Disambiguating Compound Nouns for a Dynamic HPSG Treebank of Wall Street Journal Texts

Valia Kordoni, Yi Zhang

German Research Center for Artificial Intelligence (DFKI GmbH)
Department of Computational Linguistics, Saarland University
P.O.Box 15 11 50, D-66041
Saarbrücken, Germany
{*kordoni,yzhang*}*@coli.uni-sb.de*

Abstract

The aim of this paper is twofold. We focus, on the one hand, on the task of dynamically annotating English compound nouns, and on the other hand we propose disambiguation methods and techniques which facilitate the annotation task. Both the aforementioned are part of a larger on-going effort which aims to create HPSG annotation for the texts from the Wall Street Journal (henceforward WSJ) sections of the Penn Treebank (henceforward PTB) with the help of a hand-written large-scale and wide-coverage grammar of English, the English Resource Grammar (henceforward ERG; Flickinger (2002)). As we show in this paper, such annotations are very rich linguistically, since apart from syntax they also incorporate semantics, which does not only ensure that the treebank is guaranteed to be a truly sharable, re-usable and multi-functional linguistic resource, but also calls for the necessity of a better disambiguation of the internal (syntactic) structure of larger units of words, such as compound nouns, since this has an impact on the representation of their meaning, which is of utmost interest if the linguistic annotation of a given corpus is to be further understood as the practice of adding interpretative linguistic information of the highest quality in order to give "added value" to the corpus.

1. Introduction

The annotation and disambiguation of the English compound nouns we focus on here is part of an on-going project whose aim is to semi-automatically produce rich syntactic and semantic annotations for the WSJ sections of the PTB (Marcus et al. (1993)). The task is being carried out with the help of the ERG, which is a hand-written grammar for English in the spirit of the framework of Head-driven Phrase Structure Grammar (henceforward HPSG; Pollard and Sag (1994)). Its ultimate aim is to overcome the numerous known limitations and shortcomings which are inherent in manual corpus annotation efforts, such as the German Negra/Tiger Treebank ((Brants et al., 2002)), the Prague Dependency Treebank ((Hajič et al., 2000)), the TüBa-D/Z[1], etc., all of which have clearly over the years stimulated research in various sub-fields of computational linguistics where corpus-based empirical methods are used, but as natural in manual annotation efforts they have been also proven to be very time-consuming and error-prone procedures. Specifically, the so-called *dynamic* treebank development procedure we follow is based on the use of automatic parsing outputs as guidance. Many state-of-the-art parsers are able nowadays to efficiently produce large amounts of annotated syntactic structures with relatively high accuracy. This approach has changed the role of human annotation from a labour-intensive task of drawing trees from scratch to a more intelligence-demanding task of correcting parsing errors, or eliminating unwanted ambiguities (cf., the Redwoods Treebank (Oepen et al., 2002)), and should be differentiated from so-called treebank conversion approaches, which are mainly based on the conversion back and forth of different structures, in often different formats, from one linguistic framework to another, with the potentially missing rich annotations filled in incrementally and semi-automatically. Known examples of these latter treebank conversion approaches are the conversions of the WSJ sections of the PTB to annotations in the style of Dependency Grammar, CCG, LFG and HPSG, where the influence of the original PTB annotations and the assumptions implicit in the conversion programs have made the independence of such new treebanks at least questionable. Consequently, in this *dynamic* treebank[2] development setup we are working in, where, as also mentioned above, a linguistically-driven correction of parsing errors and the elimination of unwanted ambiguities are of utmost importance, the parsing disambiguation model and its training is a central vital component with huge impact on the treebanking task itself. To demonstrate its importance and role, we use in this paper the sub-task of the annotation and disambiguation of English compound nouns in the WSJ sections of the PTB as our case study.

[1] http://www.sfs.nphil.uni-tuebingen.de/en_tuebadz.shtml

2. Compound Nouns

2.1. Background & Motivation of the Complexity of the Disambiguation Task

Disambiguating compounds is a challenging task for several reasons. The first challenge lies in the fact that the formation of compounds is highly productive. This is not

[2] The treebank under construction in this project is in line with the so-called dynamic treebanks (Oepen et al., 2002). We rely on the HPSG analyses produced by the ERG, and manually disambiguate the parsing outputs with multiple annotators. The development is heavily based on the DELPH-IN (http://www.delph-in.net/) software repository and makes use of the ERG, the PET parser ((Callmeier, 2001)), an efficient unification-based parser which is used in our project for parsing the WSJ sections of the PTB, and [incr tsdb()] ((Oepen, 2001)), the grammar performance profiling system we are using, which comes with a complete set of GUI-based tools for treebanking. Version control system also plays an important role in this project.

only true for English, but for most languages in which compounds are found. Secondly, the disambiguation of compounds is particularly tricky in English, because there are no syntactic and hardly any morphological cues indicating the relation between the nouns: as has very often to date been proposed in the relevant literature, the nouns are connected by an implicit semantic relation. Being a true Natural Language Processing task, the third difficulty in compound noun disambiguation lies in ambiguity. One could say that compound nouns are double ambiguous: a compound may have more than one possible implicit relation. Therefore, the interpretation of the compound may also depend on context and pragmatic factors. The last main challenge lies in the fact that, even though finite sets of possible relations have been proposed (by among others (Levi, 1978), (Warren, 1978)), there is no agreement on the number and nature of semantic relations that may be found in compounds. Since (Downing, 1977), it is generally assumed that theoretically, the number of possible semantic relations is infinite.

2.2. Annotation of Compound Nouns in the WSJ

The annotation of the WSJ sections of the PTB as a whole, and thus also of the compounds in this text collection, is organised into iterations of parsing, treebanking, error analysis and grammar/treebank update cycles.

Parsing Sentences from the WSJ are first parsed with the PET parser (Callmeier, 2001) using the ERG. Up to 500 top readings are recorded for each sentence. The exact n-best-first parsing mode guarantees that these recorded readings are the ones that have "achieved" the highest disambiguation scores according to the currently in-use parse selection model, without enumerating through all possible analyses.

Treebanking The parsing results are then manually disambiguated by the human annotators. However, instead of looking at individual trees, the annotators spend most of their effort making binary decisions on either accepting or rejecting constructions. Each of these decisions, called discriminants, reduces the number of the trees satisfying the constraints (cf., Figure 1).

Every time a decision is made, the remaining set of trees and discriminants are updated simultaneously. This continues until one of the following conditions is met: i) if there is only one remaining tree and it represents a correct analysis of the sentence, the tree is marked as "gold"; ii) if none of the remaining trees represents a valid analysis, the sentence will be marked as "rejected", indicating an error in the grammar[3]; iii) if the annotator is not sure about any further decision, a "low confidence" state will be marked on the sentence, saved together with the partial disambiguation decisions. Generally speaking, given n candidate trees, on average $\log_2 n$ decisions are needed in order to fully disambiguate. Given that we set a limit of 500 candidate readings per sentence, the whole process should require no more than 9 decisions. If both the syntactic and the semantic analyses look valid, the tree is recorded as the gold reading for the sentence.

[3]In some cases, the grammar does produce a valid reading, but the disambiguation model fails to rank it among the top 500 recorded candidates. In practice, we find such errors occurring frequently during the first annotation circle, but they diminish quickly when the disambiguation model gets updated.

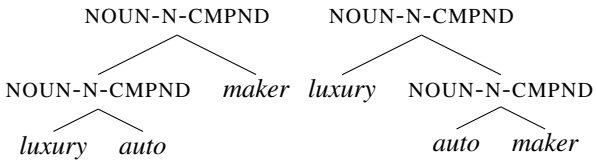


Figure 2: Two alternative analyses from the ERG

Grammar & Treebank Update While the grammar development is independent to the treebanking progress, we periodically incorporate the recent changes of the grammar into the treebank annotation cycle. When a grammar update is incorporated, the treebank also gets updated accordingly by i) parsing anew all the sentences with the new grammar; ii) re-applying the recorded annotation decisions; iii) re-annotating those sentences which are not fully disambiguated after step ii. The extra manual annotation effort in treebank update is usually small when compared to the first round annotation. Another type of update happens more frequently without extra annotation cost. When a new portion of the corpus is annotated, this is used to retrain the parse disambiguation model. This is expected to improve the parse selection accuracy and reduce the annotation workload.

2.3. Examples of Compound Nouns in the WSJ

Being a collection of financial articles, the WSJ may not represent the English language in its most typical daily usage, but it is not in short of interesting linguistic phenomena. Having an average sentence length of over 20 words, loaded with tons of jargons in the financial domain, the corpus puts many natural language processing components (POS taggers, chunkers, NE recognizers, parsers) to the ultimate test. On the other hand, rich phenomena included in the corpus make it also interesting to test deep linguistic processing techniques. One particularly frequent and puzzling phenomenon in the corpus is the vast amount of compound nouns whose syntactic and semantic analyses are potentially ambiguous. Being symbolic systems, deep grammars like the ERG will not always disambiguate all the possibilities. For example, for the compound *"luxury auto maker"*, the ERG will assign both left-branching and right branching analyses (as shown in Figure 2), using the very unrestricted compounding rule NOUN-N-CMPND.

In some cases such branching decisions seem arbitrary and are defensible either way, but there are instances where a distinction should be made clearly. Consider the following two sentences from the WSJ section 3 of the PTB:

- *A form of asbestos once used to make **Kent cigarette filters** has caused a high percentage of cancer deaths among a group of workers exposed to it more than 30 years ago, researchers reported.*
- *Lorillard Inc., the unit of New York-based Loews Corp. that makes Kent cigarettes , stopped using crocidolite in its **Micronite cigarette filters** in 1956.*

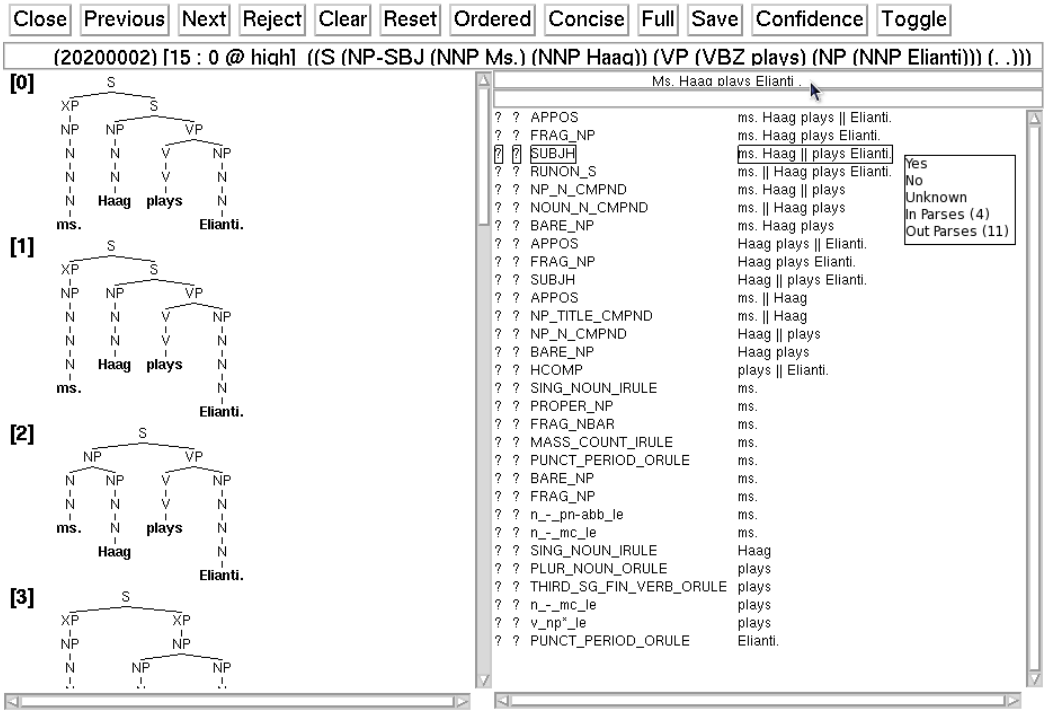


Figure 1: Treebanking Interface with an example sentence, candidate readings, discriminants and the MRS. The top row of the interface is occupied by a list of functional buttons, followed by a line indicating the sentence ID, number of remaining readings, number of eliminated readings, annotator confidence level, and the original PTB bracket annotation. The left part displays the candidate readings, and their corresponding IDs (ranked by the disambiguation model). The right part lists all the discriminants among the remaining readings. The lower part shows the MRS of one candicate reading.

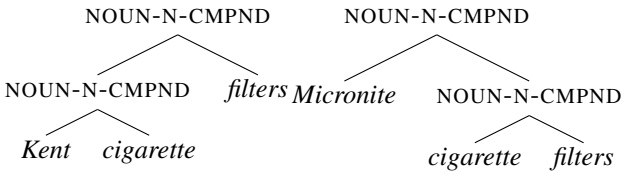


Figure 3: Similar noun compounds with different branching preferences

In some cases, such branching preferences can be easily accounted for, if part of the compound is a multi-word named entity, as in "Fortune 500 executives" and "auto maker Mazda Motor Corp.", where the words from the named entity should be grouped together.

More challenging cases come from the financial domain specific terminologies. While the majority of such terminologies conform to the largely right-branching structures of English, there are cases where left-branching structures may not be excluded in the analysis of the given compounds.

- *Nevertheless , said Brenda Malizia Negus, editor of Money Fund Report , yields " may blip up again before they blip down " because of recent rises in* ***short-term interest rates***.
- *Newsweek said it will introduce the* ***Circulation Credit Plan*** *, which awards space credits to advertisers on "renewal advertising."*

While varying branching preference can hopefully be recovered partially by a statistical disambiguation model trained on the increasing number of manually disambiguated compounds in the treebanking project, there are also problems which need special treatment in the design of features for the disambiguation model. For instance, in a compound construction containing a deverbal noun, the predicate-argument relation from the deverbal noun to the other noun in the compound is left underspecified by the grammar, for the relation can be either an argument or a modifier. Consider the compound *"stock purchase and sales"*. A valid syntactic analysis (as shown in Figure 4) leaves an unbound semantic relation.

Ideally, in this example the semantic variables *i1* and *i2* should be both bound to *x1*. But resolving such an ambiguity within the grammar involves the risk of wrongly assigning the semantic roles in cases where, say, the first noun is serving as modifier instead of argument of the deverbal noun. The current disambiguation model does not recover such a kind of underspecified semantic information, as the model is trained exclusively on disambiguated treebanked data with underspecified semantics unchanged. Furthermore, such disambiguation requires a big number of bi-lexical preferences, in order, for instance, for the distinction between arguments and modifiers to be drawn clearly.

3. Disambiguation of Compound Nouns

Due to the lack of constraints on compound nouns in the ERG, the grammar tends to generate all possible internal structures to these NPs, leading to an combinatorial explosion to the number of candidate trees. In our treebanking

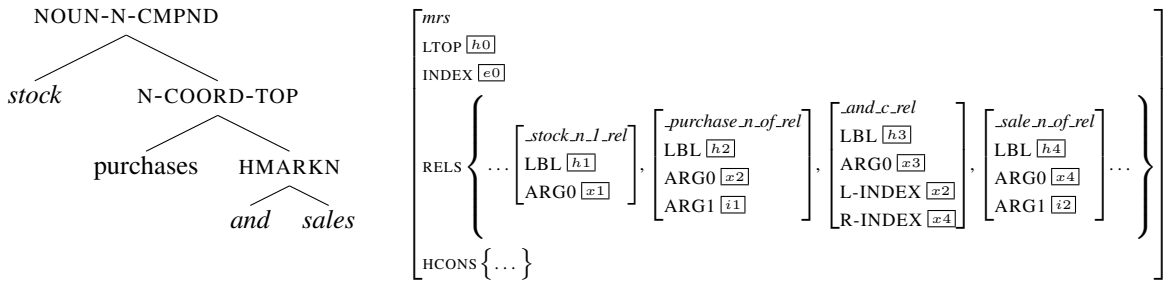


Figure 4: Missing semantic relation within a compound

project, we delay the decision on these internal structures of compounds until the other parts of the syntactic structures are disambiguated. Then the annotators go on to pick the preferred branching structures in line with the examples shown in the previous section. At the moment of writing, the HPSG treebanking project has completed the annotation of 12 WSJ sections from the PTB, which is about half-way through the entire corpus.

The human annotators are assisted with several disambiguation models that help to rank the readings and treebanking decisions. The annotators are warned to make use of this help with cautiousness. The inter-annotator agreements are checked periodically to ensure the quality of the annotation.

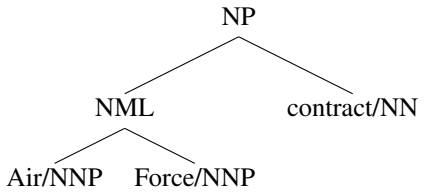
In need to further facilitate and boost the performance of the parse disambiguation model currently used for the annotation of compounds in the WSJ sections of the PTB, we are adopting the following two strategies:

1. Use the annotated sections of the WSJ to retrain the parse disambiguation model and improve the syntactic bracketing prediction accuracy. The parse disambiguation model we are using here is that proposed in (Toutanova et al., 2002) and (Toutanova et al., 2005) which has been developed for use with so-called dynamic treebanking environments, like the Redwoods treebank (Oepen et al., 2002). In such a model, features such as local configurations (i.e., local subtrees), grandparents, n-grams, etc., are extracted from all trees and used to build and (re-)train the model. Thus, as part of this procedure for our purposes, the eligible compound noun candidates are introduced to the ERG-based parsing and treebanking procedure and they get validated through annotation by the human annotators before ultimately being used for the retraining of the inherent to the parser disambiguation model. The ultimate aim here as part of future research is to incorporate the fine-grained treebanking decisions made by the human annotators as discriminative features for the automatic parse disambiguation of the compounds in the WSJ sections of the PTB.

2. Use external large corpora to gather bi-lexical preference information as auxiliary features for the maximum-entropy based parse disambiguation model mentioned above. This is similar to the approach taken in (Johnson and Riezler, 2000) and (van Noord, 2007), where pointwise mutual information association scores are used in order to measure the strength of selectional restrictions and their contribution to parse disambiguation. Because the association scores are estimated on the basis of a large corpus that is parsed by a parser and is aimed at getting improved through parse disambiguations, this technique may be described as a particular instance of self-training, which has been shown in the literature to serve as a successful variant of self-learning for parsing, as well. The idea that selection restrictions may be useful for parsing is not new. In our case at hand, i.e., the case of the disambiguation of compound nouns that we are interested in here, our approach and method is very much fine-tuned and targeted to the disambiguation of argument vs. modifier relations in the compound nouns.

4. Outlook

In the recent work of (Vadas and Curran, 2007), efforts to enrich the noun phrase annotations for the Penn Treebank have been reported. The extra binarization of the originally flat NP structures provides more information for the investigation of the internal structures of the compound nouns, although the enriched annotation adds very little information to the labels, and the semantic relations within the NPs are not explicitly revealed. More specifically, the work of (Vadas and Curran, 2007) leaves the right-branching structures (which are the predominant cases for English) untouched, and just inserts labelled brackets around left-branching structures. Two types of new labels were assigned to these new internal nodes of the PTB NPs: NML or JJP, depending on whether each time the head of the NP is a noun or an adjective. Hence, in this analysis, for instance, the NP "Air Force contract" would receive the following structure:



As a consequence of such an annotation and treatment, *Air Force* as a unit is serving the function of the nominal modifier of *contract*.

Such enriched annotation enables one to investigate the

bracketing preferences within the nominal phrases which was not available with the original PTB. By adapting the existing parsing models to use the enriched annotation, one can expect a fine-grained parsing result. Furthermore, this allows one to explore the treatment of NP in linguistically deep frameworks (see (Vadas and Curran, 2008) for an example of such study in the framework of Combinatory Categorial Grammar (CCG)).

In our ongoing HPSG treebanking project, we aim to development linguistic analyses independent from the PTB annotations. In the same spirit, we have decided not to incorporate the NP bracketing dataset from (Vadas and Curran, 2007) directly during the annotation phase of the project. On the other hand, as pointed out by the original PTB developers ((Marcus et al., 1993)), asking annotators to directly annotate the internal structure of the base-NP significantly slows down the annotation process. We have made a similar observation in our HPSG treebanking project. To help improve the annotation speed while maintaining quality, we periodically update the statistical models that re-rank the candidate trees and discriminants (binary decisions to be made by human annotators) so that the manual decision making procedure is made easier.

As an immediate next step in the research carried out for the dynamic annotation and disambiguation of English compound nouns in the WSJ sections of the PTB described here, we plan to compare the bracketing agreement with the NP dataset from (Vadas and Curran, 2007).

Acknowledgments

The second author thanks the German Excellence Cluster of Multimodal Computing and Interaction for the support of the work.

5. References

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The tiger treebank. In *Proceedings of the workshop on treebanks and linguistic theories*, pages 24–41.

Ulrich Callmeier. 2001. Efficient parsing with large-scale unification grammars. Master's thesis, Universität des Saarlandes, Saarbrücken, Germany.

Pamela Downing. 1977. On the creation and use of English compound nouns. *Language*, 53:4:810–842.

Dan Flickinger. 2002. On building a more efficient grammar by exploiting types. In Stephan Oepen, Dan Flickinger, Jun'ichi Tsujii, and Hans Uszkoreit, editors, *Collaborative Language Engineering*, pages 1–17. CSLI Publications.

Jan Hajič, Alena Böhmová, Eva Hajičová, and Barbora Vidová-Hladká. 2000. The Prague Dependency Treebank: A Three-Level Annotation Scenario. In A. Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, pages 103–127. Amsterdam:Kluwer.

Mark Johnson and Stefan Riezler. 2000. Exploiting auxiliary distributions in stochastic unification-based grammars. In *Proceedings of the 1st NAACL conference*, pages 154–161, Seattle, USA.

Judith Levi. 1978. *The Syntax and Semantics of Complex Nominals*. Academic Press, INC, New York.

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

Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. 2002. The LinGO Redwoods treebank: motivation and preliminary applications. In *Proceedings of COLING 2002: The 17th International Conference on Computational Linguistics: Project Notes*, Taipei, Taiwan.

Stephan Oepen. 2001. [incr tsdb()] — competence and performance laboratory. User manual. Technical report, Computational Linguistics, Saarland University, Saarbrücken, Germany.

Carl J. Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago, USA.

Kristina Toutanova, Christoper D. Manning, Stuart M. Shieber, Dan Flickinger, and Stephan Oepen. 2002. Parse ranking for a rich HPSG grammar. In *Proceedings of the 1st Workshop on Treebanks and Linguistic Theories (TLT 2002)*, pages 253–263, Sozopol, Bulgaria.

Kristina Toutanova, Christoper D. Manning, Dan Flickinger, and Stephan Oepen. 2005. Stochastic HPSG parse selection using the Redwoods corpus. *Journal of Research on Language and Computation*, 3(1):83–105.

David Vadas and James Curran. 2007. Adding noun phrase structure to the penn treebank. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 240–247, Prague, Czech Republic.

David Vadas and James R. Curran. 2008. Parsing noun phrase structure with CCG. In *Proceedings of ACL-08: HLT*, pages 335–343, Columbus, Ohio, June. Association for Computational Linguistics.

Gertjan van Noord. 2007. Using self-trained bilexical preferences to improve disambiguation accuracy. In *Proceedings of the Tenth International Conference on Parsing Technologies*, pages 1–10, Prague, Czech Republic.

Beatrice Warren. 1978. *Semantic Patterns of Noun-Noun Compounds*. Acta Universitatis Gothoburgensis, Göteborg.