Challenges in Mapping of Syntactic Representations for Framework-Independent Parser Evaluation

Kenji Sagae[1], **Yusuke Miyao**[1], **Takuya Matsuzaki**[1] **and Jun'ichi Tsujii**[1,2,3]
[1]Department of Computer Science, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan 113-0033
[2]School of Computer Science, University of Manchester, UK
[3]NaCTeM (National Center for Text Mining), Manchester, UK
{sagae,yusuke,matuzaki,tsujii}@is.s.u-tokyo.ac.jp

Abstract

We explore some of the issues and challenges created by the incompatibility of diverse representation schemes for syntactic parsing. In particular, we examine the problem of output format conversion for evaluation of parsers that use different formalisms. We discuss recent related efforts, and present an evaluation of different parsers that use representations that vary not only in formalisms, but also in depth of syntactic information. We attempt to compare these parsers in a domain widely used for parser evaluation, the Wall Street Journal section of the Penn Treebank, and in the academic biomedical literature, where the use of parsing technologies is expected to contribute in practical applications, such as information extraction and text mining.

1 Introduction

Several types of approaches to automatic syntactic analysis of natural language have benefited from progress in data-driven techniques for parsing technologies. There are now efficient and accurate broad-coverage parsers based on several different ways of representing syntactic information, from simple dependencies and phrase structures, to linguistic motivated formalisms such as Combinatory Categorial Grammar (Steedman, 2000) and Head-driven Phrase Structure Grammar (Pollard and Sag, 1994). Much of this work has been fueled by the availability of large gold-standard parsed corpora, especially the Penn Treebank (Marcus et al., 1994).

While many parsers work with syntactic representations that result from simple processing of the phrase structure trees in the Penn Treebank, others use significantly different representations. Although this diversity is certainly desirable from the perspectives of both parsing research and the application of parsing technologies in natural language applications, many of the potential benefits on both fronts remain unattained due to the lack of a common ground for different representations that would allow for comparisons among different types of parsers, leading to an informed selection of what approach to deploy in specific practical tasks. Intuitively it may seem that even though syntactic structures may be expressed differently in different parsers, conversion from one parser's format to another's (or conversion to a standardized common format) may be feasible, since the different syntactic representations express much of the same information. In practice, however, this has been shown to be very challenging, and results obtained so far in that line of research indicate that this remains an open issue (Briscoe and Carroll, 2006; Clark and Curran, 2007; Miyao et al., 2007).

In this paper we explore some of the issues and challenges created by the incompatibility of representation schemes for syntactic parsing. We examine the problem of output conversion for evaluation of parsers that use different formalisms. We discuss recent efforts to establish common criteria for parser evaluation, and present a case study involving parsers that use representations that vary not only in formalisms, but also in depth of syntactic information. We attempt to compare these parsers in a domain widely used for parser evaluation, the Wall Street Journal section of the Penn Treebank, and in the academic biomedical literature, where the use of parsing technologies is expected to contribute in practical applications, such as information extraction and text mining.

2 Motivation

Much of the recent progress in parsing research and in the application of syntactic analysis to practical tasks has been focused on approaches that use syntactic representations based on simplifications of the Penn Treebank annotation scheme. The most common of such simplifications include shallow phrase structure trees where empty nodes (which represent syntactic phenomena such as long-distance dependencies and ellipsis) and function tags (which indicate the grammatical or semantic function of specific phrases) are removed (Ratnaparkhi, 1997; Collins 1997; Charniak, 2000; Charniak and Johnson, 2005), and bilexical dependencies obtained from these shallow phrase structure trees (Eisner, 1996; Nivre and Scholz, 2004; McDonald et al., 2005) with the use of a head percolation table (Magerman, 1995; Collins 1999).

While parsing research has undoubtedly benefited from comparisons of parsers on a standard test set using the same evaluation criteria (precision and recall of labeled brackets for phrase structures, and accuracy for dependencies), this practice has in some ways diverted attention from other parsing approaches, which have also progressed significantly (Clark and Curran, 2004; Miyao and Tsujii, 2005; Briscoe et al., 2006), and are capable of identifying deep syntactic information (such as long-distance dependencies) that are ignored by parsers based on the shallow representations derived from the Penn Treebank (PTB). Although deep syntactic parsers based on linguistically rich formalisms such as Head-driven Phrase Structure Grammar (HPSG) and Combinatorial Categorical Grammar (CCG) have achieved high levels of both accuracy and efficiency (Matsuzaki et al., 2007; Clark and Curran, 2004), comparisons with more popular approaches based on simplified PTB representations are difficult and imperfect (Kaplan et al., 2004; Clark and Curran, 2007; Miyao et al., 2007).

3 Recent efforts towards framework-independent deep parser evaluation

Although unlabeled dependency accuracy and labeled bracketing precision and recall of shallow PTB trees are still arguably the most widely recognized evaluation metrics for wide-coverage parsing, many have already recognized that these metrics are too limited and too specific to be applied fairly to many of the deep parsing approaches in recent and current development. Some parser developers have turned to resource-specific evaluations, where results are not directly comparable to most other parsers. There have also been a few attempts at establishing specific syntactic representation formats as the basis for framework-independent parser evaluation. We pay special attention to one such effort, the Grammatical Relation (GR) scheme developed by Carroll et al. (1998), which was carefully designed specifically for parser evaluation. We use the GR scheme in our experiments presented in section 4. In addition to GR, we also examine the use of the Stanford Dependency (SD) scheme, which was largely based on Carroll et al.'s GR scheme, but intended for use in applications, not in evaluation. However, because an automatic conversion procedure from shallow PTB structures to SD is available, SD has recently been used in evaluations of parsers in the biomedical domain (Clegg and Shepherd, 2007; Pyysalo et al., 2007a).

3.1 Resource-specific dependencies

In the context of wide-coverage deep parsing, the *de facto* standard metric for parsing accuracy is precision/recall of labeled dependency relations, such as predicate-argument dependencies (Kaplan et al., 2004; Clark and Curran, 2004; Miyao and Tsujii, 2005). However, dependency relations used to evaluate different parsers are based on each parser's formalism and resources. For example, the PARC 700 DependencyBank (King et al., 2003) was used for the evaluation of LFG parsers (Kaplan et al., 2004; Burke et al., 2004), a CCG treebank (CCGBank) (Hockenmaier and Steedman, 2002) was used for the evaluation of CCG parsing models (Hockenmaier, 2003; Clark and Curran, 2004), and HPSG treebanks, which were created manually (Oepen et al., 2004) or derived from PTB data (Miyao et al., 2005), were used for the evaluation of HPSG parsers (Toutanova et al., 2004; Miyao and Tsujii, 2005; Ninomiya et al., 2007; Sagae et al., 2007). Direct relationships among these different dependency schemes are unclear, and we have no way to perform a fair comparison of these parsers.

3.2 Grammatical Relation (GR) evaluation

Recognizing the shortcomings of the widely used parser evaluation metrics of bracketing precision and recall, Carroll et al. (1998) proposed a *grammatical relation* (GR) scheme as a general parser evaluation framework, carefully designed to test a parser's ability to produce structures from which certain grammatical relations (subject, object, modifier, auxiliary, etc.) can be determined. A gold-standard test set of 500 sentences from the SUSANNE corpus was released initially[1], and has been followed by the a set of 700 sentences from the commonly used test section of the Wall Street Journal section of the PTB (the 700 sentences are the same as in the PARC700 corpus). While the use of this evaluation scheme requires post-processing to the parser's output to extract the GRs used in the scheme, the newer 700-sentence gold-standard corpus has recently been used in the evaluation of a CCG parser (Clark and Curran, 2007), an HPSG parser (Miyao et al., 2007), and the RASP parser (Briscoe and Carroll, 2006). Preiss (2003) conducted a GR evaluation of the Collins (1997) parser and the Charniak (2000) parser, using the older SUSANNE-based corpus.

As an example, the GR annotation of the sentence *Regulators also ordered CenTrust to stop buying back the preferred stock* consists of a list of grammatical relations as follows:

- `(ncsubj ordered Regulators _)`
- `(ncsubj stop CenTrust _)`
- `(ncsubj buying CenTrust _)`
- `(ncmod _ ordered also)`
- `(xcomp to ordered stop)`
- `(xcomp _ stop buying)`
- `(dobj buying stock)`
- `(det stock the)`
- `(passive preferred)`
- `(ncsubj preferred stock obj)`
- `(ncmod _ stock preferred)`
- `(ncmod prt buying back)`
- `(dobj ordered CenTrust)`

The first GR (ncsubj) indicates a non-clausal subject relationship, where *Regulators* is the subject of *ordered*, and the seventh GR indicates that *stock* is the (head of the) direct object of *buying*. For a comprehensive list of GR types and what they represent, see Briscoe (2006).

GR annotations are syntactic in nature, and not intended to evaluate some of the semantic relationships that deep parsers may be able to compute. However, the GR scheme does take into account long-distance dependencies, such as control/raising and wh-movement, which schemes based on shallow PTB trees fail to capture. In the example above, the third GR in the list (ncsubj buying CenTrust _) indicates a control relation (*CenTrust* is the subject of *buying*). Such structures are computed by some parsers based on linguistically motivated formalisms, but not by more popular shallow PTB parsers.

3.3 Stanford Dependency (SD) evaluation

The Stanford Dependency (SD) scheme was originally proposed for providing dependency relations that are more useful for applications than phrase structure trees (de Marneffe et al., 2006). This scheme was designed based on Carroll et al. (1998)'s grammatical relations and King et al. (2003)'s dependency bank, and modified to represent more fine-grained and semantically valuable relations (such as apposition and temporal modification), while at the same time leaving out certain relations that are particularly problematic for the shallow PTB parsers it was intended to work with (such as long-distance dependencies[2]). Although no hand-annotated data is available, a program to convert shallow PTB style phrase structures into SD relations is available as part of the Stanford Parser[3] (Klein and Manning, 2003). That is, in principle, any PTB-style treebank can be converted into SD gold standard data. In practice, however, the conversion from phrase structure trees to SD is only approximate, and converting gold standard phrase structure trees results in only partially correct SD annotations. Unfortunately, the accuracy of these annotations is unknown, since the conversion itself has never been evaluated. This scheme was recently used for the evaluation of shallow PTB-style parsers in the biomedical domain (Clegg and Shepherd, 2007; Pyysalo et al.,

[1] Available for download at http://www.informatics.sussex.ac.uk/research/groups/nlp/carroll/greval.html

[2] Although such structures can be represented according to de Marneffe et al.'s description of SD, they are ignored in practice when conversion from shallow PTB trees is performed.

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

2007a) using GENIA (Kim et al., 2003), and of Link Grammar (Sleator and Temperley, 1993) parsers using BioInfer (Pyysalo et al., 2007b) and GENIA.

The same sentence used as an example for GR annotation (*Regulators also ordered CenTrust to stop buying back the preferred stock*) has the following representation in SD format (as converted automatically from the gold-standard PTB tree by the program provided with the Stanford parser):

- `nsubj(ordered-3, Regulators-1)`
- `advmod(ordered-3, also-2)`
- `dobj(ordered-3, CenTrust-4)`
- `aux(stop-6, to-5)`
- `xcomp(ordered-3, stop-6)`
- `partmod(stop-6, buying-7)`
- `prt(buying-7, back-8)`
- `det(stock-11, the-9)`
- `amod(stock-11, preferred-10)`
- `dobj(buying-7, stock-11)`

Although some of the relations are very similar to those in the GR representation (such as the subject relation between *Regulators* and *ordered*, and the direct object relation between *stock* and *buying*), one interesting difference is that long-distance dependencies are not represented in SD (the subjects of *buying* and *stop* are not represented). Another difference is the relation between *stock* and *preferred*: in the GR representation, *preferred* is considered a passive verb, with *stock* being a complement (a surface subject, but initially an object), while in SD *preferred* is considered simply an adjective of *stock*. Finally, we once again note that while the GR representation was created manually, the SD representation was converted automatically from the gold-standard PTB representation, and in this example we see that *buying* is incorrectly identified in SD as a participial modifier (partmod) of *stop* (while the correct relation is xcomp). As previously mentioned, we are not aware of any attempts to estimate the frequency of conversion errors such as this one.

4 Experiments

Based on the GR and SD proposals for parser evaluation described in section 3, we performed evaluations for different parsers in two different domains: (1) the WSJ section of the PTB, and (2) biomedical abstracts from the GENIA Treebank. Additionally, we also conducted the more common evaluation of precision and recall of shallow PTB labeled brackets in both domains.

4.1 Set-up

Our general approach was to convert the output of each parser into GR, SD and shallow PTB representations. The PTB parsers used in our evaluations were the Charniak (2000) parser, and the Charniak and Johnson (2005) reranking parser[4]. These parsers output shallow PTB phrase structure trees, and conversion to SD is performed with the conversion utility provided with the Stanford parser. The conversion to GR was very difficult, even when the SD output is used as an intermediate format. The deep syntactic parser we used was Enju[5] (Miyao and Tsujii, 2005), which is based on HPSG and outputs both (dependency-like) predicate-argument relations (Miyao, 2007) and phrase structure trees (although these do not follow the PTB scheme for phrase structure trees) in an XML format. Conversion to GR was less problematic than with the shallow PTB parsers, since Enju's syntactic representation is richer, but still quite challenging. This was done by mapping Enju's predicate-argument relations into GRs. Conversion to shallow PTB trees was done by mapping tree patterns from Enju's phrase structure output into the corresponding shallow PTB tree patterns, and conversion to SD was done by first converting Enju's output to shallow PTB format, then applying the same PTB-to-SD utility used with the PTB parsers. In addition to these three parsers, we also report previously published comparable results for other parsers.

4.2 Format conversion

To perform the format conversions as mentioned above, we developed the following converters: SD→GR, HPSG→GR HPSG→PTB. Because a converter from PTB to SD was already available, the three additional converters make it possible to obtain each of the three representations with either the shallow PTB parsers or the HPSG parser. To develop the HPSG→GR and HPSG→PTB converters, we used only gold-standard annotations as

[4] Both are available for download via FTP at ftp.cs.brown.edu/pub/nlparser

[5] Available for download at http://www-tsujii.is.s.u-tokyo.ac.jp/enju

reference. Our HPSG→GR conversion followed a similar methodology as Clark and Curran (2007)'s conversion from the output of their CCG parser to GR. Because both the gold-standard GR-annotated version of the PARC700 corpus and the HPSG Treebank (Miyao et al., 2004) were derived from sentences taken from the WSJ section of the Penn Treebank, we had gold-standard annotations for both formats for a set of 700 sentences. We used the same set of 140 sentences as Clark and Curran for development of conversion rules. We then used the remaining 560 sentences to test the accuracy of the conversion. Table 1 shows the conversion accuracy (from gold-standard HPSG annotations, evaluated on the gold standard GR corpus), and for comparison purposes, the accuracy of Clark and Curran's (C&C) conversion from gold-standard CCG annotations. The HPSG→GR conversion accuracy establishes an upper bound for the performance of Enju in the GR evaluation.

The SD→GR conversion was by far the most problematic. As described in section 3, although SD and GR representations are superficially similar, there are significant differences that make conversion difficult. In addition, unlike in the HPSG→GR conversion, which was developed based on gold-standard sets, the SD→GR conversion was developed using the 700-sentence GR gold-standard, and the same sentences with SD annotations obtained from automatic conversion from gold-standard PTB trees. As discussed in section 3.3, this automatic conversion introduces an unknown number of errors. Additionally, in about 5% of all dependencies, the automatic conversion cannot determine the dependency type, and the SD annotation is left underspecified. One of the differences between SD and GR that makes conversion difficult is that SD structures (like shallow PTB trees) do not include long-distance dependencies. Because GR representation does include them (see the example in section 3.2 and 3.3), an automatic loss in recall in incurred. Another mismatch is that in SD prepositional phrases are not assigned a grammatical role (and are simply attached to a head in a relation named *prep*). The GR scheme, on the other hand, does assign a grammatical function to PPs, most often as adjuncts or complements of verbs or nouns. This results in a significant loss in both precision and recall. Differences that were addressed specifically in the mapping between the two formats include differences in the treatment of copula (SD attaches the verb as a dependent of the predicate nominal, while GR attaches the predicate nominal as a complement of the verb), coordination, and differences in head assignments. The accuracy of SD→GR conversion is also shown in table 1.

Conversion	Precision	Recall	F-score
HPSG→GR	87.49	86.79	87.14
SD→GR	80.84	69.16	74.54
(C&C)CCG→GR	86.86	82.75	84.76

Table 1: Precision, recall and F-score of GR representations obtained with our mapping from the gold-standard HPSG Treebank and SD annotations obtained from gold-standard PTB trees. For comparison, we also include figures for the conversion from the gold-standard CCGbank performed by Clark and Curran (2007), denoted by C&C.

Finally, conversion from HPSG-style phrase structures to shallow PTB phrase structures was developed using the gold-standard trees from the Penn Treebank and the HPSG Treebank. Because the HPSG Treebank includes most of the sentences in the WSJ section of the Penn Treebank, the availability of data for development of the mapping rules was much more favorable than the 140 sentences available for development of the conversions to GR. The abundance of development data and the different nature of the conversion (no need to map to grammatical functions) resulted in much higher accuracy for this type of conversion. Measured in precision, recall and F-score of labeled brackets, gold-standard data from the HPSG Treebank was evaluated against the Penn Treebank at, respectively, 98.12%, 98.07%, and 98.09%.

4.3 WSJ evaluation

We first evaluate the Enju HPSG parser (Enju), the Charniak parser (Ch), and the Charniak and Johnson reranking parser (C&J) on the gold-standard GR test set. As seen in the previous subsection, the upperbounds dictated by conversion accuracy are vastly different for HPSG and PTB parsers. This is not surprising, given how syntax is represented in each of these schemes (predicate-argument relations, included in the output of Enju, are much closer to GRs than PTB phrase structures), and the level of linguistic detail contained in them. Table 2 shows the GR results for each of the three parsers, as measured according to Carroll et al. (1998)'s microaveraged precision, recall and F-score. For comparison, we also include previously published results on the same test set for RASP (Briscoe and Carroll, 2006), and the C&C CCG parser (Clark and Curran, 2007). The results for Enju and the C&C parser are close, and well above the results for the other parsers. It is not surprising that these two parsers do well in this evaluation, since they are deep parsers that work at a finer level of linguistic granularity than the other parsers in table 2. It is, however, somewhat surprising that the results are this close, considering that each parser's formalism is different, and output mapping was done using separate conversion schemes developed by separate groups (but using the same GR development data). We also note that although the results for Ch and C&J seem low, they do appear consistent with results reported by Preiss (2003) and Kaplan et al. (2004) using similar parsers in similar (but not the same) data sets.

In the bracketing evaluation, the usual choice for parsers that output PTB-like trees, both C&J and Ch outperform Enju. The results are in table 3.

Parser	Precision	Recall	F-score
Enju (HPSG→GR)	83.57	81.73	82.64
C&J (PTB→SD→GR)	79.08	67.46	72.81
Ch (PTB→SD→GR)	78.41	67.68	72.65
C&C	82.44	81.28	81.86
RASP	77.66	74.98	76.29

Table 2: GR evaluation results, including previously published results for C&C and RASP.

Parser	Precision	Recall	F-score
Enju (HPSG→PTB)	87.13	87.16	87.14
C&J	91.79	91.16	91.48
Ch	89.88	89.63	89.75

Table 3: Labeled bracketing evaluation.

In the SD evaluation the results for the three parsers are much closer, as seen in table 4, even though the same conversion program was applied to every parser's shallow PTB output, where differences were greater. Unfortunately, it is difficult to say whether this is because some of the difference in bracketing accuracy is not significant to the identification of certain syntactic relationships, or because the SD scheme, which was not designed for parser evaluation, blurs some of the distinctions in the output of the three parsers. For comparison, we also include results obtained with the Stanford parser (Klein and Manning, 2003).

Parser	Precision	Recall	F-score
Enju (HPSG→PTB→SD)	87.13	87.16	87.14
C&J (PTB→SD)	88.36	88.45	88.40
Ch (PTB→SD)	87.05	87.10	87.07
Stanford parser	85.36	83.16	84.25

Table 4: SD evaluation, including results from the Stanford parser (Klein and Manning, 2003), for comparison.

4.4 GENIA evaluation

Because gold-standard GR data is not available in the biomedical domain, our evaluation of parsers on the GENIA Treebank includes only SD and shallow PTB bracketing. Unlike recent evaluations (Clegg and Shepherd, 2007; Pyysalo, 2007a) on data from GENIA, we do not evaluate parsers trained only on the WSJ section of the Penn Treebank. Since the GENIA Treebank is available in both the PTB annotation scheme and the HPSG Treebank annotation scheme, we use parsers trained on GENIA (except for the reranker in C&J, which is trained on WSJ, although the first-pass n-best parser in C&J is trained on GENIA). Sections 1 to 900 were used for training, and 901 to 1050 were used for testing.

Tables 5 and 6 show the results for PTB and SD evaluations, respectively. We also include results for the BioLG parser recently published by Pyysalo et al. (2007a), and results for the Lease and Charniak (2005) parser, as published by Clegg and Shepherd (2007). These parsers used a different portion of the GENIA Treebank for testing (Clegg and Shepherd, 2007).

Parser	Precision	Recall	F-score
Enju (HPSG→PTB→SD)	81.74	81.65	81.70
C&J (PTB→SD)	81.99	81.84	81.96
Ch (PTB→SD)	81.16	81.20	81.18
BioLG*	76.9	72.4	74.6
Lease & Charniak	-	-	77.0

Table 5: SD evaluation on the GENIA Treebank. Figures for BioLG, published by Pyysalo et al (2007), correspond to uncollapsed SD (other parsers' figures correspond to collapsed SD). F-score for the Lease & Charniak parser published by Clegg and Shepherd (2007).

The SD evaluations on WSJ and GENIA (tables 4 and 5) show little difference in the results obtained with the three parsers (Enju, C&J and Ch), even though Enju uses a significantly different parsing approach. In the PTB bracketing evaluation for GENIA (table 6), the difference between Enju and C&J is smaller than in the WSJ evaluation, perhaps because C&J's reranker was not trained on GENIA. It should be noted that Enju is penalized in the conversion from its native output format to PTB, as mentioned in section 4.2.

Parser	Precision	Recall	F-score
Enju (HPSG→PTB→SD)	86.20	81.51	83.79
C&J (PTB→SD)	88.55	82.78	85.56
Ch (PTB→SD)	86.97	81.86	84.34
Lease & Charniak	-	-	80.2

Table 6: Labeled bracketing evaluation on the GENIA Treebank. F-score for Lease & Charniak published by Clegg and Shepherd (2007), using a different test set from the GENIA Treebank.

5 Conclusion

We have explored the issue of evaluation across different parsing frameworks through mapping of parser output to different representations. Our evaluation using Carroll et al.'s GR scheme confirms previous findings that conversion to GR is challenging, even from the output of a deep syntactic parser (but even more so from the output of shallower parsers). We have also found that converting from the phrase structure output of a deep parser to shallow PTB phrase structures can be done with relatively high accuracy. The use of this conversion in evaluations in two domains confirmed the intuition that state-of-the-art deep parsers can produce nearly the same level of accuracy in shallow bracketing as more widely used PTB parsers, while at the same time covering additional syntactic information. We also found that although SD may be more useful in some applications than phrase structures, its use as an evaluation metric added little information when a detailed (GR) and a shallow evaluation (PTB) are already being performed. However, SD might still be valuable for high-accuracy conversion from other dependency-based schemes, as shown by Pyysalo et al. (2007a). Although many open questions remain, the continued investigation into how different parsing approaches can be compared will benefit not just parsing research, but eventually also the applications where syntactic analysis is applied.

Acknowledgements

This work was partially supported by Grant-in-Aid for Scientific Research on Priority Areas (MEXT, Japan).

References

Briscoe, T. 2006. An introduction to tag sequence grammars and the RASP system parser. Computer Laboratory Technical Report 662, University of Cambridge.

Briscoe, T. and Carroll, J. 2006. Evaluating the Accuracy of an Unlexicalized Statistical Parser on the PARC DepBank. In *Proc. COLING/ACL 2006 Poster Session*.

Briscoe, T., Carroll, J. and Watson, R. 2006. The second release of the RASP system. In *Proc. COLING/ACL-06 Demo Session*.

Burke, M., Cahill, A., O'Donovan, R., van Genabith, J. and Way, A. 2004. Evaluation of an automatic annotation algorithm against the PARC 700 Dependency Bank. In *Proc. 9th International Conference on LFG*.

Carroll, J., Briscoe, T. and Sanfilippo, A. 1998. Parser Evaluation: a Survey and a New Proposal. In *Proc. LREC 1998*, pages 447–454.

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

Charniak, E. and Johnson, M. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proc. ACL 2005*.

Clark, S. and Curran, J. 2007. Formalism-Independent Parser Evaluation with CCG and DepBank. In *Proc. ACL 2007*.

Clark, S. and Curran, J. 2004. Parsing the WSJ using CCG and log-linear models. In *Proc. 42nd ACL*.

Clegg, A. B. and Shepherd, A. 2007. Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC Bioinformatics* 8(1), 24.

Collins, M. 1997. Three Generative, Lexicalised Models for Statistical Parsing. In *Proc. 35th ACL*.

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

de Marneffe, M.-C., MacCartney, B. and Manning, C. D. 2006. Generating typed dependency parses from phrase structure parses. In *Proc. LREC 2006*.

Eisner, J. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96).*

Hockenmaier, J. 2003. Parsing with Generative Models of Predicate-Argument Structure. In *Proc. 41st ACL.*

Hockenmaier, J. and Steedman, M. 2002. Acquiring compact lexicalized grammars from a cleaner treebank. In *Proc. LREC-2002*, Las Palmas, Spain.

Kaplan, R. M., Riezler, S., King, T. H., III, J. T. Maxwell and Vasserman, A. 2004. Speed and accuracy in shallow and deep stochastic parsing. In *Proc. HLT/NAACL'04.*

Kim, J. D., Ohta, T., Tateisi, Y. and Tsujii, J. 2003. GENIA corpus — a semantically annotated corpus for bio-textmining. *Bioinformatics* 19, i180–182.

King, T. H., Crouch, R., Riezler, S., Dalrymple, M. and Kaplan, R. M. 2003. The PARC 700 Dependency Bank. In *Proc. LINC'03.*

Klein, D. and Manning, C. D. 2003. Accurate Unlexicalized Parsing. In *Proc. ACL 2003.*

Lease, M. and Charniak, E. 2005. Parsing biomedical literature. In *Proc. of the 2nd IJCNLP*. Korea.

Magerman, D. 1995. Statistical decision-tree models for parsing. In *Proc. of ACL 1995.*

Marcus, Mitchell P., Santorini, Beatrice and Marcinkiewicz, Mary Ann. 1994. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics* 19(2), 313–330.

Matsuzaki, T., Miyao, Y. and Tsujii, J. 2007. Efficient HPSG Parsing with Supertagging and CFG filtering. In *Proc. IJCAI 2007.*

McDonald, R., Pereira, F., Ribarov, K. and Hajic, J. 2005. Non-Projective Dependency Parsing using Spanning Tree Algorithms. In *Proc. of HLT/EMNLP 2005*. Vancouver, BC.

Miyao, Y., Ninomiya, T. and Tsujii, J. 2005. Corpus-oriented Grammar Development for Acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank. In Keh-Yih Su, Jun'ichi Tsujii, Jong-Hyeok Lee and Oi Yee Kwong (eds.), *Natural Language Processing - IJCNLP 2004*, volume 3248 of LNAI, pages 684–693, Springer-Verlag.

Miyao, Y., Sagae, K. and Tsujii, J. 2007. Towards framework-independent evaluation of deep linguistic parsers. In *Proceedings of the 2007 Workshop on Grammar Engineering across Frameworks*. Stanford University.

Miyao, Y. 2007. Enju 2.2 Output Specifications. Technical Report TR-NLP-UT-2007-1, Tsujii Laboratory, University of Tokyo.

Miyao, Y. and Tsujii, J. 2005. Probabilistic disambiguation models for wide-coverage HPSG parsing. In *Proc. ACL 2005*, pages 83–90.

Ninomiya, T., Matsuzaki, T., Miyao, Y. and Tsujii, J. 2007. A log-linear model with an n-gram reference distribution for accurate HPSG parsing. In *Proc. IWPT 2007.*

Nivre, J. and Scholz, M. 2004. Deterministic Dependency Parsing of English Text. In *Proc. COLING 2004.*

Oepen, S., Flickinger, D. and Bond, F. 2004. Towards Holistic Grammar Engineering and Testing — Grafting Treebank Maintenance into the Grammar Revision Cycle. In Proc. *IJCNLP-04Workshop "Beyond Shallow Analyses".*

Pollard, C. and Sag, I. 1994. *Head-driven Phrase Structure Grammar*. Chicago: University of Chicago Press and Stanford: CSLI Publications.

Preiss, J. 2003. Using Grammatical Relations to Compare Parsers. In *Proc. EACL 2003*, pages 291–298.

Pyysalo, S., Ginter, F., Haverinen, K., Laippala, V., Heimonen, J. and Salakoski, T. 2007a. On the unification of syntactic annotations under the Stanford dependency scheme: A case study on BioInfer and GENIA. In *Proc. BioNLP 2007*, pages 25–32.

Pyysalo, S., Ginter, F., Heimonen, J., Bjorne, J., Boberg, J., Jarvinen, J. and Salakoski, T. 2007b. BioInfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(50).

Sampo Pyysalo, Tapio Salakoski, Sophie Aubin, andAdeline Nazarenko. 2006. Lexical adaptation of linkvgrammar to the biomedical sublanguage: a comparativevevaluation of three approaches. *BMC Bioinformatics*, 7(Suppl 3).

Ratnaparkhi, A. 1997. A linear observed time statistical parser based on maximum entropy models. In *Proc. EMNLP 1997.*

Sagae, K., Miyao, Y. and Tsujii, J. 2007. HPSG Parsing with Shallow Dependency Constraints. In *Proc. ACL 2007.*

Sleator, D. and Temperley, D. 1993. Parsing English with a Link Grammar. In *Proc. 3rd IWPT.*

Steedman, M. 2000. *The Syntactic Process*. MIT Press.

Toutanova, K., Markova, P. and Manning, C. 2004. The Leaf Projection Path View of Parse Trees: Exploring String Kernels for HPSG Parse Selection. In *Proc. EMNLP 2004.*