker or both ploughed. A competent speaker resolves the ambiguity mainly along prag-matic considerations (eg why should a farmer hire a worker if not for ploughing). By contrast, students with the background of a European language have considerable problems to find out who is acting upon what. Nevertheless, the phenomenon has not received much attention in Tibeto-Linguistics. Preliminary reflections can be found in Andersen [1] and Zeisler [16]. Earlier, Zimmermann [19] had pointed to the fact that a patient often becomes the subject of a following intransitive clause in Classical Tibetan. A similar relation was termed 'ergative coreference' by Cooreman et al. [4], but their classification for Indonesian does not account for the reference relation in sentence (2) where a patient is continued as an agent of a 'transitive' clause.

Our research project aims at the formulation of rules or at least statistical preferences for the identification of antecedents of empty arguments for different Tibetan varieties. We expect different frequencies of deletion according to the saliency of an argument corresponding to its semantic role and its position in the verb frame, but we also expect changes in these frequencies diachronically. To provide an empirical basis for this research, we are building a syntactically annotated corpus of written and spoken texts from different periods: Old Tibetan (8th–10th century), Classical Tibetan (11th–19th century), and contemporary West Tibetan as spoken in Ladakh (India).

2 General design of the annotation

All texts are annotated in XML. The annotation is done semi-automatically (by incrementally using chunks of already annotated text, see Wagner and Zeisler [15]) and provides rich syntactic information about phrasal and textual structures as well as information about the argument structure of each verb and the realisation or non-realisation of the arguments in the text. The information is encoded as embedded annotation (ie the markup is placed locally at or around the corresponding text).

The basic unit to be annotated is a clause, which typically contains a verb and possibly other elements. The internal structure of a clause is represented as an XML hierarchy tree. One or more clauses constitute a sentence. Higher textual units, tagged as divisions (<div>), consist of a number of sentences. Hence, sub-clausal and super-clausal hierarchies do not overlap and can be captured within a single document hierarchy. Concurrent hierarchical units occur only marginally and are not of primary importance; such boundaries are marked by empty XML elements (eg <pb/> for a page break), which do not violate the well-formedness of the document.

The lowest level of annotation marks the tokens (ie 'words') (<tok>) with their orthographic realisation (<orth>) and part-of-speech classification (<pos>). The phrase level is encoded by <ntNode> (non-terminal node) elements. An <ntNode> spans an inflectional group: a noun phrase if this group forms an argument and an adverbial phrase otherwise. This distinction is marked by an element <ntNodeCat>, which contains the category NP or AvP, respectively. A clause (<clause>) encompasses a verb token (always at final position), associated arguments or adverbials, and, if present, embedded clauses. Participle clauses may also be part of an <ntNode>. An element <clauseCat> specifies the type of the clause (simple, chained, embedded, etc). Tokens, phrases, and clauses may receive a further linguistic description (<desc>) specifying the case for sub-clausal phrases and the reference for anaphoric elements. For verb tokens, the corresponding argument structure and its realisation in the text is encoded. Above the clause level, sentences (<s>) are marked. The annotation of the textual level specifies discourse units such as direct or indirect speech, poems or songs, and text segments.

3 Annotation of empty arguments

It would be in contradiction with a naturalistic representation of the clause structure to assign slots within a clause for the representation of what is not there. Furthermore, due to a comparatively free word order, we cannot predict where such slots should be placed. Neutral word order is SOV, or more precisely: Agent > Recipient > Patient > Location > Verb and similarly for inagentive verbs.[1] The verb always comes last, but the order of the arguments can be changed for topicalisation or focussing. We thus chose to represent empty arguments in a second layer: the argument structure of the verbs.

Each verb token receives a serial ID number as an attribute and a special description of its subcategorisation frame (within <desc>). This description comprises (a) the 'canonical' argument structure as listed in the lexicon (a list of <complement> elements within a <frame> element), and (b) the 'real' frame, ie the realisation of the arguments in the clause (a list of <realComplement> elements within a <realFrame> element). For each canonical and real complement, a <role> element is specified, and each canonical complement receives a specification of its case, as does each real complement whose case deviates from the canonical assignment. The redundant structure is motivated by a higher convenience for annotation, XSLT transformations, and queries. Since the frame is to be found identically in the verb lexica accompanying each text, we might delete it from the final annotation as soon as the corresponding lexicon is linked with the annotation.

To encode cross-clausal reference, each <realComplement> receives an ID based on the verb number. The order of the arguments in the <realFrame> corresponds to the order of the 'canonical' frame. Inverted word order is in-dicated by an attribute on the <realFrame>. Empty arguments receive an attribute marking emptiness and a pointer to the antecedent in the text, which in most cases will be a <realComplement> specified in the argument structure of some previ-

[1] Tibetan is commonly classified as ergative language (cf Tournadre [14] for modern Central Tibetan), because the (semantic) subject of an intransitive sentence receives the same (zero) case marking as the direct object of the transitive sentence, whereas the (semantic) subject of a transitive sentence receives a different (overt) case marker. For Tibetan this definition poses quite a few problems (Zeisler [18]).

ous clause. In general, a pointer is encoded as a reference tag (<ref>) with an attribute 'target' that points to the ID number of the corresponding referee, see example (3), the annotated version of (1). In case, no antecedent can be located, the reference tag will remain without target attribute, but might contain additional contextual information. Reference tags are also used for all other anaphoric elements, such as personal (line 9) and demonstrative pronouns (line 29) or referential adjectives.

(3) *khoŋ gñis·kas de žiŋ·po·la sa·bon btab·ste* (v13) *rmos* | (v14)

```
        <s>
        ...
          <clause>
            <ntNode>
5             <tok>
                <orth>khoŋ</orth>
                <pos>PRON3:pl</pos>
                <desc>
                  <ref target="v10c1"></ref>
10              </desc>
              </tok>
              <tok>
                <orth>gñis·kas</orth>
                <pos>NUM</pos>
15            </tok>
              <ntNodeCat>NP</ntNodeCat>
              <desc>
                <case>Erg</case></desc>
            </ntNode>
20          <tok>
              <orth>yaŋ</orth>
              <pos>CONJ</pos>
            </tok>
            <ntNode>
25            <tok>
                <orth>de</orth>
                <pos>DEMfar</pos>
                <desc>
                  <ref target="v10c3" type="precDEMfar"></ref>
30              </desc>
              </tok>
              <tok>
                <orth>žiŋ·po·la</orth>
                <pos>NOM:inan-df</pos>
```

```
   </tok>
   <ntNodeCat>NP</ntNodeCat>
   <desc>
      <case>DatLoc</case>
   </desc>
</ntNode>
<ntNode>
   <tok>
      <orth>sa·bon</orth>
      <pos>NOM:inan</pos>
   </tok>
   <ntNodeCat>NP-inc</ntNodeCat>
   <desc>
      <case>Abs</case>
   </desc>
</ntNode>
<tok id="v13">
   <orth n="2">btab·ste</orth>
   <pos>V</pos>
   <desc>
      <frame>
         <complement>
            <role>eA3</role>
            <case>Erg</case>
         </complement>
         <complement status="omissible">
            <role>DIR</role>
            <case>~Loc</case>
         </complement>
         <complement status="inc">
            <role>P3</role>
            <case>Abs</case>
         </complement>
      </frame>
      <realFrame>
         <realComplement id="v13c1">
            <role>eA3</role>
         </realComplement>
         <realComplement id="v13c2">
            <role>DIR</role>
         </realComplement>
         <realComplement status="inc" id="v13c3">
            <role>P3</role>
         </realComplement>
```

```
                  </realFrame>
80             </desc>
            </tok>
            <clauseCat>chained</clauseCat>
         </clause>
         <clause>
85          <tok id="v14">
               <orth>rmos</orth>
               <pos>VFIN</pos>
               <desc>
                  <frame>
                     <complement>
90                      <role>eA2</role>
                        <case>Erg</case>
                     </complement>
                     <complement>
95                      <role>P2</role>
                        <case>Abs</case>
                     </complement>
                  </frame>
                  <realFrame>
100                  <realComplement status="empty" id="v14c1">
                        <role>eA2</role>
                        <ref target="v13c1"></ref>
                     </realComplement>
                     <realComplement status="empty" id="v14c2">
105                     <role>P2</role>
                        <ref target="v13c2"></ref>
                     </realComplement>
                  </realFrame>
               </desc>
110         </tok>
            <clauseCat>endchain</clauseCat>
         </clause>
         <punct>|</punct>
      </s>
```

In clause 14 (starting with line 84), the verb 'plough' canonically selects two arguments, one with *ergative* case for the *agent* (eA2) and one with *absolutive* case for the *patient* (P2). None of the arguments is realised in the clause, and each realised counterpart can be found exactly in the preceding clause. Therefore, a reference tag with the attribute: target="v13c1" (line 102) points from the first empty <realComplement> (id="v14c1") to its antecedent, the first

<realComplement> of the previous clause (line 70). A second reference tag with the attribute: target= "v13c2" (line 106) points from the second empty <realComplement> (id="v14c2") to the second <realComplement> of the previous clause (line 73) as its antecedent.

The detailed annotation of argument structure and cross-clausal reference allows to evaluate the distances between empty arguments and their antecedents and to observe the changes of role and/or case in chains of corresponding empty arguments referring to the same antecedent. While the target attribute has to be set manually (via a selection tool), the implicit information about reference distance, intervening textual boundaries, and changes in role or case can be retrieved automatically and made explicit via XSLT transformations. The information is stored in a list of antecedents (<antecedList>) for each empty argument or other anaphoric element and a list of anaphors (<anaphList>) for each element that has been referred to. The number of entries in the list is indicated by an attribute 'size'. Each element <antecedent> or <anaphor> contains the pointer (<ref>), the specification of the reference type <refType>, the source of reference, ie its complement ID <refCompID>, various tags counting intervening textual boundaries, and two tags specifying the matching or mismatching between antecedent and anaphoric element with respect to role and case, see example (4) for the list of anaphora. The list of antecedents has the same structure.

```
(4) <realFrame>
      ...
      <realComplement id="v13c2">
         <role>DIR</role>
         <anaphList size="6">
            <anaphor>
               <ref target="v13c2"></ref>
               <refType>empty</refType>
               <refCompId>v14c2</refCompId>
               <clauseDist>1</clauseDist>
               <sDist>0</sDist>
               <qDist>0</qDist>
               <seqDist>0</seqDist>
               <finDist>0</finDist>
               <divDist>0</divDist>
               <roleMatch congr="n">
                  <antRole>DIR</antRole>
                  <refRole>P2</refRole>
               </roleMatch>
               <caseMatch congr="n">
                  <antCase>~Loc</antCase>
                  <refCase>Abs</refCase>
            </caseMatch>
          </anaphor>
          ...
        </anaphList>
      </realComplement>
      ...
    </realFrame>
```

From this redundant information we can now easily derive detailed statistics, eg concerning the role relation between empty argument and antecedent, its total frequency and its frequency with respect to the clause distance. While the overall picture derived by this process corresponds fairly well to our expectations, the surprises lie in the detail. Eg we expected that active or neutral coreference would be dominant in Ladakhi. Accordingly, our annotation yields a preference for congruent reference relations (31%) and agency (all agent-agent relations independent of valency >50%). One would also expect that reference relations depend on the reference distance, thus agency relations would become even more prominent over long distances.

What we did not expect, however, was that neutral coreference between two patients would be the preferred reference relation (23%) within a distance of only one clause, followed by active (two-place←one-place: 9%, one-place←two-place: 7%) and neutral agentive coreference (two-place: 6%). Since clause chains with agent continuity are much more common than such with patient continuity, chains of the type *X cut an apple and ø ate ø* must be typically embedded in longer chains with a deleted agent, thus *X verbed ... and ø cut an apple and ø ate ø (and ...)*. The annotation also proved very helpful to identify marginal reference relations, particularly in complex embedded structures where they are no longer evident. Thus we believed that the typologically not predicted relation between a patient and a transitive (ergative) agent could not be found at all in the Ladakhi Kesar epic (Francke [8]). While this relation is predictably rare and apparently restricted to contexts of social interaction or exchange, we could locate four instances alone in the first chapter (583 clauses out of ca. 8500).

4 Some particularities of Tibetan and their impact on the annotation

The annotation merely serves a tool, although a quite useful one, for our empirical research. The annotation scheme thus is not derived from theoretical reflections or existing models, but has developed and is still developing out of the needs of our project and along with the growing understanding of what exactly we are dealing with. Thus, the most important prerequisites are practicability, a structure that focuses on the essential, transparency and flexibility. Eg in order not to loose the overview over the complex Tibetan clause and phrase structures we chose to specify the node category after the node content in contradiction to the common linguistic practise.

For the same reason, morphological information is encoded on different levels, depending on its transparency, necessity for the understanding of the text, and saliency for the research project. The non-obligatory use of plural or definiteness markers, eg, is integrated into the part-of-speech tag (3), line 34. Verbal polarity items (negation and question markers) and the use of modal auxiliaries are specified with the help of <feature> tags within the description of the verb token.

Tense (and aspect), however remain unspecified. The reason for this seemingly inconsistent decision is simple: there is no agreement among Tibetologists about what exactly is encoded: absolute tense according to the Tibetan grammatical tradition, 'aspect' (certainly not of the slavic type) according to the present mainstream, or relative tense according to a minority view (Zeisler [17]).[2] As for Old and Classical Tibetan, our unknown entity is basically expressed through (up to four) derivational stem forms, which may vary according to the orthographical tradition. We decided to indicate these stem forms in the accompanying verb lexicon, particularly also to be able to compare the spellings of different texts with each other and with the idealised forms of the dictionaries. A mere stem can have a finite or non-finite function, ie it typically terminates a chain of clauses (a final sentence particle may be added, but is not necessary), but it may be also used in particular contexts in place of a non-finite verb form, such as a verbal noun or a converb.

Besides the mere stems, Old and Classical Tibetan also show a great number of finite periphrastic constructions, based on stems, nominalisers, auxiliaries, and additional material. The morphological material itself is functionally intransparent, ie it is not possible to derive the meaning of the whole expression from its parts; it can only be deduced from its usage in discourse. Research into this topic is still missing, and it would be impeded by the fact that Classical Tibetan obviously integrated various regional varieties using different constructions or similar constructions with different meanings. It is practicably impossible to know the exact function of most of these constructions. The situation is different for the modern varieties and thus for Ladakhi, but we decided to keep the same annotation scheme and to concentrate on the more relevant issues.

[2] Tibetan languages are still poorly described. The best resource for Classical Tibetan (although with some conceptional errors) is the textbook of Hahn [9]. For contemporary Ladakhi see Francke [7] and Koshal [10], for Balti Read [12] and Bielmeier [2].

Nevertheless we hope to develop an annotation tool that would allow to add this information automatically.

The most salient feature for the project is the encoding of syntactic-semantic relations between a verb and its arguments. These relations are encoded by what we call 'case markers' and 'postpositions'. Postpositions are typically lexically transparent inflected nouns joined to their head via the genitive case, eg *khaŋpa-ḥi naŋ-la* (house-Genitive interior-Dative/Locative) 'into the house' lit. 'to the interior of the house'. Like the European prepositions, postpositions are mainly used for the more specific spatial relations. Case markers are more evidently integrated into the intonational unit 'word' as most of them show assimilation features, and they express relations that are less semantically transparent. Old and Classical Tibetan has the following morphological markers: *{kyi}* Genitive, *{kyis}* Instrumental, *na* Locative, *la* Dative/ Locative (Allative), *nas* and *las* Ablative, *{tu}* Locative/Purposive (Allative), and *daŋ* Comitative. Ladakhi has only one Ablative marker /nas/~/na/~ /ne/ and only one productive Dative/Locative marker /(l)a/; the Instrumental as a peripheral argument marker has been replaced by the Comitative.

Except for the Genitive, all case markers can be replaced by a corresponding postposition. Therefore, we use variables in the argument structure, but specify case marking and use of postposition in the annotation of the text. We might change the design when we can predict the use of the locative markers and postpositions. According to our approach, different syntactic slots have different functional weight, conditioning the possibilities of variation in the markers or exchange of position. We thus differentiate between subject and non-subject markers, reaching thus at a set of eight case variables: Erg[ative] (=Instrumental), Aes[thetive] (=Dative/ Locative), '~Loc' (ie the functionally overlapping locative or allative case markers and postpositions), Abl[ative], Instr[umental], Com[itative], Gen[itive], and Abs[olutive] or zero-marking.

Generally, we attempt to keep the annotation as flat as possible. Thus we reduce non-terminal nodes to tokens wherever possible: elementary parts of speech such as conjunctions, cf (3), line 22, are annotated as token, not as phrase nodes. Similarly, we do not annotate the argument structure of verbal nouns if they are lexicalised. On the other hand, since Tibetan shows nominal group inflection, ie only the last element of a noun phrase is modified by a case marker or postposition, we annotate non-terminal nodes for each inflectional group, even if it consists of a single noun. But although an inflectional group may show internal subgrouping through relational case marking, either Genitive 'of' or Comitative 'and', we leave the internal structure unspecified, adding only a description for the case marker to the respective token.

Based on the syllabic appearance of the script, Old and Classical Tibetan has traditionally been treated as being monosyllabic. It is true, that many elementary lexemes consist of only one syllable, such as *rta* 'horse', or could be analysed as consisting of one lexemic syllable and one or more derivational morphemes, eg *rtapa* 'rider'. In this case, the morpheme *pa*, typically functioning as a nominaliser with verbs, can be described as a derivational suffix, expressing a relation of possession or attribution thus *rta-pa* 'someone who has a horse as his attribute''. But in the case of *khaŋpa* 'house', there is no meaningful word **khaŋ* and the function of the second element is completely intransparent. Furthermore, Tibetan is very productive in forming compounds such as *rta-mgo* 'horse head', *rta-mchog* 'best of horses', *rta-bskrags* 'clattering (train of horsemen)', etc. Derivations as well as compounds form intonational units that also extend over following syntactic morphemes. Intonational units can be defined as displaying assimilation features (including tone contour in the modern tonal varieties) and prefix dislocation at the lexeme or morpheme boundary, such as Ladakhi /stap-raks/ for the above *rta-bskrags*. Some classical spellings such as *bud-med* 'woman' probably from *bu-dmad* 'of low birth' indicate a certain tendency for treating compounds as lexical units.

Instead of annotating functionally intransparent particles with dummy designations for no other purpose than to unify them with their lexemes to higher units, we are annotating lexical units and (hypothetical) intonational units or 'words' as the basic units (<token>). Following our observations in the spoken language we treat postpositions as being part of the intonational unit (this is also done by Bielmeier [2]), and we likewise treat complex verbal expressions as one 'word', although we might revise this in the future for modal constructions. We cannot avoid that such decisions are sometimes debatable and thus in conflict with our attempt to keep the annotation theory neutral and open for other users. But we think that it is legitimate to concentrate our limited resources on those features that are relevant for the project.

5 Frames and 'semantic' roles

We started our project with the basic assumption that ergative languages, by definition, could not have the syntactic categories subject and object. We understood that the concepts of 'agent' and 'patient' were introduced as their semantic substitutes, in the discussion of ergativity at least. Accordingly, we initially thought that Tibetan case markers and postpositions would refer to semantic roles. But, actually, how semantic are 'semantic roles'? An, in principle, unlimited number of truly semantic, verb specific roles was certainly not what we were looking for. We were thus attracted by the promising idea of a limited set of semantic role types, only to find this reduced to a merely binary set of "thematic proto-roles" (Dowty [5]), not so different from the notions of subject and object, not enough for ergative languages, where two different types of agent are encoded, and even less convincing in the view of yet another case marking for experiencer-subjects as in Ladakhi (Zeisler [18]).

We finally realised that (case, thematic, or semantic) roles are neither purely semantic nor purely syntactic categories. In order to cope with the interaction of these two layers, we hope to further develop the Indian concept of *kāraka*-relations as a syntactic-semantic interface (Cardona [3]). For the time being we conceive of our roles as 'macro-roles', possibly associated with properties such as [±animacy] or [±control]. They fill particular syntactic slots, but in doing so, the semantic component might be either bleached (eg when an instrument is treated as agent) or might be open for further specification or 'micro-roles': Verbs of the '(un)load' type may have either 'container' and 'content' argument in the two-place patient slot, but only the 'content' argument in the three-place patient slot. Our roles are thus necessarily differentiated with respect to valency and further specified according to their specific position in the frame, and their syntactic behaviour. Presently, we work with an inventory of 34 roles.

Semantic considerations lead to a terminological distinction of 'patient' (the second or third argument of an agentive verb) and 'undergoer' (the sole argument of an inagentive verb). The terminological identification of these two roles without consideration of valency, as in the most neutral designation "Arg_1" (eg Palmer et al. [12]) might be useful for languages like English with no [±control] distinction and plenty of ambitransitive verbs or for other types of valency raising, such as the 'experiencer derivation' in Ladakhi (Zeisler [18]). But Tibetan [–control] one-place verbs like *ḥbye*, *bye* 'open' or *ḥchag*, *chag* 'break' exclude the possibility of an intentional agent controlling the event (an external force argument might be added). Whereas [+control] two-place verbs like *ḥbyed*, *phyes*, *dbye*, *phyes* 'open' and *gcog*, *bcag*, *gcag*, *chog* 'break' inevitably presuppose the presence of an agent (or an agent-like force or agent-related instrument), even if this agent is not mentioned.

Our concept of 'frame' is based on a bottom-up description of how verb arguments are encoded in an utterance, rather than on intuitions about event structures or conceptualisations of events as eg in the FrameNet approach of Erk & Padó [6]. We do not have the empirical base for such top-down analysis: the dictionaries are often silent about the argument structure, and we will hardly find a native speaker of Classical Tibetan, not to mention of Old Tibetan. With respect to contemporary Ladakhi, we have sampled a relatively complete set of verbs for one dialect. In this database, which is going to be enlarged with data on other dialects, we have classified each case pattern (eg 03 Abs–~Loc; 06 Aes–Abs; 07 Erg–~Loc; 08 Erg–Abs). These schemes allow us to identify verbs of similar behaviour (including frame variation) and establish something like Levin's [11] verb classes. More basically, however, they help to reflect our guesses about Old and Classical Tibetan argument structure and may serve as a model for similar work on other Tibetan varieties.

References

[1] Andersen, Paul Kent (1987) Zero-anaphora and related phenomena in Classical Tibetan. *Studies in Language* 11, 279–312.

[2] Bielmeier, Roland (1985) *Das Märchen vom Prinzen Čobzaṅ. Eine tibetische Erzählung aus Baltistan. Text, Übersetzung, Grammatik und westtibetisch vergleichendes Glossar.* St. Augustin: VGH Wissenschaftsverlag.

[3] Cardona, George (1976) *Pāṇini. A survey of research.* The Hague, Paris: Mouton.

[4] Cooreman, A., Fox, B. and Givón, Talmy (1984) The discourse definition of ergativity. *Studies in Language* 8, 1–34.

[5] Dowty, David (1991) Thematic proto-roles and argument selection. *Language* 67.3, 547–629.

[6] Erk, Katrin and Sebastian Padó (2004) A powerful and versatile XML format for representing role-semantic annotation. In *Proceedings of LREC-2004*. Lisboa, Portugal.

[7] Francke, August Hermann. 1901. Sketch of Ladakhi grammar. *JASB* 70, 1-63. Reprint 1979 as *Ladakhi and Tibetan grammar.* Delhi: Seema Publications.

[8] Francke, August Hermann (1905-41) *Gšamyulna bšadpaḥi Kesargyi sgruŋs bžugs. A Lower Ladakhi version of the Kesar saga*. Calcutta: Royal Asiatic Society of Bengal.

[9] Hahn, Michael (1985) *Lehrbuch der klassischen tibetischen Schriftsprache.* Bonn: Indica et Tibetica.

[10] Koshal, Sanyukta. 1979. *Ladakhi Grammar*. Delhi, Varanasi, Patna.

[11] Levin, Beth (1993) *English verb classes and alternations: a preliminary investigation.* Chicago: Univ. of Chicago Press.

[12] Palmer, Martha, Mitch Markus, Scott Cotton, Kate Forbes, Daniel Gildea, Paul Kingsbury, Joseph Rosenzweig (2002) Proposition Bank: Automatic Content Extraction (ACE) at the University of Pennsylvania. http://www.cis.upenn.edu/~ace/

[13] Read, A.F.C. (1934) *Balti grammar.* London: The Royal Asiatic Society.

[14] Tournadre, Nicolas (1996) *L'ergativité en tibétain. Approche morphosyntaxique de la langue parlée.* Paris: Louvain.

[15] Wagner, Andreas und Bettina Zeisler (2004) A syntactically annotated corpus of Tibetan. In *Proceedings of LREC-2004*. Lisboa, Portugal.

[16] Zeisler, Bettina (1994) Ergativ, Passiv und Antipassiv. Entwicklungen im Tibetischen. *Zentralasiatische Studien* 24, 45-78.

[17] Zeisler, Bettina (2004) *Relative Tense and aspectual values in Tibetan languages. A comparative study.* Berlin, New York: Mouton de Gruyter.

[18] Zeisler, Bettina (to appear), Case patterns and pattern variation in Ladakhi: a field report. In: Bielmeier, Roland and Felix Haller (eds.) *Himalayan Linguistics and Beyond. Papers submitted to the 8th Himalayan Languages Symposium, held at the University of Berne, Switzerland, Sept. 19-22, 2002.*

[19] Zimmermann, Heinz (1979) *Wortart und Sprachstruktur im Tibetischen.* Wiesbaden: Harrassowitz.

Syntactic Interpretation of an Early New High German Corpus

Ulrike Demske, Nicola Frank, Stefanie Laufer, Hendrik Stiemer
Saarland University

1 Introduction

The last couple of years have seen a growing number of large natural language corpora leading to an increasing interest in empirical issues within the field of linguistics. Tools were developed to accommodate the need in detailed descriptions at word, phrase and sentence level in a variety of languages. So far, research focused on corpora of Present-Day languages with some notable exceptions regarding in particular the history of English (cf. the Penn-Helsinki-Corpus among others). This paper reports on a treebank project with the aim of building a syntactically annotated treebank for Early New High German (ENHG) encoding parts of speech as well as syntactic structure. The corpus currently consists of 7,500 sentences (130,500 words) making up the *Mercurius* (M), a weekly published newspaper in Hamburg in 1667 (cf. Gieseler/Kühnle-Xemaire [7] for further information). The syntactic annotation is based on the tool *Annotate* (cf. Brants [2], [3]; Brants et al. [4]) developed for the interactive annotation of a Present-Day German (PDG) corpus, i.e. the NEGRA and the TIGER corpus, respectively (Brants et al. [1]). Hence, a further aim of the project is to meet the particular challenges provided by applying this tool to a corpus of historical texts. The challenges to be met are in particular:
(i) To develop diagnostics for manually identifying sentence boundaries, since punctuation provides no means to automatically identify sentence boundaries in ENHG.
(ii) To modify the part-of-speech-tagging tool in such a way as to cope with sequences of two or three graphical words, with the linguistic context suggesting that the respective sequence corresponds to only one morphological word as exemplified by the pattern of so-called genitive compounds in ENHG.

The outline of our paper is as follows: In section 2, we introduce the ENHG facts with respect to sentence structure, suggesting to consider the position of the finite verb as indicative of the existence of independent sentences. Section 3 addresses the questions arising with part-of-speech-tagging of so-called genitive compounds, and section 4 gives an outlook.

2 Identifying sentence boundaries

The sentence is a standard textual unit in language processing applications. Before even starting to syntactically analyze, the predefinition of sentence boundaries within an ENHG newspaper corpus was a necessary task to be done in the pre-processing since the PDG use of punctuation marks as indicators of sentence boundaries has to be differentiated from the function they had in ENHG. In ENHG, the system of punctuation was rather a rhetorical one as opposed to the grammatical-syntactic function it has nowadays. In accordance to Stolt [12], ENHG punctuation marks serve to divide a sentence not into syntactic but into information units, i.e. instead of indicating the end of syntactic phrases they indicate the end of rhetorical parts of speech. Thus, in the predefinition of sentence boundaries, punctuation marks had to be largely disregarded and, instead, new criteria had to be defined which would be sufficient for the classification of whole sentences. Going on the assumption that the position of the finite verb has already been a valid criterion in ENHG as it is nowadays for the categorization of clause types, it seemed the obvious way to set a syntactic criterion for the determination of sentence boundaries. Since we claim that there are no independent VL-clauses in ENHG, our assumption is contradictory to Lötscher [9]. His argumentation is crucially based on the assumption that *d*-pronouns introducing VL-clauses are anaphoric whereas only the corresponding *w*-pronouns function as subordinators in ENHG. This classification of *d*-pronouns, however, holds for PDG but does not carry over to the pronominal system of ENHG with *d*-pronouns occurring in many linguistic contexts where only *w*-pronouns are licensed in PDG. *W*-pronouns, on the other hand, exhibit rather low frequencies suggesting that in ENHG a *d*-pronoun is not indicative of the independence of a clause but has to be analyzed as a relative pronoun introducing a relative clause in many instances (cf. Ebert [6], 167).

The position of the finite verb as a criterion for syntactic (in-) dependance of a clause is a fact that can easily and uncontroversially be observed, while the informational status of a clause in a larger context is subject to the respective interpretation (as Lötscher himself does admit). Therefore, the verb position turns out to be a clear-cut criterion, allowing an unambiguous handling of a large set of sentences. According to this criterion, we identified the following cases:

- Verb first (V1) verb second (V2) in independent or main clauses:

(1) a. und haben sie ihr Urtheil ehist zu erwarten.
and have they their verdict soon to expect

‘And they have to expect their verdict soon.’ (M. 88.18)[1]

b. und wir haben auch aus derselben Ursache solches
and we have too out the_same reason this

apprehendiren müssen.
learn must
‘And we have had to learn this, too, because of the same reason.’ (M. 90.1)

- Final position (VL) in subordinate clauses:

(2) Viel hoffen was gutes hiervon/[...]. **Welches sich bald äussern**
Many hoped sth. good out_of which itself soon utter

wird.
will.
‘Many hoped that it would result in something good [...], which will be proved soon.’ (M. 166.2)

Note that clauses with verb final position are treated like subordinate clauses even if preceded by a punctuation mark triggering a capital letter as in the ENHG example under (2), while initial and second position of the finite verb are taken as indicators of main clauses (cf. (1a) and (1b)). Lötscher in contrast analyzes examples as (2) as including two independent clauses with the latter instantiating the class of “relative-like” clauses. VL-clauses with the complementizer starting with a capital letter, however, are not restricted to “relative-like” contexts exhibiting loose informational relations between the clauses in question:

(3) Warschauer Briefe melden [...]/ **daß der Koenig sich wol auff**
Warsawian letters report that the king himself well

befinde. Daß aus der Vkraine ein Expresser
feels that from the Ukraine a courier

[...] gekommen Ø
arrived
‘Letters from Warsaw report that the king feels well [and] that a courier has arrived from the Ukraine’ (M. 48.24)

[1] The first number indicates the page(s) and the second number(s) indicate(s) the line(s) where the example starts.

In (3), both *daß*-clauses are complements of the predicate *melden* 'report' and must have the same syntactic relation to the matrix clause. We take examples such as (3) to provide further evidence for our analysis of (2) as one sentence containing an embedded relative clause.

- Even embedded clauses lacking any finite auxiliary occur frequently in ENHG (Schröder [10] among others), cf. (4a):

(4) a. Cap. Cornelius Schreck / **so unter den 5 Kriegs-**
Captain Cornelius Schreck / who under the 5 battle-

Schiffen gewesen Ø
ships been
'Captain Cornelius Schreck who has been among the five battle ships.' (M. 14.18)

b. der jenige Moßkowitische Gesandter/ **welcher am Käyserlichen**
that Moscowian minister who at imperial

Hofe gewesen ist
court been is
'that minister of Moscow who has been at the imperial court' (M. 814.10)

Clauses which obviously lack a finite auxiliary verb as in (4a) - cf. also the second *daß*-clause under (3) - are to be categorized as subordinate clauses due to their coexistence with frequent equivalent clause structures where the finite auxiliary verb is in final position (cf. (4b)).

3 Coping with genitive compounds

The word formation pattern of so-called genitive compounds is well known from PDG. The term 'genitive compound' refers to N + N compounds with the first constituent allegedly exhibiting genitive case as in (5a).

(5)	a. Kind-s-kopf	b. Liebe-s-brief	c. Freund-es-kreis
	child-SG.GEN_head	love-Ø_letter	friend-Ø_circle
	'big kid'	'love letter'	'circle of friends'

As a matter of fact, the supposed case marker -(*e*)*s* does not indicate genitive singular. At least in PDG, it is a mere linking element as can be shown by examples where the inflectional paradigm of the first constituent lacks any case marker indicating genitive case, cf. (5b)[2], or by compounds where the interpre-

[2] The genitive of *Liebe* 'love' is *Liebe* and not *Liebes*.

tation of the compound excludes an analysis of the first constituent in terms of a singular noun as exemplified by (5c).

So-called genitive compounds already occur in ENHG. Thereby, different spellings are to be found:

(6) a. der Reich-s-Tag
the empire-Ø_meeting
b. der Reich-s Tag
the empire-Ø meeting
'the meeting of the empire'

According to Demske [5], the word formation pattern of genitive compounds is the result of reanalysis, i.e. ENHG speakers analyze ambiguous patterns like (7) no longer as (7a) but as (7b) with the syntactic structure being reanalyzed as a morphological structure:

(7) dieser Stadt Graben

a. [[dies-er Stadt] Graben]
that-GEN city.GEN moat.NOM

b. [dies-er [Stadt Graben]]
that-NOM city moat.NOM

In (7a), *dieser Stadt Graben* is analyzed as a complex noun phrase introcued by a prenominal modifier with the determiner refering to the nominal modifier. In (7b), however, *Stadt Graben* is analyzed as a compound with the determiner *dieser* refering to the head noun, i.e. *Graben*. In this case, the compound has the same surface structure as the complex noun phrase because the determiner *dieser* can be analyzed as exhibiting either genitive or nominative case. In contrast to (7), no ambiguity arises in cases where the determiner exhibits different inflectional forms to indicate nominative or genitive case, respectively: There is no question that we deal with a complex noun phrase[3] in (8a), whereas example (6b), repeated here as (8b), instantiates a morphological structure:

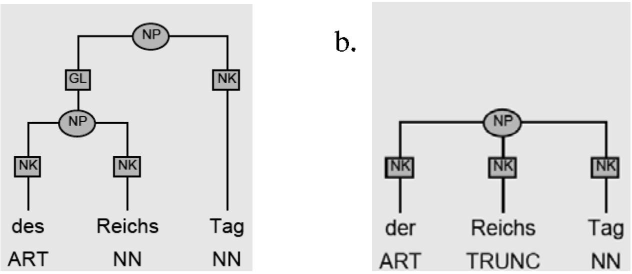
(8) a. [[d-es Reich-s] Tag]
the-GEN empire-GEN meeting
b. [d-er [Reich-s Tag]]
the-NOM empire-Ø meeting
'the meeting of the empire'

Hence, *-s* functions as a genitive case marker only in (8a), whereas in (8b) it has to be a mere linking element, cf. the PDG compounds given under (5).

[3] Though prenominal genitive phrases are quite common in ENHG, this particular phrase is not attested in the underlying corpus.

With respect to tagging, cases such as (8a) do not cause any problems, the tag NN (= normal noun) and the analysis as GL (= genitive left) is appropriate for *(des) Reichs* (cf. 10a). Ascribing the tag NN to *Reichs* in (8b), however, yields the wrong result: *Reichs* functions as the left constituent of a complex noun as suggested by the linguistic context. Providing on the other hand the two graphical words *Reichs Tag* with only one tag (NN) would mean to alter the handed down spelling, testifying a language change in progress throughout this period in the history of German. We therefore ascribe to *Reichs* in (8b) the tag TRUNC (= first constituent of a composition), a tag that is also used in PDG annotations for the first constituent in coordinated compounds such as *Pop- and Jazz-Fans* 'pop- and jazz-fans'. Thus, *Reichs* is to be bound as NK ('noun kernel') to the parent NP ('noun phrase') as shown in (10b). Cases such as (7), with the reference of the determiner being ambiguous, are treated like prenominal genitives, i.e. like (10a).

(10) a.



b.

4 Outlook

In our view, *Annotate* seems a suitable tool to build a treebank also for older stages of German: Accommodations regarding the segmentation of historical texts and the encoding of syntactic phenomena restricted to individual periods in the history of German are easily met. Furthermore, the particular design of *Annotate* allows to encode discontinuous constituents so numerous in older stages of German in a straightforward way. The high frequency of variation in spelling just requires a larger amount of data to be submitted to the training tool of *Annotate* in order to reach the same high probabilities in interactive part-of-speech-tagging in ENHG as in PDG.

Building the MERCURIUS Treebank, we provide the first syntactically annotated corpus for a historical period of German, i.e. ENHG. Though the treebank is rather restricted in size, the tools developed to meet the particular needs of a historical corpus can be applied to other historical texts. As for ENHG, a representative corpus of 1.500 texts has been compiled in the late seventies of the 20th century (cf. Hoffmann/Wetter [8], Solms/Wegera [11] for the corpus structure), such that the results of the MERCURIUS Treebank project can be used to build a comprehensive source for any research questions concerning the historical syntax of ENHG. Future work will even comprise the construction of treebanks for all periods of the history of German (http://www.deutschdiachrondigital.de/).

Source

[M] = Mercurius 1667. Nordischer Mercurius. Welcher kürtzlich vorstellet/ was in diesem 1667. Jahre an Novellen aus Europa einkommen ist. Hamburg 1667.

References

[1] Brants, Sabine/Dipper, Stefanie/Hansen, Silvia/Lezius, Wolfgang/Smith, George (2002): The TIGER Treebank. Proceedings of the Workshop on Treebanks and Linguistic Theories Sozopol/Bulgarien, 24-41.

[2] Brants, Thorsten (1999): Tagging and Parsing with Cascaded Markov Models – Automation of Corpus Annotation. Saarbrücken: DFKI, Saarbrücken Dissertations in Computational Linguistics and Language Technology Vol. 6.

[3] Brants, Thorsten (2000): TnT - A Statistical Part-of-Speech Tagger. Proceedings of the Sixth Conference on Applied Natural Language Processing ANLP-2000. Seattle/WA.

[4] Brants, Thorsten/Skut, Wojciech/Uszkoreit, Hans (1999): Syntactic Annotation of a German Newspaper Corpus. Proceedings of the ATALA Treebank Workshop. Paris, 69-76.

[5] Demske, Ulrike (1998): Case compounds in the History of German. Germanistische Linguistik 141-142, 149-176.

[6] Ebert, Robert Peter (1986): Historische Syntax des Deutschen II: 1300 – 1750. Frankfurt/M.: Peter Lang.

[7] Gieseler, Jens/Kühnle-Xemaire, Elke (1995): Der Nordische Mercurius - eine besondere Zeitung des 17. Jahrhunderts? Eine sprachwissenschaftliche Untersuchung der Hamburger Zeitung. Publizistik 40, 2.

[8] Hoffmann, Walter/Wetter, Friedrich (1987): Bibliographie frühneuhochdeutscher Quellen: ein kommentiertes Verzeichnis von Texten des 14. -17. Jahrhunderts. 2. überarbeitete Auflage. Frankfurt/M.: Peter Lang.

[9] Lötscher, Andreas (2000): Verbendstellung im Hauptsatz in der deutschen Prosa des 15. und 16. Jahrhunderts. Sprachwissenschaft 25, 153-191.

[10] Schröder, Werner (1985): Auxiliar-Ellipsen bei Geiler von Kayserberg und bei Luther. Wiesbaden/Stuttgart: Steiner.

[11] Solms, Hans-Joachim/Wegera, Klaus-Peter (1998): Das Bonner Frühneuhochdeutschkorpus. Rückblick und Perspektiven. In: Rolf Bergmann (Hg.): Probleme der Textauswahl für einen elektronischen Thesaurus. Beiträge zum ersten Göttinger Arbeitsgespräch zur historischen deutschen Wortforschung am 1. und 2. November 1996. Stuttgart: Hirzel, 22-39.

[12] Stolt, Birgit (1990): Redeglieder, Informationseinheiten: Cola und Commata in Luthers Syntax. In: Anne Betten (Hg.): Neuere Forschungen zur historischen Syntax des Deutschen. Tübingen: Niemeyer, 379-392.

SearchTree - A user-friendly treebank search interface

Lars Nygaard and Janne Bondi Johannessen

University of Oslo
The Text Laboratory
`http://www.hf.uio.no/tekstlab/`
`{larsnyg, jannebj}@ilf.uio.no`

1 Introduction

Treebanks constitute a valuable resource for linguists, but their usefulness is often reduced by hard-to-use search interfaces, often requiring the user to learn the detailed knowledge of query languages or regular expressions as well as of tag sets, often with non-intuitive tag names and abbreviations. Writing complex queries becomes a slow and error-prone process. In addition, the user will often have to learn several query languages, with smaller and larger differences, adding to the confusion.

We think that user-friendliness is as important for treebank use as it is for the use of text corpora generally. In this paper we describe SearchTree, a web-based interface for queries in treebanks. SearchTree is not tied to any particular treebank, although its main motivation comes from the need for a proper search interface for the Sofie Treebank – a parallel treebank of mainly North European languages (Danish, Dutch, English, Estonian, Faroese, Finnish, German, Icelandic, Norwegian, Swedish).[1]

In the following, we will provide a description of SearchTree, and exemplify with monolingual searches in the Penn Treebank, and with parallel searches in the Sofie Treebank. We will then briefly perform a a comparison with other treebank search interfaces.

2 The advantages of SearchTree

SearchTree is implemented in HTML, JavaScript and Perl. As a search engine it uses TGrep2 (Rohde 2004), a query engine for linguistically annotated trees. TGrep2 has good functionality for many kinds of tree search, but cannot deal

[1] The Sofie Treebank, which is still under development, is the result of joint work of the members of the Nordic Treebank Network, http://w3.msi.vxu.se/~nivre/research/nt.html.

with crossing branches or secondary edges. An additional underlying query system will be provided in the near future to handle these.

There are several advantages in the SearchTree system.

- SearchTree is publically available; it is open source (see the reference list).
- It is accessible to the user via a web browser; no installation is necessary.
- SearchTree provides all tags and categories that are used in the searchable treebank(s); the user need not learn the tagsets before starting the queries.
- SearchTree is completely graphical in an intuitive interface.
- The results are presented in a user-friendly way, showing the query subtree, plus the whole sentence as text, and with the option of seeing the whole tree.
- SearchTree is not tied to any specific corpus or formalism.

Most users will be linguists with little wish to learn the syntax and terminology of a complex search language like TGrep2. We think that the user is better served by clicking in boxes than having to formulate complex queries in a complex query language. But other users will also be pleased not having to face the danger of putting a parenthesis in the wrong place etc. Furthermore, users should not need to know by heart all the names of parts of speech and categories used. Such expressions should be given as lists.

3 The SearchTree query interface

In this section, we will present the SearchTree query interface. The basic features of the web interface for monolingual searches are illustrated below:

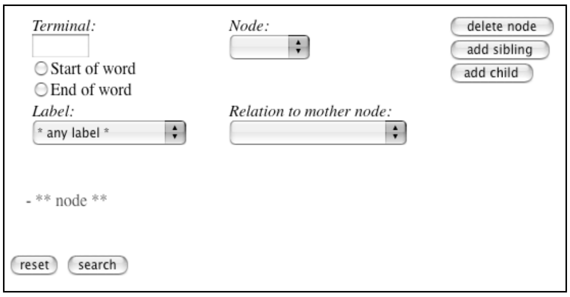


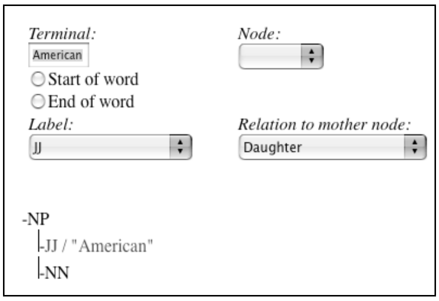
Figure 1

The user starts by clicking on the red, highlighted string node, activating this as the first node. The next step is to choose a label for this node from the pull-down menu above it. A new node can now be added by clicking on one of the boxes on the right-hand side, which will insert a new node in the required place (i.e., as sibling or daughter). Again a label for this node must be chosen. Further specifications can be picked from the pull-down menu for relations between nodes, or for writing a terminal in the box on the upper left, or for specifying the ”modality” of the node; optional, negated or obligatory. The user can only activate one node at the time. Any option that is being chosen will apply to that particular activated node. We will illustrate this.

Let us say that we are interested in NPs that contain at least the adjective *American* and an optional, common noun. This would give the following TGrep2 query:

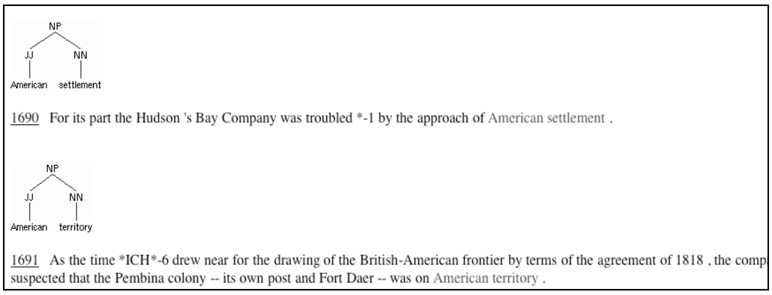
(1) (NP < (JJ < /^American$/)?< (NN))

In order to be able to write this query, the user would ordinarily have to be able to know the syntactic tags, and the syntax and inventory of the TGrep2 search language. In SearchTree, the query is formulated instead as in figure 2, with results given in figure 3:


Terminal: American
○ Start of word
○ End of word
Node:
Label: JJ
Relation to mother node: Daughter

-NP
 └-JJ / "American"
 └-NN

Figure 2: Searching for a subtree in a monolingual treebank



1690 For its part the Hudson 's Bay Company was troubled *-1 by the approach of American settlement .

1691 As the time *ICH*-6 drew near for the drawing of the British-American frontier by terms of the agreement of 1818 , the comp
suspected that the Pembina colony -- its own post and Fort Daer -- was on American territory .

Figure 3: Resulting hits in the Penn Treebank.

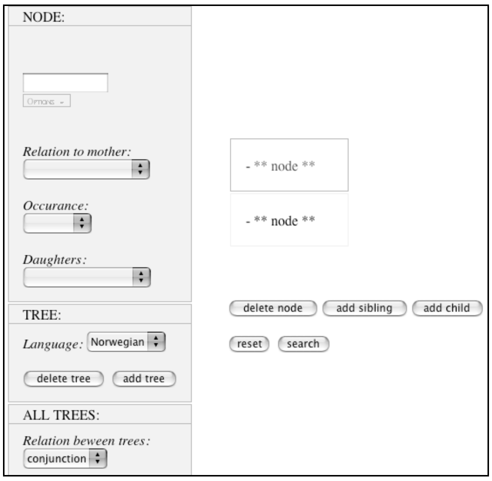
The hits are shown in a way that emphasises user-friendliness:

• For each hit, the subtree that matches the searchtree is illustrated with labels and terminals.

• The full sentence is shown below it as a text string, and with the relevant search phrase highlighted.

- The tree for the whole sentence can be viewed by clicking on its left.
- The search expression in TGrep2 query syntax is shown at the results page, providing the user with a way of checking the query, but also to be used as a starting point for a more complex query than the graphical interface itself allows, or even for learning the syntax of TGrep2.

A parallel treebank faces more challenges. First, the same ones apply here as with a monolingual treebank. Second, the treebanks that constitute the parallel treebank may have different tagsets. Third, there may be a series of languages that should be searchable as “source” and “target” corpora at the same time. Fourth, if disjunction is a possible query option, the various combinations multiply and make a user-friendly interface quite hard to maintain. SearchTree has tried to cater for many of the problems. Below is an illustration of the search interface.



Language: Norwegian

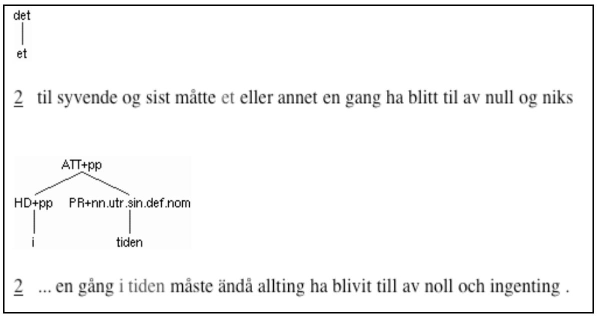
Relation beween trees: conjunction

Figure 4: A query interface for parallel treebanks.

As before, the active node is highlighted. In the window above, there are two nodes that can be activated; any search must satisfy both criteria in the languages that are specified for them. For each node, a new node can be added or deleted. The upper left box is for writing the string of a terminal node, and the pop-up menu underneath gives the possibility of specifying parts of the search string, and label. The menus underneath specify the relation the active node has to its mother; optional, obligatory or forbidden; and the relationship it has to its daughters (immediate dominance or other).

The parallel search interface options are seen in the TREE and ALL TREES parts of the window. For each node, a language has to be chosen. The relationship between the trees has to be specified; conjunctive or disjunctive. When a language is chosen, the label menu will reflect its tagset.

Below is part of a results page. The query specified any Norwegian sentence containing a node with the label *det* (determiner) and any sentence-aligned Swedish sentence containing the node PP.



2 til syvende og sist måtte et eller annet en gang ha blitt til av null og niks

2 ... en gång i tiden måste ändå allting ha blivit till av noll och ingenting .

Figure 5: Search results in parallel treebank.

4 Other treebank interfaces

The TIGERin search interface (Voormann and Lezius 2002), based on Tiger Search, is impressive in its expressive power and in its graphics. However, we think that the TIGERin also has some drawbacks that we have solved:

• TIGERin has to be downloaded and installed locally by every user. This can and often does lead to unforeseen, but trivial problems.

• The user must have a local copy of the full treebank.

• TIGERin is highly graphical. However, we think that it is not very intuitive, although this can be a matter of taste. The user is required to click on invisible objects.

• TIGERin presents all tags and categories in ready menus, which is good. However, not all tag names are equally transparent.

• The results from a search with TIGERin are presented as trees of the full sentences in which the search-tree occurs; and with the sentence presented as terminals as a straight line at the bottom of the tree. While this representation has some advantages, such as presenting a full overview of each sentence, the results may be overwhelming. A lot of scrolling is often necessary.

SearchTree avoids some of the problems above for the following reasons: it is web-based; it is totally based on clicking on options that are all visible; the tag names are presented as full names, not abbreviations; the query results are shown in a two step way. First: each hit with the subtree that matches the query, and with the full sentence as a text string with a highlighted search phrase. Second (after optional clicking on the left-hand side): the full sentence tree.

TIGERin makes available the search options of Tiger Search, e.g. for crossing edges, and disjunctive search. SearchTree at the moment is built on top of TGrep2, which makes crossing and secondary edges unavailable for search. This is on the list of future work.

The VIQTORYA query tool (Steiner and Kallmeyer 2002) was developed for the Tübingen German Treebank, and has in common with the current version of SearchTree that it does not cater for crossing and secondary branches. Unlike SearchTree it is based on a tailor-made (first-order logic) query system. VIQTORYA is an abbreviation for “a visual query tool for syntactically annotated corpora”, but relies on information that is external to the interface: The annotation scheme must be looked up in specific stylebooks and guidelines that are found elsewhere.

NetGraph (Mirovsky et al, manuscript) and Oraculum (Ljubopytnov et al. 2002) are two systems for searching through the Prague Dependency Treebank. NetGraph, like SearchTree, functions in an Internet environment, and both aim at having graphical interfaces. Oraculum is claimed to be more advanced than NetGraph, but we have not been able to confirm this. Icecup is the search interface for the ICE corpora. It is very advanced w.r.t. queries on trees, but its user interface, although graphical, is not very user-friendly, with a wealth of unintuitive symbols. Unlike SearchTree it must be downloaded on a local computer with specific technical requirements.

5 Conclusion and future work

We have described SearchTree, a user-friendly interface for monolingual and parallel treebank queries. We will continue to increase the flexibility of the interface; increase support for other search engines (at the moment it system works for Microsoft Explorer and Opera); expand the system to support other tree drawing methods; make the system more agnostic to the linguistic approaches to treebank annotation. Especially, we want to support dependency and complex nodes, crossing and secondary branches. In its current form, SearchTree cannot express the full flexibility of the TGrep2 query language. Particularly, it is a hard problem to allow a graphical user interface to express complex bracketing of nodes with disjunction and conjunction conditions in a simple and intuitive way.

References

[1] Icecup: http://www.ucl.ac.uk/english-usage/ice-gb/icecup.htm

[2] SearchTree: http://logos.uio.no/SearchTree

[3]The Sofie Treebank - A Parallel Treebank of North European languages: http://omilia.uio.no/sofie/

[4] The Tiger treebank, search: http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/

[5] TGrep2: http://tedlab.mit.edu/~dr/TGrep2/

[6] Brants, S., S. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. The Tiger treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories, September 20-21*, Sozopol, Bulgaria.

[7] Ljubopytnov, V., P. Nemec, M. Pilatov, J. Reschke, and J. Stuchl. 2002. Oraculum, a System for Complex Linguistic Queries. In M. Bjelikov (ed.); *SOFSEM 2002 Student Research Forum*, pp. 27-34.

[8] Mirovsky, J., R. Ondruska, and D. Prusa. Searching through the Prague Dependency Treebank – Conception and Architecture. Manuscript. Faculty of Mathematics and Physics. Charles University, Prague.

[9] Rohde, D.L.T. 2004. TGreo2 User Manual version 1.12. http://tedlab.mit.edu/~dr/TGrep2/

[10] Steiner, I., and L. Kallmeyer.2002. VIQTORYA – A Visual Query Tool for Syntactically Annotated Corpora. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, Las Palmas, Gran Canaria, pp. 1704-1711.

[11] Voormann, H., and W. Lezius. 2002. TIGERin – Grafische Eingabe von Benutzeranfragen für ein Baumbank-Anfragewerkzeug. In S. Busemann (ed.); *Proceedings der 6. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2002)*. Saarbrücken.

A Treebank-Driven Approach to Semantic Lexicons Creation

Kiril Simov, Petya Osenova
BulTreeBank Project
http://www.BulTreeBank.org
Linguistic Modelling Laboratory, Bulgarian Academy of Sciences
Acad. G. Bonchev St. 25A, 1113 Sofia, Bulgaria
kivs@bultreebank.org, petya@bultreebank.org

1 Introduction

In this paper we aim at showing how the information from a richly annotated treebank can be used for facilitating the construction of a semantic lexicon when such a lexicon lacks for a certain language. We demonstrate this idea with the Bulgarian treebank (BulTreeBank).

The structure of the paper is as follows: the next section briefly describes the levels of linguistic interpretation in the treebank. In section 3 we present the model of a semantic lexicon which we are using. Section 4 outlines the algorithm for the extraction of the semantic information from the treebank. The last section concludes the paper.

2 The Levels of Linguistic Interpretation in BulTreeBank

Our treebank (200 000 words) is a part of a morphosyntactic corpus (1 000 000 words). It is manually processed and consists of the following analytical levels:

1. Token level - the tokens are divided into common words, names, abbreviations, numericals, symbols, punctuation.
2. Morphosyntactic level - the correct POS tag with the appropriate characteristics is selected among alternatives (if any). At this level we designate different semantic types of adverbials: time, place, manner, quantity, modal and named entities: person, organization, local, other. This classification

helps us to form the verb frames at later stages. In contrast to [1] we do not exclude them from the 'inner participants' list.

3. HPSG-oriented syntactic level - it combines the constituent representation, grammatical roles assignment and head-dependent distinction. For each phrasal domain we annotated the role of the dependent element(s): complement, subject, adjuncts.

4. Intrasentential co-reference relations level - here we rely on the structure-sharing mechanisms in HPSG and we assume different relations between nominals or nominalized elements that reflect the phenomena binding, pro-drop, control etc.

In NLP literature there are a number of schemes for annotating more complex co-reference relations in treebanks, see [5], and [4] among others. For the moment we have annotated the following referential relations: equality, subset-of and member-of (we have not annotated relations like part-of). We capture all main co-references of the following syntactic representations: subject and object relations, reflexivity, possession, clitic-doubled structures, secondary predicated adjectives with the subject or the object. Also we represent co-reference between synonymic expressions, changed referring expressions in direct-indirect speech, nominalizations. Part of the co-reference relations within a sentence are not explicated because they can be easily inferred from the syntactic structure like co-reference between the relative pronoun and the head noun when a relative clause modifies a noun phrase.

HPSG theory implies a lexicon, which in a general way reflects the idea of the 'frame-semantic approach' as stated in [7] and [3]. For instance, the semantics of the verb *give* will include a representation of the relation 'give' with corresponding arguments:

$$\begin{bmatrix} \text{CONTENT} & \left\{ \begin{bmatrix} \text{REL} & \text{give} \\ \text{ARG1} & \text{giver} \\ \text{ARG2} & \text{given} \\ \text{ARG3} & \text{givee} \end{bmatrix} \right\} \end{bmatrix}$$

In order to supply this type of information and/or to make it more concrete, we use two dictionaries in the semantic annotation of the treebank: the machine-readable Valency lexicon and the Seed Semantic lexicon[1]. On the one hand, these

[1] We call this lexicon a Seed Semantic lexicon because it contains only about 3000 nouns and does not contain other parts of speech. But otherwise it follows the chosen model for the sematic dictionary we want to construct.

lexicons represent the model of argument-predicate and semantic representation. But, on the other hand, they are far from covering all the treebank data.

In the following section we describe the model of the semantic dictionary which we follow in our work.

3 The Semantic Lexicon

Our aim is to show how the construction of a semantic lexicon can be facilitated by using the annotated linguistic relations in the treebank as supplementary to the available, but incomplete Valency and Seed Semantic lexicons.

In our view, an elaborate semantic lexicon has to contain both pieces of information: subcategorization and semantics. Additionally, the argument positions in the subcategorization need to be syntactically and semantically constrained.

As it was mentioned above, semantic information plays a crucial role in the process of parse discrimination on which the construction of our treebank depends. Thus, in order to support the selectional restrictions imposed by the valency dictionary and to facilitate its usage, we decided to compile a semantic lexicon along the guidelines of SIMPLE project — [6]. Generally, the structure of the lexical items follow the structure of predetermined templates which contain several fields and relations between them. For consistency each template is connected to a concept in the SIMPLE core ontology. It is worth mentioning that we follow an extended variant of the core ontology, namely - with taking into account Pustejovsky's qualia. Also the SIMPLE model of semantic lexicon includes representation of the valency of the words together with constraints over the arguments. Another important advantage of the SIMPLE model is that it is compatible with WordNet model of a semantic lexicon. Thus it is a good model for the creation of a semantic lexicon for HPSG. Our goal is to create such a dictionary for Bulgarian with wide coverage.

In our work we used two lexicons which were at our disposal before the experiment with the extraction of additional information from the treebank:

The Valency Dictionary consists of 1000 most frequent verbs and their valency frames. Each verb has a gloss and one or more frames. Each frame defines the number and the kind of the arguments imposing morphosyntactic and semantic restrictions over them. The original semantic restrictions over the arguments are extracted and matched against the SIMPLE core ontology. The frames of the most frequent verbs are compared to the corpus data (the morphologically annotated corpus) and repaired if necessary (new frames are added, some of the existing frames are deleted or fine-grained). We envisage to enlarge the coverage of this dictionary with the help of some derivational means, such as the verb prefixes.

The second lexicon contains 3000 of the most frequent nouns. They are classified with respect to the ontological hierarchy without specifying the synonymic relations between them. Also, the named entities and the adjectives have been classified with respect to the same ontology. We call this dictionary Seed Semantic lexicon.

In order to extend both lexicons we use the information encoded in the treebank. First, we annotated all the words in the treebank with the information available in the lexicons. Then we used the syntactic and co-referential information encoded within the treebank in order to disambiguate the annotated words. Afterwards, we collected the new information and inspected it manually.

4 The Algorithm

In order to extend the coverage of the semantic information, we decided to rely on the following corpus-based 'scratch' method along with the classification of the words against the SIMPLE ontology:

1. Verb annotation.

   Each verb in a sentence of the treebank is annotated with the frame descriptions from the Valency dictionary (if there is a lexical entry for the verb). Each of the arguments in a frame of the verb is connected to some of the verb dependents in the syntactic annotation. This is possible for the subject, the direct object and the indirect object. Note that sometimes there is a mapping from an indirect object in the Valency dictionary to an adjunct role in the annotation of the treebank.

2. Noun annotation.

   Each noun in the treebank is annotated with all the semantic classes in the semantic lexicon (if there is a lexical entry). On the one hand, this information is important for the verbs to select the appropriate arguments. On the other hand, it helps to classify named entities with better accuracy.

3. Disambiguation.

   This step is based on the idea of lexical chains: a set of coherently interrelated words in the text as presented in [2]. The connection between the words is defined on the basis of lexical relations like synonymy, hyperonymy, meronymy etc, which are classified as extra-strong, strong, medium, etc. The words in a lexical chain are connected with relations that represent different degrees of ontological similarity. We focused on extra-strong (i.e. literal repetition) and some of the strong relations, namely - the first type: when there is a synset (a set of synonyms) common to two different words, such as human and person, and the third type, namely when there is some kind of link between a synset associated with each word if one word is a compound word or a phrase that includes the other, such as *school* and *private school*. The second type (when there is a horizontal link between synsets associated with two different words, such as pre- cursor and successor) as well as medium strong relations are not considered, because apart from the upper part, the rest of the hierarchy is rather flat and therefore - unreliable. In the treebank we define lexical chains on the basis of co-referential relations and apply the idea of ontological similarity between the co-referent elements.

For each verb annotated with more than one frame we check whether some of the arguments in some of the frames disagree with the morphological and/or semantic information of the head noun of the corresponding element in the syntactic structure. If such a disagreement exists we delete the frame from the annotation of the verb.

For each noun annotated with more than one semantic class we check two things: (1) whether some of the semantic classes disagree with the selectional restrictions of some frame of the verb in the sentence (if the noun is a head noun mapped to some of the arguments in the frame). In this case we remove the class from the annotation of the noun; (2) we are using the coreferential relations with nouns or pronouns[2] to rule out more semantic classes.

These disambiguation rules are applied only when there are sure indicators for them, otherwise we leave the ambiguity in the annotation unresolved.

4. Classification.

   We classify the nouns in the text in equivalent classes on the basis of their participation in a coreferential relation or their headedness towards the same argument for different occurrences of the same verb. If there is an ambiguity, several equivalent classes are constructed.

5. Manual validation.

   An expert manually checks over the equivalent classes and creates appropriate lexical entries. For example, in the phrase ‘to write an application’, ‘application’ is added to the semantic class of the word ‘letter’.

[2] We assume that two semantic classes agree with each other if they are the same or one is a superclass of the other.

5 Conclusion

Thus, a semantic lexicon can be built in a bootstrapping manner. It is unordered (i.e. most of the hypernymic, synonymic, meronymic relations are hidden), but lexically rich. Later, gradually, the lexical relations will be added to this lexicon.

Note that the Treebank contains implicitly other predicate-argument patterns, which are extracted and processed as well. Here we have in mind not only all the cases of type verb-dependent, but also some fixed phrases: idioms, parenthetical expressions, verbs of saying which uniquely determine the semantic classes of their syntactic context (dependent elements or heads).

References

[1] Jan Hajič, Jarmila Panevová, Zdeňka Urešová, Alevtina Bémová, Veronika Kolářová and Petr Pajas. 2003. *PDT-VALLEX: Creating a Large-coverage Valency Lexicon for Treebank Annotation.* In the Proceedings of The Second Workshop on Treebanks and Linguistic Theories. Sweden, pp. 57–68.

[2] Graeme Hirst and David St-Onge. *Lexical chains as representations of context for the detection and correction of malapropisms.* In: Christiane Fellbaum (editor), WordNet: An electronic lexical database, Cambridge, MA: The MIT Press, 1998.

[3] Paul Kingsbury and Martha Palmer. 2003. *PropBank: the Next Level of TreeBank*. In the Proceedings of The Second Workshop on Treebanks and Linguistic Theories. Sweden.

[4] Lucie Kučová and Eva Hajičová*Coreferential relations in the Prague Dependency Treebank.* In the Proceedings from FDSL5, Leipzig, 26–28 November 2003.

[5] Kerstin Kunz and Silvia Hansen-Schirra. 2003. *Coreference Annotation of the TIGER Treebank.* In the Proceedings of The Second Workshop on Treebanks and Linguistic Theories. Sweden, pp. 221–224.

[6] Alessandro Lenci et. al. 2000. *SIMPLE Work Package 2 — Linguistic Specifications, Deliverable D2.1,* ILC-CNR, Pisa.

[7] John B. Lowe, Coilin F. Baker, Charles J. Fillmore. 1997. *A Frame-Semantic Approach to Semantic Annotation*. In Proceedings of the SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?. USA.

Interrelating Treebanks with Language-Specific Descriptions of Information Structure

Daniel Storbeck Sanghee Kwon Felix Sasaki
Andreas Witt

Bielefeld University
Department of Linguistics and Literary Studies

1 Subject of the paper

The motivation for this article is to describe a methodology for interrelating and analyzing language and theory-specific corpus data from various languages. As an example phenomeon we use information structure (IS, see [3]) in treebanks from three languages: Spanish, Korean and Japanese. Korean and Japanese are typologically close, while both are typologically different from Spanish. Therefore, the problem of annotating IS is that there are diverging language-specific formal linguistic means for the realization of IS-functions (like "topicalization / contrast") on various levels like prosody, morphology and word-order. Hence, it is necessary to describe the relations between language-specific formal means and functional views on IS, and how to operationalize these relations for corpus analysis.

2 The methodology

There are projects which deal with a corpus-based approach to IS, e.g. [2]. Nevertheless, these projects mainly deal with european languages, and — different to our approach — they don't rely on a dedicated methodology how to interrelate various levels of IS-function and formal means. Our methodology (see fig. 1) relies on (A) annotated corpora, which make use of a specific annotation format, and (B) a "conceptual level", which serves as a formal description of IS and which encompasses a set of language-specific or language-general "conceptual models".

(A) Usually, a single document grammar is used for the creation and validation of a corpus. This imposes restrictions on the possibilities of structuring the annotations and on their extensibility. [6] outlines a solution to this problem that

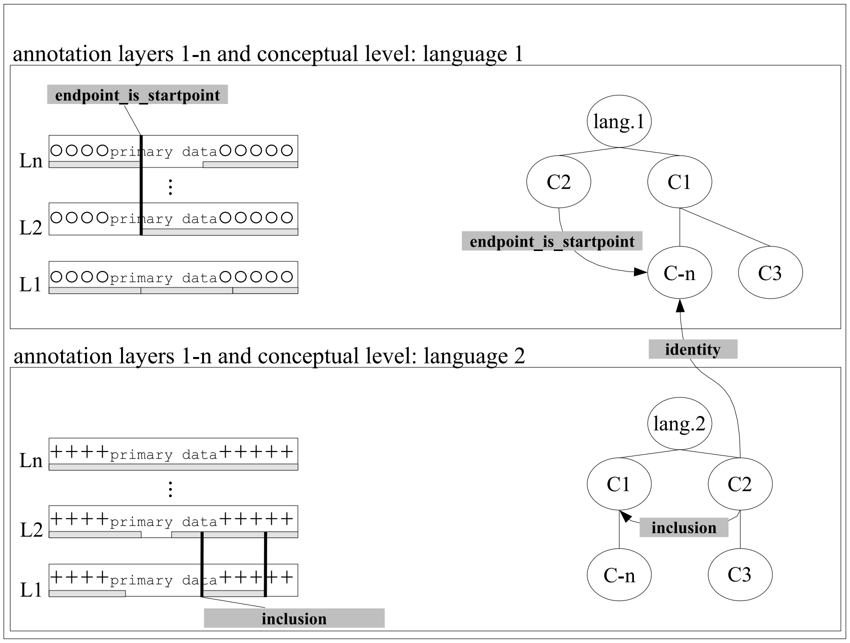


Figure 1: Overview of the methodology

allows for "multiple annotation hierarchies". Key features of this annotation format, which is visualized in the left part of fig. 1, are that it is XML-based, that modelling of alternative annotations based on different theoretical assumptions is possible, that each annotation layer can be created separately, and that new layers can be added at any time. Each annotation layer is represented in a separate XML-file and has its own document grammar. For each language, the primary data serve as an implicit link between multiple, separate annotations. Interrelations between the annotations are declared separately, i.e. on the conceptual level (see below and the right part of fig. 1), in order not to affect the annotation.

The annotations are transformed into Prolog facts. The Prolog fact base can be queried with a dedicated query language. The query language makes use of a closed set of predicates which define "positional relations" between annotations on separate layers. Such relations like "endpoint_is_startpoint"[1], "endpoint identity" or "inclusion"[2], can be derived from [1]. They are applied to textual data in the corpus by referring to character positions in the string, e.g. two annotations have the relation "identity" if they span the same range of characters.

[1] See for example the annotations on layer n and layer 2 of language 1 in fig. 1.

[2] See for example the annotations on layer 1 and layer 2 of language 2 in fig. 1.

(B) A key characteristic of the methodology is that these positional relations can be hypothetically declared at the conceptual level and / or heuristically derived from the corpus data, cf. [4]. The relations can be applied in three ways. First, as described above, they allow for the interrelation of annotations on different layers. Second, as can be seen in the right part of fig. 1, the relations can be used to describe "interconceptual relations" in a conceptual model for a language. For example the relation "endpoint_is_startpoint" between some annotations for language 1 can be interpreted as a relation between the concepts C2 and C-n, which are part of a conceptual model for that language. And third, the relations serve as interconceptual relations between different conceptual models, i.e. for different languages. For example in fig. 1, the concept C2 from the model for language 2 and the concept C-n from the model for language 1 are related via the relation "identity".

[5] illustrate the application of this methodology to japanese data, which are annotated according to several heterogeneous, theory-specific models of linguistic phenomena. The description of relations between such heterogene annotation units is done within the conceptual level. In the remainder of this paper, after describing the properties of IS in the three languages in question (section 3), several use cases for the methodology in the area of IS will be exemplified:

1. to describe relations between linguistic forms and IS-functions within one language, making use of annotations on several layers (see example 1 in section 4.1);

2. to interrelate language-specific, corpus-based descriptions of IS-functions on the conceptual level (see example 2 in section 4.2); and

3. to explicate typological differences of two given languages on the conceptual level (see example 3 in section 4.2).

3 Language-specific characteristics of IS

In terms of morphosyntax, it is usually said that Korean and Japanese are typologically closely related. Grammatical functions are normally morphologically marked by particles or verbal suffixes which contain their own meaning features. Referring to morphological elements such as the topic particles neun/eun (kor.) and wa (jap.) and case particles i/ga/(l)eul (kor.) and ga/o (jap.), categories like definiteness, genericity, topic/focus and contrast in a sentence can be described. The basic korean and japanese word oder is SOV but it is relatively flexible, i.e. scrambling is permitted. Spanish has a mixed morphology using analytic as well as synthetic formation principles. This is especially true of the verbal system with person, number and tense inflection but Spanish also makes extensive use of analytic tense and periphrastic aspect formation. The basic constituent order is considered SVO. While Japanese and Korean have postpositions and are of the “complement–verb” and “modifier–modified” order types, Spanish has the reverse order types and prepositions. In the following section we focus on the means each language uses to realize the topic-focus-articulation and the related phenomenon of contrast. Spanish relies primarily on constituent order and intonation to realize these phenomena, while Japanese and Korean additionally have explicit morphological means for this purpose.

4 Application of the methodology to IS

In order to illustrate what can be done with our methodology, we will present possible cases for intra- and interlingual comparisons of the specific and general sets of IS categories.

4.1 Description of intralingual relations

Case 1: In the first case we propose to interrelate annotations of language-general, IS-functional categories with annotations of language-specific, formal categories. This can be done describing the positional relations between annotations with respect to the primary textual data.

```
   Example 1: A korean sentence with possible annotation layers.
   Transl.: ``The child drinks _tea_.'' (And not someting else.)
   Morph. : child-nom  tee-topic  drink-present-deklarative

1:Primary data: ai------ga cha-neun masi--n----da
2:Sentence    : ---------------S-----------------
3:Word        : ----W----- ----W--- ------W------
4:Particles   :        nom     top-
5:IS-function :            topical./contrast
```

In this example there are the primary textual data (visualized as layer 1) and four annotation layers. Layers 2 and 3 express sentence and word segmentation. Layer 4 contains the annotations of language-specific morphemes while layer 5 contains annotations of IS-functions. Bearing in mind that there are not enough data available to derive a stable language-specific definition of IS-functions, e.g. “topicalization / contrast”, this annotation permits a hypothetical and partial definition, though, as follows: Topicalization / contrast is positionally included in a stretch of text representing a sentence and it has a positional identity relationship with a word unit which has a positional inclusion and endpoint identity relationship with a topic marker. Furthermore the word unit has neither starting point nor endpoint identity with the including sentence unit. Since IS-functions can be realized via several combinations of language-specific formal means, their resulting definitions as derived from the annotation are actually sets of definitions, each one of them describing a pattern of a characteristic configuration of language-specific means.

The advantage of the approach of multiple annotation is that such definitions can be established and refined during every phase of elaboration of the corpus. More layers can be added when more annotations are available or when diverse, new theoretical viewpoints on the same primary data are adopted. The annotations can be flexibly extended because they are not tied to a single, immutable document grammar.

4.2 Description of interlingual relations

The following two cases illustrate how an interrelation of IS across languages is possible. Different to the example in the last section, these cases mainly rely on the conceptual level.

Case 2: For structurally similar languages like Japanese and Korean it can be useful to query treebanks of the two languages using only one of the specific sets of categories. In this case a treebank of Japanese could be transparently queried with categories of a specific model for Korean. A common problem in defining correspondences between the two sets of categories is that one of them has more finegrained definitions of some categories than the other. Suppose, a given model for Korean annotates two types of topicalization constructions: 1. topicalization with object fronting (T1) and 2. topicalization including a topic morpheme (T2), see example 2. In a given model for Japanese only a general category “topicalization” (T) might be annotated. A plain identification of the topicalization types in the following example leads to a loss of information.

```
  Example 2:

*     Korean
1:Prim. dat.: cha---reul ai----ga masi--n----da
2:Sentence  : ---------------S-----------------
3:Word      : ----W----- ----W--- ------W------
4:IS-catego.: ----T1----                       (object fronting)
5:Particles :       akk-      nom    pres.decl.
```

```
*   Japanese
1:Prim. dat.: cha-----wa kodomo--ga nomu
2:Sentence  : ------------S-----------
3:Word      : -----W---- ----W----- -W--
4:IS-catego.: -----T----
                                       *Added layer:
5:Particles :        top        nom
                                       *Reconstructed categories:
6:IS-catego.: ----T2----
```

To solve this problem, a distinction of the two types of topic like in the korean annotation is created in the japanese annotation. This is done creating an additional annotation layer describing particles. Then, using this morphological information, the definition of topicalization in the japanese model can be refined to distinguish the types T1 and T2. In the conceptual level, these definitions are mapped on corresponding general concepts. These concepts allow for the interrelation of language-specific categories across languages.

Case 3: The last example focusses on the conceptual level and illustrates how the methodology can be used to explicate typological differences between two languages. Suppose, Spanish and Korean are to be compared and for both languages the same language-general concepts have been defined. Then, these concepts can be redefined into several subordinate concepts, in order to make the typological differences explicit.

```
  Example 3:

*     Korean
1:Prim. dat.: cha---reul ai----ga masi--n----da
2:Particles :       akk-      nom
3:Stress    : ---

1:Prim. dat.: cha---neun ai----ga masi--n----da
2:Particles :       top-      nom
3:Stress    :

*    Spanish                    (_tea_ drinks the child)
1:Prim. dat.: té bebe el niño
2:Stress    : --
3:Gram. role: O- -V-- ---S---
```

Continuing with an example of the IS-function concept “topicalization”, in Korean the subconcepts “topicalization with fronted and stressed object” as well as "topicalization with object fronting plus topic marker" can be created. As for Spanish, only the first concept can be applied. In this way, general superordinate concepts are linked to the corresponding language-specific subordinate concepts, i.e. definitions for each language. Thus, this simple conceptual level of topicalization types expresses a typological difference between Korean and Spanish, i.e. that the korean concept "topicalization with object fronting plus topic marker" has no corresponding realization in Spanish.

References

[1] Allen, James F. and Ferguson, George (1994): "Actions and events in interval temporal logic". Technical report 521. URL http://www.cs.rochester.edu/u/james/.

[2] Baumann, Stefan; Brinckmann, Caren; Hansen-Schirra, Silvia; Kruijff, Geert-Jan; Kruijff-Korbayová, Ivana; Neumann, Stella; Steiner, Erich; Teich, Elke and Uszkoreit, Hans (2004): "The MULI Project: Annotation & Analysis of Information Structure in German & English". In: Proceedings of LREC 2004, Lisbon.

[3] Lambrecht, Knud (1994): "Information structure and sentence form. Topic, focus, and the mental representation of discourse referents". Cambridge University Press.

[4] Sasaki, Felix (2004): "Secondary Information Structuring - A Methodology for the Vertical Interrelation of Information Resources". In: Proceedings of Extreme Markup Languages 2004, Montreal.

[5] Sasaki, Felix; Witt, Andreas and Metzing, Dieter (2003): "Declarations of relations, differences and transformations between theory-specific treebanks: a new methodology". In: Proceedings of Second Workshop on Treebanks and Linguistic Theories (TLT 2003), Växjö, Sweden.

[6] Witt, Andreas (2004): "Multiple hierarchies: new aspects of an old solution". In: Proceedings of Extreme Markup Languages 2004, Montreal.