# Estimating the Utility of Simplified Discriminants in Grammar-Based Treebanking

Arne Skjærholt and Stephan Oepen

Language Technology Group, Department of Informatics
University of Oslo
E-mail: {arnskj,oe}@ifi.uio.no

**Abstract**

We investigate different types of discriminants in grammar-based treebanking, grounded in a review of earlier work; with an eye towards annotation by non-experts, we propose different simplifications of common discriminant types and quantify their 'discriminative power' as well as efficacy in the preparation of training data for a discriminative parse ranker.

## 1 Introduction

So-called *discriminants*, first suggested by Carter [2], are a vital concept in grammar-driven treebanking (van der Beek et al. [1], Oepen et al. [7], Rosén et al. [9]), enabling annotators to easily select the correct parse from a parse forest of hundreds or even millions of candidate parses. This power stems from the fact that discriminants represent localized 'atomic' ambiguities (individual factors in combinatorial explosion) and, thus, allow the annotator to perform what is essentially a binary search over the parse forest, requiring only $O(\log_2 n)$ decisions to fully disambiguate a forest of $n$ trees; for example, disambiguating a million trees can be accomplished through only approximately 20 decisions.

The first application of discriminants to treebanking is the work of Carter [2], whose TreeBanker presents several kinds of discriminant to the user: word senses (for example *serve* in the sense of *fly to* vs. *provide*), labelled phrase structure spans and sentence type (the special case of a labelled span covering the entire input), semantic triples of two word senses connected by a preposition or conjunction, and specific grammar rules used to build a constituent.

The next application of discriminants for treebanking is the Alpino Dependency Treebank of Dutch (van der Beek et al. [1]), which is couched in the framework of HPSG. In this case, annotators could choose between lexical (coarse-grained PoS tags), unlabelled constituent, and dependency path discriminants. The Alpino dependency paths are paths from the root of the tree to either a word or a phrase, and these discriminants are additionally pruned to only show the shortest paths. That is, if two discriminants decide between the exact same sets of trees, the discriminant that has the shorter path will be preferred.

At roughly the same time and also working in HPSG, Oepen et al. [7] identified four types of discriminants in building the LinGO Redwoods treebank: the lexical type of a token (a fine-grained PoS tag including information about argument structure), the identity of the HPSG construction applied to a span, the semantic predicate associated with a constituent, and simplified constituent labels in terms of 'traditional' phrase structure categories. In more recent Redwoods development, only the first two types were used. Later, a third type of discriminant was added: predicate–argument triples extracted from the underspecified logical forms generated by the grammar (Oepen and Lønning [6]).

Finally, discriminants have been applied to treebanking using LFG grammars by Rosén et al. [9]. They identify four discriminant types: lexical (corresponding to PoS ambiguities), morphological (corresponding to homographs and morphological syncretism), c-structure (ordinary phrase structure), and f-structure (corresponding to discriminating values in syntactico-semantic feature structures).

There is clearly considerable framework-specific variation in the details of discriminant-based annotation, but nevertheless discriminants can be grouped into four broad categories: lexical information, syntactic constituents (either labelled or unlabelled), syntactic dependencies, and semantic predicate–argument information. PoS information can be considered a special case of syntactic constituents of one word, but considering them a separate class is beneficial for the annotators as ambiguities involving a single word are usually very easy to decide (van der Beek et al. [1], Rosén et al. [9]).

However all of these applications have in common that they are intended for relatively well-trained annotators, with the goal of efficiently finding a single gold-standard tree among the trees in the parse forest.[1] In this paper, with an eye towards reducing annotation costs, we investigate the potential of only using only simpler discriminants. While these discriminants do not, in the general case, allow an annotator to recover a single correct parse, they do allow an annotator to decide important classes of ambiguity. In return for this loss of precision, we get an annotation problem that is significantly simplified, allowing us to tap a wider pool of annotators.

## 2 Simplified HPSG Discriminants

We take as our point of departure the LinGO Redwoods syntactic discriminants. As mentioned above, there are two predominant types of these discriminants: lexical

[1] While some discriminant-based annotation tools in fact operate directly over the packed parse forest, others actually require extracting a (possibly partial) list of full parses prior to discrimnant extraction and annotation. Although important technically and conceptually, this distinction has no immediate consequences for our experiments.

types of individual words, and grammatical constructions applied to spans. Figure 1 shows what is known as the *derivation trees* of both analyses licensed by the (1212 version of the) LinGO English Resource Grammar (ERG; Flickinger [3]). Here preterminal nodes are labelled with lexical types and the remaining internal nodes contain the construction applied at that constituent. Together with a copy of the grammar used to parse the sentence, this information enables us to reconstruct the full HPSG feature structure corresponding to that particular parse.

In Figure 1, nodes that correspond to discriminants are highlighted in bold face. In total there are 11 such spans, 4 in the topmost tree (which is the gold tree in the treebank), and 7 in the bottom one. These discriminants are both very specific and very general. The lexical types are highly specialised, encoding not only part of speech, but information such as argument selection (for example, v_np*_le in Figure 1 designates a verb that takes an optional nominal complement); the LinGO ERG contains some 1200 different lexical types. The syntactic rules however, as a consequence of HPSG being a highly lexicalised theory, are in the main comprised of general construction types such as the subject–head and head–complement rules (sb-hd_mc_c and hd-cmp_u_c) at the top of the tree in Figure 1; the ERG contains some 220 such constructions.

In this paper we consider a number of different simplified discriminants, derived from the standard types. The first two types are lexical in nature. An obvious first choice here is the lexical types of the grammar. We do not consider these particularly useful for a wider pool of annotators however, and rather we study this type to see how it compares with a simplified set of lexical types where all additional information (argument preferences, etc.) is stripped, yielding a coarse-grained part-of-speech tagset similar to that of Petrov et al. [8]. These simplified tags are capable of deciding between important classes of ambiguity, such as the noun vs. verb ambiguity of the word *saw*, but not the lemma ambiguity of the same word between the present tense of *saw* and the past tense of *see*.[2]

A slightly more complex kind of discriminant is phrasal discriminants. We consider three discriminants in this class. The first of these is simply unlabelled spans, i.e. bracketing a sequence of tokens as a constituent (of an arbitrary category). While clearly not able to handle all classes of ambiguity, important cases such as PP and other modifier attachments can be disambiguated using such discriminants. For example, whether “the man in the park” is a constituent or not decides between high and low attachment in the case of “I saw a man in the park”.

A slightly more complex discriminant type is labelled spans. In this case, the labels are not individual constructions of the grammar, but rather a simplified set of phrase structure labels like S, VP, NP, etc. This is clearly a more powerful type of discriminants, as the distinction between a modifier PP and a selected-for PP is not discernible without bracket labels. The third and final type of phrasal

[2] LFG morphological discriminants do distinguish the two possible lemmas; however these are not directly portable to HPSG as inflectional morphology is handled by unary rules in the lower layers of the tree. In Figure 1, the rule v_3s-fin_olr corresponds to the present tense inflection of *plays*.

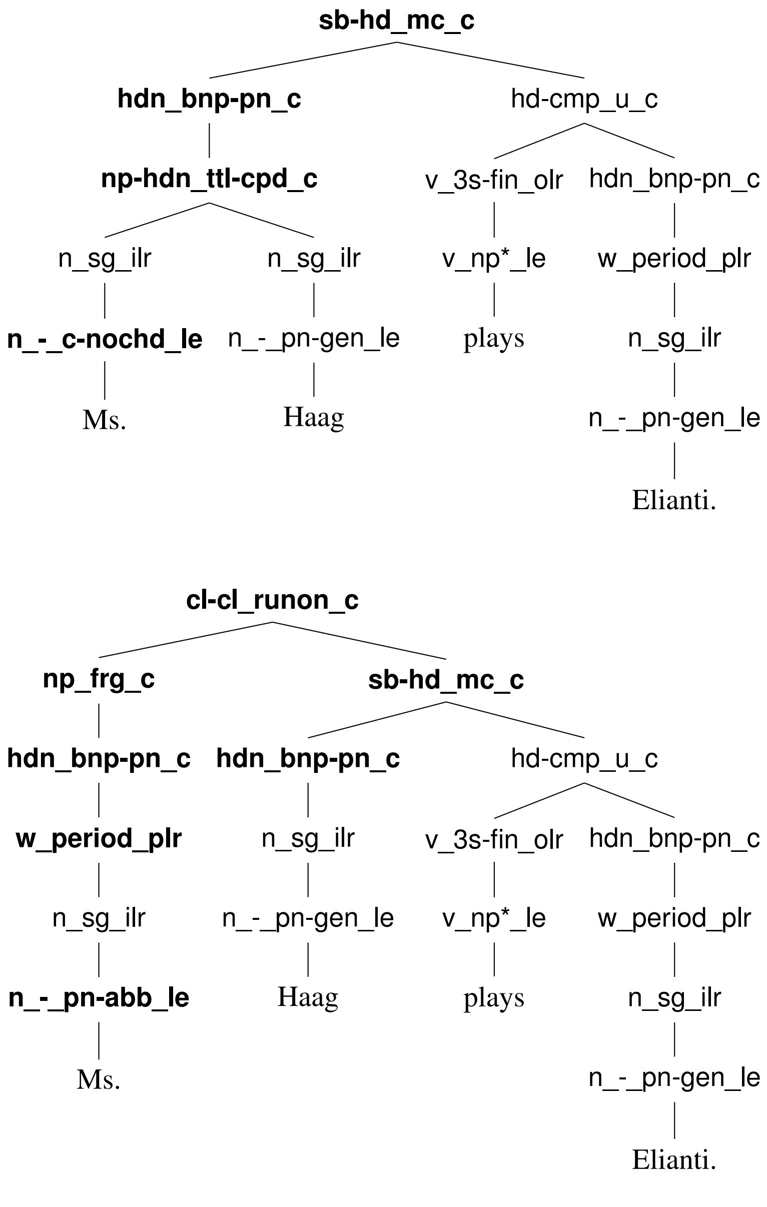

Figure 1: ERG derivation trees for our running example.

discriminant is also labelled, but with a slightly simplified label set compared to that just described. In the topicalized variant “In the park, I see a man.” the phrase “I see a man” receives the label S/PP (denoting a sentence containing a ‘gap’ for the extracted PP). In this final type we strip the trailing slash category,

## 3 Experimental Protocol

We will evaluate different types of simplified discriminants both intrinsically and extrinsically, using the DeepBank (Flickinger et al. [4]) reannotation of the venerable Wall Street Journal corpus in the LinGO Redwoods framework.

For the intrinsic evaluation we will compute what we term the *discriminative potential* for each type of discriminant. Using gold-standard DeepBank annotations, we can construct an oracle to decide whether a discriminant is good (the correct tree has this property) or bad (the correct tree does not have this property). Then, for each sentence in the corpus we can compute the ratio of the number of trees removed in the presence of the oracle ($r$) to the number of non-gold analyses generated by the grammar ($g$). ‘Strong’ discriminant types will score higher, as they are able to prune away a larger fraction of non-gold trees than the less powerful discriminants. We will then evaluate the discriminative potential of a discriminant type as the mean $r/g$ over the DeepBank corpus. Additionally we will take note of the number of sentences that can be fully disambiguated by a discriminant and the number of sentences where no distinctions can be made.

Our extrinsic evaluation metric will be the performance of parse rankers trained on partially disambiguated data. In the LinGO ecosystem, a presumed correct parse is selected from the parse forest generated by the grammar by a discriminative maximum entropy ranker, as described by Toutanova et al. [11]. Normally the parse ranker is trained on fully disambiguated sentences, but it is equally possible to train on a partially disambiguated forest. Partially disambiguated training data will obviously make available to the ranker less information, but it will nevertheless convey important information about preferred vs. dispreferred parse types, especially for discriminant types that are able to prune away large parts of the forest.

We will create the partial forests using essentially the same technique as we use to compute the discriminative potential of a discriminant type, marking parses that are excluded by the discriminant oracle as dispreferred and leaving the remainder of the parses as preferred. We will then use the resulting modified treebanks, DeepBank Sections 00 through 20, to train parse rankers, and evaluate them on Section 21 using common metrics for this problem, the fraction of sentences where the correct parse is ranked the highest (sentence accuracy), and the mean ParseEval score when comparing the top-ranked parse with the gold parse from the treebank.

| Type | Mean (%) | Median (%) | Complete | None |
|---|---|---|---|---|
| Labelled span | 96.8 | 99.4 | 6 745 | 312 |
| Simple labelled span | 96.3 | 99.2 | 5 724 | 367 |
| Unlabelled span | 90.6 | 96.6 | 1 458 | 898 |
| Simple lexical | 53.0 | 57.7 | 410 | 2 930 |
| Lexical type | 86.3 | 92.6 | 2 323 | 397 |

Table 1: Discrimination rates on WSJ00–19

## 4 Results

The results of our intrinsic evaluation are shown in Table 1; to avoid artificially inflating the values, we do not count sentences where all trees licensed by the grammar are marked as gold. This leaves us with a total of 33650 out of 34105 sentences in the first 20 sections of DeepBank. The distribution of the values themselves are not terribly surprising: the more information, the better the discrimination rate. As shown by the median values, the distributions are clearly not normal, with a small peak caused by the sentences where no disambiguation is possible.

There is also a dramatic drop when going from the very detailed lexical types of the ERG to the simplified PoS tagset, from an average 86% for the full lexical types to 53% for the simple tagset. Still structural knowledge is more powerful, with the unlabelled spans outperforming the full lexical types by some 5 percentage points. Structure is still more important to syntax than detailed lexical information. Also of some interest is the difference (or lack thereof) between the full and simple labelled span types; there is some benefit from the slashes in the labels, but the drop in mean discrimination is only about half a percentage point. Still, the difference in fully disambiguated sentences is 1000, about 3% of the corpus.

The results of the extrinsic evaluation are shown in Table 2, with the correlations between discrimination rate and ranker performance shown in Figure 2. There is a very marked drop going from the 'baseline' ranker, trained on the fully disambiguated treebank, to even the ranker trained on data disambiguated by the labelled span discriminant. Once again, the simple and full labelled span discriminant are neck and neck in performance, and likewise the unlabelled spans and full lexical types being relatively similar. The simple lexical types are, as expected, quite a ways behind the other types.

It appears that much of the information required for high ranker performance may be in the very fine distinctions discernible only in a fully disambiguated treebank, but in contrast to the 'baseline' ranker we have yet to tune the hyper-parameters of the models trained on partially disambiguated treebanks.[3] One possi-

[3] For experiments on the scale reported here, exploring the space of plausible hyper-parameters in the discriminative learning set-up is computationally rather costly. For the current results, we merely applied the hyper-parameters found by Zhang et al. [12] for training on fully disambiguated data.

| Type | SA (%) | PE (%) |
|---|---|---|
| Baseline | 39.5 | 96.8 |
| Labelled span | 16.9 | 86.6 |
| Simple labelled span | 15.4 | 86.2 |
| Unlabelled span | 10.4 | 81.7 |
| Simple lexical | 5.45 | 64.5 |
| Lexical type | 9.61 | 72.9 |

Table 2: Extrinsic evaluation results. Sentence accuracy (SA) and ParsEval (PE) scores.

ble interpretation of this is that the information required to eliminate clearly wrong interpretations of a sentence are relatively easy to acquire. The finer distinctions on the other hand, such as choosing between high and low attachment for prepositional phrases is far harder to come by. This tendency is reflected in the correlation curve, where better training data has relatively little impact on performance, until the critical point of about 95% discrimination is reached, at which point ranker performance sky-rockets.

## 5 Conclusions and Future Work

In our estimation, simplified discriminants clearly have the potential to be a useful tool in grammar-driven treebanking, enabling the use of annotators without years of experience in syntactic theory and the particular grammar used. Furthermore, knowing the relative strengths of the different kinds of discriminant should have implications in the design of treebanking tools. To our knowledge, there have been no formal studies of the impact of user interface on annotation efficiency, but just like preprocessing quality can have an important impact on speed (cf. Fort and Sagot [5] for morphological annotation and Skjærholt [10] for syntax) it should be possible to leverage this information in order to make grammar-based annotation more efficient. And while our experiments are grounded in the LinGO ecosystem of HPSG tools, we believe these results should generalise well to other formalisms.

The parse ranker results are less satisfying so far. While we did hypothesise a non-linear correlation between discrimination, the extreme effects we did observe are something of a disappointment (but see Footnote 3). While there is some potential for improved results with a more tailored approach to the ranker learning, the general shape of the learning curve is not likely to change appreciably. Thus, it is not likely that partially disambiguated data alone is enough to train an adequate parse ranker. However, there is some potential for the use of partially disambiguated data as additional data in a domain adaptation setting.

There are several interesting avenues of further work following on this. First of all, it remains to be determined whether the trends observed in our extrinsic

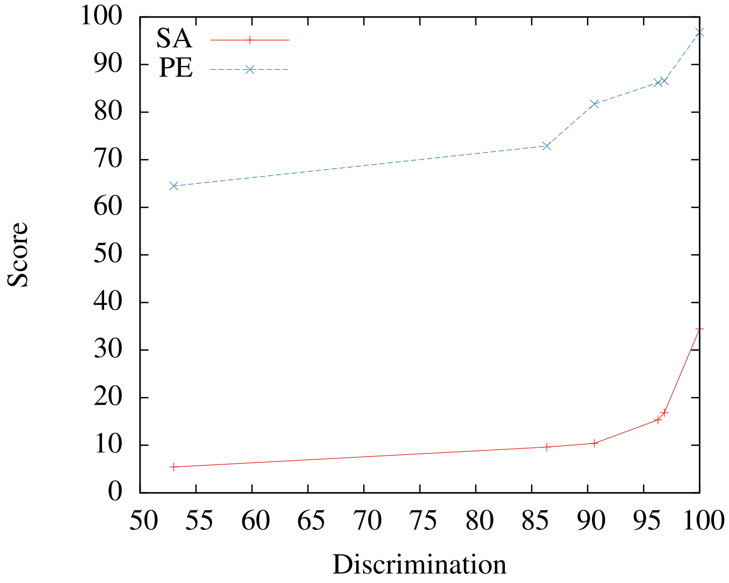

Figure 2: Correlation between disambiguation and ranker performance

evaluation remain true once we complete tuning of hyper-parameters in training from partially disambiguated treebanks. Second, it would be interesting to compare these results with similar discriminant types in other frameworks, and in particular how discriminants like LFG's morphological discriminants, not applicable in the exact same form to HPSG, compare to the types covered in this work. Third, we have not investigated the interaction of these simplified discriminants. For example, it would be very interesting to see how the combination of simplified lexical types and unlabelled spans perform. We did not perform these experiments as our experiments are computationally quite resource-intensive, and constraints on both time and available compute power necessitated a slightly limited scope. Finally, and arguably most importantly, we will seek to shed light on how easy or difficult different discrimant types are to judge reliably by non-experts, e.g. undergraduate students and ultimately crowd-sourcing workers.

## Acknowledgements

We are grateful to our colleagues at the Oslo Language Technology Group and within the larger Deep Linguistic Processing with HPSG Initiative (DELPH-IN) for many fruitful discussions, as well as to three anonymous reviewers for insightful comments. In particular, we would like to thank Johan Benum Evensberget for his input, and Dan Flickinger for making available the grammar and treebanks. Large-scale experimentation and engineering is made possible through access to the ABEL high-performance computing facilities at the University of Oslo, and we are grateful to the Scientific Computing staff at UiO, as well as to the Norwegian Metacenter for Computational Science and the Norwegian tax payer.

## References

[1] Leonoor van der Beek, Gosse Bouma, Rob Malouf, and Gertjan van Noord. The Alpino dependency treebank. In Mariët Theune, Anton Nijholt, and Hendri Hondorp, editors, *Computational Linguistics in the Netherlands 2001. Selected papers from the Twelfth CLIN Meeting*. Rodopi, Amsterdam, The Netherlands, 2002.

[2] David Carter. The TreeBanker. A tool for supervised training of parsed corpora. In *Proceedings of the Workshop on Computational Environments for Grammar Development and Linguistic Engineering*, page 9 – 15, Madrid, Spain, 1997.

[3] Dan Flickinger. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6 (1):15 – 28, 2000.

[4] Dan Flickinger, Yi Zhang, and Valia Kordoni. DeepBank. A dynamically annotated treebank of the Wall Street Journal. In *Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories*, page 85 – 96, Lisbon, Portugal, 2012. Edições Colibri.

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

[6] Stephan Oepen and Jan Tore Lønning. Discriminant-based MRS banking. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, page 1250 – 1255, Genoa, Italy, 2006.

[7] Stephan Oepen, Daniel Flickinger, Kristina Toutanova, and Christopher D. Manning. LinGO Redwoods. A rich and dynamic treebank for HPSG. *Research on Language and Computation*, 2(4):575 – 596, 2004.

[8] Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, page 2089 – 2096, Istanbul, Turkey, May 2012.

[9] Victoria Rosén, Paul Meurer, and Koenraad De Smedt. Designing and implementing discriminants for LFG grammars. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the 12th International LFG Conference*, Stanford, USA, 2007.

[10] Arne Skjærholt. Influence of preprocessing on dependency syntax annotation: Speed and agreement. In *Proceedings of the Seventh Linguistic Annotation Workshop and Interoperability with Discourse*, page 28 – 32, Sofia, Bulgaria, 2013.

[11] Kristina Toutanova, Christopher D. Manning, Dan Flickinger, and Stephan Oepen. Stochastic HPSG Parse Disambiguation using the Redwoods Corpus. *Research on Language and Computation*, 3:83 – 105, 2005.

[12] Yi Zhang, Stephan Oepen, and John Carroll. Efficiency in unification-based n-best parsing. In *Proceedings of the 10th International Conference on Parsing Technologies*, page 48 – 59, Prague, Czech Republic, July 2007.