A Hybrid Constituency-Dependency Parser for Swedish

Johan Hall
Växjö University
jha@msi.vxu.se

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

Jens Nilsson
Växjö University
jni@msi.vxu.se

Abstract

We present a data-driven parser that derives both constituent structures and dependency structures, alone or in combination, in one and the same process. When trained and tested on data from the Swedish treebank Talbanken05, the parser achieves a labeled dependency accuracy of 82% and a labeled bracketing F-score of 75%.

1 Introduction

Most natural language parsers use representations that are based on constituency or dependency. While the relative merits of constituency and dependency representations are still a matter of debate, it is quite clear that they provide partly independent, complementary views of syntactic structure. It is therefore increasingly common that syntactic representations combine elements of both, in particular in annotation schemes for treebanks, such as the TIGER Treebank for German (Brants et al., 2002), the Alpino Treebank for Dutch (Van der Beek et al., 2002), and Talbanken05 for Swedish (Nivre et al., 2006c).

However, there are not many parsers available that can produce hybrid constituency-dependency representations. Widely used statistical parsers, like those of Collins (1997; 1999) and Charniak (2000) output a pure constituency representation (despite making heavy use of lexical dependencies for internal processing) and have to rely on post-processing to add information about grammatical functions (Blaheta and Charniak, 2000). More recently, Gabbard et al. (2006) have shown how a version of the Collins parser can be used to derive the full Penn Treebank annotation including both constituent structure and grammatical function tags. It is also worth mentioning that many grammar-driven parsers, based on frameworks such as LFG (Riezler et al., 2002) and HPSG (Toutanova et al., 2002), produce representations that combine elements of constituency and dependency.

In this paper, we show how hybrid representations can be parsed in a dependency-based encoding inspired by Collins (1999). We evaluate the technique using an existing data-driven dependency parser (MaltParser), trained and tested on Swedish treebank data (Talbanken05). The results show that it is possible to derive hybrid constituency-dependency representations with only a marginal loss in accuracy compared to pure representations of either kind.

The rest of the paper is structured in the following way. Section 2 introduces hybrid constituency-dependency representations, and section 3 describes the dependency-based encoding and parsing strategy adopted in this paper. Section 4 presents the results of the experimental evaluation, and section 5 contains our conclusions.

2 Hybrid Representations

A constituent structure representation for a sentence $w_1, \ldots, w_n$ typically consists of a rooted tree where leaf nodes are labeled with the words $w_1, \ldots, w_n$ and internal nodes are labeled with constituent categories, as illustrated in figure 1.

A dependency structure representation instead consists of a rooted tree where *all* nodes are labeled with the words $w_1, \ldots, w_n$ and edges are labeled

Joakim Nivre, Heiki-Jaan Kaalep, Kadri Muischnek and Mare Koit (Eds.)
NODALIDA 2007 Conference Proceedings, pp. 284–287

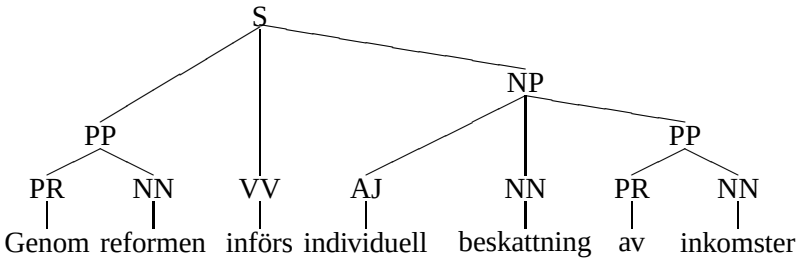

Figure 1: Constituent structure for Swedish sentence

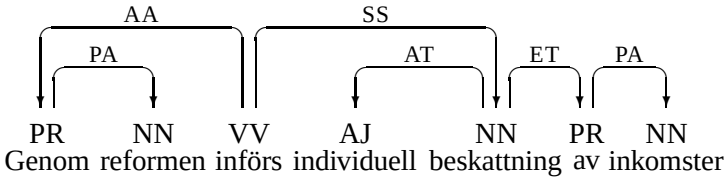

Figure 2: Dependency structure for Swedish sentence

with dependency types, as seen in figure 2.

In the general case, there is no simple mapping from constituent structures to dependency structures or vice versa, especially not if non-projective dependencies are permitted (which correspond to discontinuous constituents). But under certain conditions it is possible to merge the two types of representations into one. Let w_j^* be the substring $w_i, \ldots, w_k$ of the sentence such that all the words in $w_i, \ldots, w_k$ are dominated by w_j in the dependency representation (where dominance is the reflexive and transitive closure of the edge relation). Then, the two representations can be merged if, for every word w_j, w_j^* is the yield of some nonterminal N_j in the constituency representation. In the merged representation, the dependency label of the incoming edge of w_j is added to the incoming edge of the corresponding nonterminal N_j, while other nonterminals get their incoming edge labeled HD (for *head*). Figure 3 shows the hybrid representation obtained by merging the representations in figures 1 and 2.

3 Dependency-Based Hybrid Parsing

Hybrid representations can be parsed in a variety of ways. In this paper, we investigate a dependency-driven approach, where hybrid representations are encoded as dependency structures, by extending the dependency label l_j on the incoming edge to w_j into $l_j|N_j$, if the corresponding nonterminal N_j is not a preterminal, and into $l_j|*$ otherwise. This dependency encoding of the hybrid representation is illustrated in figure 4.

Given such an encoding, any dependency parser can be used to derive hybrid representations. However, in order for the dependency structure output by the parser to be mappable to the desired hybrid representation, we must impose an additional constraint on the relation between the constituent structure and the dependency structure, namely that only preterminal nodes in the constituent structure may have a yield that does not coincide with the complete projection of a lexical head w_j^*. (Note that we can also derive a pure dependency representation or a pure constituency representation by omitting the second half or the first half of the labels, respectively.) In the experiments below, we use the freely available MaltParser (Nivre et al., 2006a) to evaluate this parsing scheme.

4 Experimental Evaluation

The data for the experiments are taken from the professional prose section of the Swedish treebank Talbanken05 (Nivre et al., 2006c), derived from the older Talbanken76 (Einarsson, 1976), developed at Lund University in the 1970s. More precisely, we use the Deepened Phrase Structure version of the

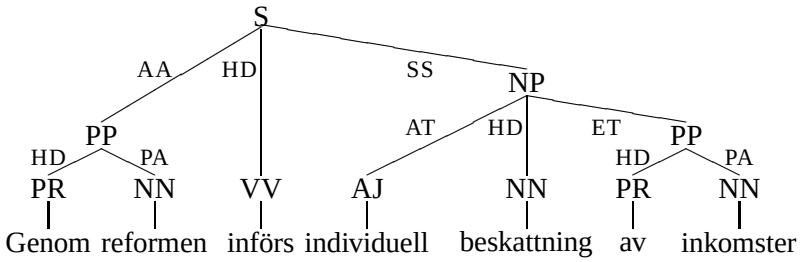

Figure 3: Hybrid representation

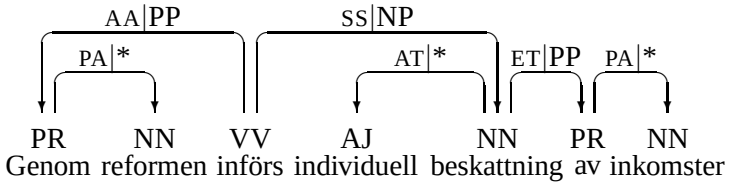

Figure 4: Dependency encoding of hybrid representation

treebank, which combines constituency and dependency annotation in a way that satisfies the constraints discussed in sections 2 and 3.

The data from the professional prose section (roughly 100,000 tokens) were first converted to a dependency-based encoding, as defined in section 2, with three different sets of labels:

1. Constituency only (C)
2. Dependency only (D)
3. Constituency + Dependency (C+D)

While it is only the composite encoding (C+D) that allows the target representation to be derived, the pure constituency (C) and dependency (D) versions are useful for comparison.[1]

The data were then split into 80% for training, 10% for development, and 10% for final testing, and MaltParser was trained on the three versions of the training set (C, D, and C+D). Parsing accuracy was evaluated using two sets of evaluation metrics:

1. The labeled (LR, LP, LF) and unlabeled (UR, UP, UF) recall, precision, and F-measure, as implemented in the evalb software (Sekine and Collins, 1997), measure the percentage of correct constituents in relation to true constituents (recall) and output constituents (precision), with the F-measure being the harmonic mean of recall and precision. As is customary, these measures are reported both for sentences up to 40 words and sentences up to 100 words.

2. The labeled (LAS) and unlabeled (UAS) attachment score, as implemented in the official scoring software of the CoNLL-X shared task (Buchholz and Marsi, 2006), measure the percentage of tokens that have the correct head and (if labeled) the correct dependency relation.

Note that the constituency-based evaluation metrics (LR, LP, LF, UR, UP, UF) can only be meaningfully applied to representations C and C+D, while the dependency-based metrics (LAS, UAS) are only applicable to representations D and C+D.

The results of the evaluation on the final test set are found in table 1. We see that the best dependency accuracy (LAS = 82.43%, UAS = 88.93%) is obtained with the pure dependency representation (D), but we also see that the drop in accuracy when requiring the parser to derive constituent structure as well is less than one percentage point for LAS (81.48%) and only 0.4 percentage points for UAS (88.53%). For constituency, the difference is a little greater, with a drop of about 1.5 percentage points

[1] Note that it is not possible, in the general case, to derive C and D in parallel and simply merge them, since the two output representations may fail to satisfy the constraints required for merging.

	LAS	UAS	LR	LP	LF	UR	UP	UF	
C			75.94	76.54	76.24	80.51	81.15	80.83	≤ 40
			74.56	75.20	74.88	79.15	79.83	79.49	≤ 100
D	82.43	88.93							
C+D	81.48	88.53	74.62	74.76	74.69	79.26	79.41	79.33	≤ 40
			73.39	73.54	73.47	78.12	78.27	78.19	≤ 100

Table 1: Results of the experimental evaluation

in both labeled and unlabeled F-measure (both for sentences up to 40 words and sentences up to 100 words), and the best result is again obtained with the pure representation (C) (LF = 74.88%, UF = 79.49% for sentences up to 100 words).

The results for dependency accuracy are comparable to the best reported results for Talbanken05. It is a little lower than the top score in the CoNLL-X shared task, but that result was based on a training set twice as large (Buchholz and Marsi, 2006; Nivre et al., 2006b). For constituency parsing there is no previous work on Talbanken05, but the results look promising and can probably be improved with better tuning of the parser.

5 Conclusion

We have presented a novel technique for syntactic parsing with hybrid constituency-dependency representations through dependency-based encodings. The method has been evaluated on Swedish, using an existing data-driven dependency parser. The evaluation shows that hybrid representations can be produced with only a marginal loss in accuracy for dependency and constituency considered separately. With better tuning we believe it will be possible to eliminate this loss and perhaps even achieve better accuracy than for separate constituency and dependency parsing.

References

D. Blaheta and E. Charniak. 2000. Assigning function tags to parsed text. In *Proceedings of NAACL*, 234–240.

S. Brants, S. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. TIGER treebank. In *Proceedings of TLT*, 24–42.

S. Buchholz and E. Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL*, 149–164.

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of NAACL*, 132–139.

M. Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of ACL*, 16–23.

M. Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

J. Einarsson. 1976. Talbankens skriftspråkskonkordans. Lund University, Department of Scandinavian Languages.

R. Gabbard, S. Kulick, and M. Marcus. 2006. Fully parsing the Penn treebank. In *Proceedings of HLT-NAACL*, 184–191.

J. Nivre, J. Hall, and J. Nilsson. 2006a. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC*, 2216–2219.

J. Nivre, J. Hall, J. Nilsson, G. Eryiğit, and S. Marinov. 2006b. Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of CoNLL*, 221–225.

J. Nivre, J. Nilsson, and J. Hall. 2006c. Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *Proceedings of LREC*, 1392–1395.

S. Riezler, M. King, R. Kaplan, R. Crouch, J. Maxwell, and M. Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of ACL*, 271–278.

S. Sekine and M. J. Collins. 1997. The evalb software. http://cs.nyu.edu/cs/projects/proteus/evalb.

K. Toutanova, C. D. Manning, S. M. Shieber, D. Flickinger, and Stephan Oepen. 2002. Parse disambiguation for a rich HPSG grammar. In *Proceedings of TLT*, pages 253–263.

L. Van der Beek, G. Bouma, R. Malouf, and G. Van Noord. 2002. The Alpino dependency treebank. In *Language and Computers, Computational Linguistics in the Netherlands 2001. Selected Papers from the Twelfth CLIN Meeting*, 8–22. Rodopi.