Improving Dependency Parsing with Semantic Classes

Eneko Agirre*, Kepa Bengoetxea*, Koldo Gojenola*, Joakim Nivre+
* Department of Computer Languages and Systems, University of the Basque Country UPV/EHU
+ Department of Linguistics and Philosophy, Uppsala University
{e.agirre, kepa.bengoetxea, koldo.gojenola}@ehu.es joakim.nivre@lingfil.uu.se

Abstract

This paper presents the introduction of WordNet semantic classes in a dependency parser, obtaining improvements on the full Penn Treebank for the first time. We tried different combinations of some basic semantic classes and word sense disambiguation algorithms. Our experiments show that selecting the adequate combination of semantic features on development data is key for success. Given the basic nature of the semantic classes and word sense disambiguation algorithms used, we think there is ample room for future improvements.

1 Introduction

Using semantic information to improve parsing performance has been an interesting research avenue since the early days of NLP, and several research works have tried to test the intuition that semantics should help parsing, as can be exemplified by the classical PP attachment experiments (Ratnaparkhi, 1994). Although there have been some significant results (see Section 2), this issue continues to be elusive. In principle, dependency parsing offers good prospects for experimenting with word-to-word-semantic relationships.

We present a set of experiments using semantic classes in dependency parsing of the Penn Treebank (PTB). We extend the tests made in Agirre et al. (2008), who used different types of semantic information, obtaining significant improvements in two constituency parsers, showing how semantic information helps in constituency parsing.

As our baseline parser, we use MaltParser (Nivre, 2006). We will evaluate the parser on both the full PTB (Marcus et al. 1993) and on a sense-annotated subset of the Brown Corpus portion of PTB, in order to investigate the upper bound performance of the models given gold-standard sense information, as in Agirre et al. (2008).

2 Related Work

Agirre et al. (2008) trained two state-of-the-art statistical parsers (Charniak, 2000; Bikel, 2004) on semantically-enriched input, where content words had been substituted with their semantic classes. This was done trying to overcome the limitations of lexicalized approaches to parsing (Magerman, 1995; Collins, 1996; Charniak, 1997; Collins, 2003), where related words, like *scissors* and *knife* cannot be generalized. This simple method allowed incorporating lexical semantic information into the parser. They tested the parsers in both a full parsing and a PP attachment context. The experiments showed that semantic classes gave significant improvement relative to the baseline, demonstrating that a simplistic approach to incorporating lexical semantics into a parser significantly improves its performance. This work presented the first results over both WordNet and the Penn Treebank to show that semantic processing helps parsing.

Collins (2000) tested a combined parsing/word sense disambiguation model based in WordNet which did not obtain improvements in parsing.

Koo et al. (2008) presented a semisupervised method for training dependency parsers, using word clusters derived from a large unannotated corpus as features. They demonstrate the effectiveness of the approach in a series of dependency parsing experiments on PTB and the Prague Dependency Treebank, showing that the cluster-based features yield substantial gains in performance across a wide range of conditions. Suzuki et al. (2009) also experiment with the same method combined with semi-supervised learning.

Ciaramita and Attardi (2007) show that adding semantic features extracted by a named entity tagger (such as PERSON or MONEY) improves the accuracy of a dependency parser, yielding a 5.8% relative error reduction on the full PTB.

Candito and Seddah (2010) performed experiments in statistical parsing of French, where terminal forms were replaced by more general symbols, particularly clusters of words obtained through unsupervised clustering. The results showed that word clusters had a positive effect.

Regarding dependency parsing of the English PTB, currently Koo and Collins (2010) and Zhang and Nivre (2011) hold the best results, with 93.0 and 92.9 unlabeled attachment score, respectively. Both works used the Penn2Malt constituency-to-dependency converter, while we will make use of PennConverter (Johansson and Nugues, 2007).

Apart from these, there have been other attempts to make use of semantic information in different frameworks and languages, as in (Hektoen 1997; Xiong et al. 2005; Fujita et al. 2007).

3 Experimental Framework

In this section we will briefly describe the data-driven parser used for the experiments (subsection 3.1), followed by the PTB-based datasets (subsection 3.2). Finally, we will describe the types of semantic representation used in the experiments.

3.1 MaltParser

MaltParser (Nivre et al. 2006) is a trainable dependency parser that has been successfully applied to typologically different languages and treebanks. We will use one of its standard versions (version 1.4). The parser obtains deterministically a dependency tree in linear-time in a single pass over the input using two main data structures: a stack of partially analyzed items and the remaining input sequence. To determine the best action at each step, the parser uses history-based feature models and SVM classifiers. One of the main reasons for using MaltParser for our experiments is that it easily allows the introduction of semantic information, adding new features, and incorporating them in the training model.

3.2 Dataset

We used two different datasets: the full PTB and the Semcor/PTB intersection (Agirre et al. 2008). The full PTB allows for comparison with the state-of-the-art, and we followed the usual train-test split. The Semcor/PTB intersection contains both gold-standard sense and parse tree annotations, and allows to set an upper bound of the relative impact of a given semantic representation on parsing. We use the same train-test split of Agirre et al. (2008), with a total of 8,669 sentences containing 151,928 words partitioned into 3 sets: 80% training, 10% development and 10% test data. This dataset is available on request to the research community.

We will evaluate the parser via Labeled Attachment Score (LAS). We will use Bikel's randomized parsing evaluation comparator to test the statistical significance of the results using word sense information, relative to the respective baseline parser using only standard features.

We used PennConverter (Johansson and Nugues, 2007) to convert constituent trees in the Penn Treebank annotation style into dependency trees. Although in general the results from parsing Pennconverter's output are lower than with other conversions, Johansson and Nugues (2007) claim that this conversion is better suited for semantic processing, with a richer structure and a more fine-grained set of dependency labels. For the experiments, we used the best configuration for English at the CoNLL 2007 Shared Task on Dependency Parsing (Nivre et al., 2007) as our baseline.

3.3 Semantic representation and disambiguation methods

We will experiment with the range of semantic representations used in Agirre et al. (2008), all of which are based on WordNet 2.1. Words in WordNet (Fellbaum, 1998) are organized into sets of synonyms, called *synsets* (SS). Each synset in turn belongs to a unique *semantic file* (SF). There are a total of 45 SFs (1 for adverbs, 3 for adjectives, 15 for verbs, and 26 for nouns), based on syntactic and semantic categories. For example, noun semantic files (SF_N) differentiate nouns denoting acts or actions, and nouns denoting animals, among others. We experiment with both full synsets and SFs as instances of fine-grained and coarse-grained semantic representation, respectively. As an example of the difference in these two representations, *knife* in its tool sense is in the EDGE TOOL USED AS A CUTTING INSTRUMENT singleton synset, and also in the ARTIFACT SF along with thousands of other

words including *cutter*. Note that these are the two extremes of semantic granularity in WordNet.

As a hybrid representation, we also tested the effect of merging words with their corresponding SF (e.g. knife+ARTIFACT). This is a form of semantic specialization rather than generalization, and allows the parser to discriminate between the different senses of each word, but not generalize across words. For each of these three semantic representations, we experimented with using each of: (1) all open-class POSs (nouns, verbs, adjectives and adverbs), (2) nouns only, and (3) verbs only. There are thus a total of 9 combinations of representation type and target POS: SS (synset), SS_N (noun synsets), SS_V (verb synsets), SF (semantic file), SF_N (noun semantic files), SF_V (verb semantic files), WSF (wordform+SF), WSF_N (wordform+SF for nouns) and WSF_V (for verbs).

For a given semantic representation, we need some form of WSD to determine the semantics of each token occurrence of a target word. We experimented with three options: a) gold-standard (GOLD) annotations from SemCor, which gives the upper bound performance of the semantic representation, b) first Sense (1ST), where all token instances of a given word are tagged with their most frequent sense in WordNet, and c) automatic Sense Ranking (ASR) which uses the sense returned by an unsupervised system based on an independent corpus (McCarthy et al. 2004). For the full Penn Treebank experiments, we only had access to the first sense, taken from Wordnet 1.7.

4 Results

In the following two subsections, we will first present the results in the SemCor/PTB intersection, with the option of using gold, 1st sense and automatic sense information (subsection 4.1) and the next subsection (4.2) will show the results on the full PTB, using 1st sense information. All results are shown as labelled attachment score (LAS).

4.1 Semcor/PTB (GOLD/1ST/ASR)

We conducted a series of experiments testing:

- Each individual semantic feature, which gives 9 possibilities, also testing different learning configurations for each one.
- Combinations of semantic features, for instance, SF+SS_N+WSF would combine the semantic file with noun synsets and wordform+semantic file.

	System	LAS	
Baseline		81.10	
Gold	SS	81.18	+0.08
	SS_N	81.40	+0.30
	SS_V	*81.58	+0.48
	SF	**82.05	+0.95
	SF_N	81.51	+0.41
	SF_V	81.51	+0.41
	WSF	81.51	+0.41
	WSF_N	81.43	+0.33
	WSF_V	*81.51	+0.41
	SF+SF_N+SF_V+SS+WSF_N	*81.74	+0.64
ASR	SS	81.30	+0.20
	SS_N	*81.56	+0.46
	SS_V	*81.49	+0.39
	SF	81.00	-0.10
	SF_N	80.97	-0.13
	SF_V	**81.66	+0.56
	WSF	81.32	+0.22
	WSF_N	*81.62	+0.52
	WSF_V	**81.72	+0.62
	SF_V+SS_V	81.41	+0.31
1ST	SS	81.40	+0.30
	SS_N	81.39	+0.29
	SS_V	*81.48	+0.38
	SF	*81.59	+0.49
	SF_N	81.38	+0.28
	SF_V	*81.52	+0.42
	WSF	*81.57	+0.46
	WSF_N	81.40	+0.30
	WSF_V	81.42	+0.32
	SF+SS_V+WSF_N	**81.92	+0.81

Table 1. Evaluation results on the test set for the Semcor-Penn intersection. Individual semantic features and best combination.
(**: statistically significant, $p < 0.005$; *: $p < 0.05$)

Although there were hundreds of combinations, we took the best combination of semantic features on the development set for the final test. For that reason, the table only presents 10 results for each disambiguation method, 9 for the individual features and one for the best combination.

Table 1 presents the results obtained for each of the disambiguation methods (gold standard sense information, 1st sense, and automatic sense ranking) and individual semantic feature. In all cases except two, the use of semantic classes is benefi-

	System	LAS	
Baseline		86.27	
1ST	SS	*86.53	+0.26
	SS_N	86.33	+0.06
	SS_V	*86.48	+0.21
	SF	**86.63	+0.36
	SF_N	*86.56	+0.29
	SF_V	86.34	+0.07
	WSF	*86.50	+0.23
	WSF_N	86.25	-0.02
	WSF_V	*86.51	+0.24
	SF+SS_V+WSF_N	*86.60	+0.33

Table 1. Evaluation results (LAS) on the test set for the full PTB. Individual features and best combination.

(**: statistically, $p < 0.005$; *: $p < 0.05$)

cial albeit small. Regarding individual features, the SF feature using GOLD senses gives the best improvement. However, GOLD does not seem to clearly improve over 1ST and ASR on the rest of the features. Comparing the automatically obtained classes, 1ST and ASR, there is no evident clue about one of them being superior to the other.

Regarding the best combination as selected in the training data, each WSD method yields a different combination, with best results for 1ST. The improvement is statistically significant for both 1ST and GOLD. In general, the results in Table 1 do not show any winning feature across all WSD algorithms. The best results are obtained when using the first sense heuristic, but the difference is not statistically significant. This shows that perfect WSD is not needed to obtain improvements, but it also shows that we reached the upperbound of our generalization and learning method.

4.2 Penn Treebank and 1st sense

We only had 1st sense information available for the full PTB. We tested MaltParser on the best configuration obtained for the reduced Semcor/PTB on the full treebank, taking sections 2-21 for training and section 23 for the final test. Table 2 presents the results, showing that several of the individual features and the best combination give significant improvements. To our knowledge, this is the first time that WordNet semantic classes help to obtain improvements on the full Penn Treebank.

It is interesting to mention that, although not shown on the tables, using lemmatization to assign semantic classes to wordforms gave a slight increase for all the tests (0.1 absolute point approximately), as it helped to avoid data sparseness. We applied Schmid's (1994) TreeTagger. This can be seen as an argument in favour of performing morphological analysis, an aspect that is many times neglected when processing morphologically poor languages as English.

We also did some preliminary experiments using Koo et al.'s (2008) word clusters, both independently and also combined with the WordNet-based features, without noticeable improvements.

5 Conclusions

We tested the inclusion of several types of semantic information, in the form of WordNet semantic classes in a dependency parser, showing that:

- Semantic information gives an improvement on a transition-based deterministic dependency parsing.
- Feature combinations give an improvement over using a single feature. Agirre et al. (2008) used a simple method of substituting wordforms with semantic information, which only allowed using a single semantic feature. MaltParser allows the combination of several semantic features together with other features such as wordform, lemma or part of speech. Although tables 1 and 2 only show the best combination for each type of semantic information, this can be appreciated on GOLD and 1ST in Table 1. Due to space reasons, we only have showed the best combination, but we can say that in general combining features gives significant increases over using a single semantic feature.
- The present work presents a statistically significant improvement for the full treebank using WordNet-based semantic information for the first time. Our results extend those of Agirre et al. (2008), which showed improvements on a subset of the PTB.

Given the basic nature of the semantic classes and WSD algorithms, we think there is room for future improvements, incorporating new kinds of semantic information, such as WordNet base concepts, Wikipedia concepts, or similarity measures.

References

Eneko Agirre, Timothy Baldwin, and David Martinez. 2008. Improving parsing and PP attachment performance with sense information. In *Proceedings of ACL-08: HLT*, pages 317–325, Columbus, Ohio.

Daniel M. Bikel. 2004. Intricacies of Collins' parsing model. *Computational Linguistics*, 30(4):479–511.

Candito, M. and D. Seddah. 2010. Parsing word clusters. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Language*, Los Angeles, USA.

M. Ciaramita and G. Attardi. 2007. Dependency Parsing with Second-Order Feature Maps and Annotated Semantic Information, In *Proceedings of the 10th International Conference on Parsing Technology*.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proc. of the 15th Annual Conference on Artificial Intelligence* (AAAI-97), pages 598–603, Stanford, USA.

Eugene Charniak. 2000. A maximum entropy-based parser. In *Proc. of the 1st Annual Meeting of the North American Chapter of Association for Computational Linguistics* (NAACL2000), Seattle, USA.

Michael J. Collins. 1996. A new statistical parser based on lexical dependencies. In *Proc. of the 34th Annual Meeting of the ACL*, pages 184–91, USA.

Michael Collins. 2000. A Statistical Model for Parsing and Word-Sense Disambiguation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge.

Sanae Fujita, Francis Bond, Stephan Oepen, and Takaaki Tanaka. 2007. Exploiting semantic information for HPSG parse selection. In *Proc. of the ACL 2007 Workshop on Deep Linguistic Processing*.

Richard Johansson and Pierre Nugues. 2007. Extended Constituent-to-dependency Conversion for English. In *Proceedings of NODALIDA 2007*, Tartu, Estonia.

Erik Hektoen. 1997. Probabilistic parse selection based on semantic cooccurrences. In *Proc. of the 5th International Workshop on Parsing Technologies*.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08*, pages 595–603, USA.

Terry Koo, and Michael Collins. 2008. Efficient Third-order Dependency Parsers. In *Proceedings of ACL-2010*, pages 1–11, Uppsala, Sweden.

Shari Landes, Claudia Leacock, and Randee I. Tengi. 1998. Building semantic concordances. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, USA.

David M. Magerman. 1995. Statistical decision-tree models for parsing. In *Proc. of the 33rd Annual Meeting of the ACL*, pages 276–83, USA.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2):313–30.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant senses in untagged text. In *Proc. of the 42nd Annual Meeting of the ACL*, pages 280–7, Barcelona, Spain.

Joakim Nivre. 2006. *Inductive Dependency Parsing*. Text, Speech and Language Technology series, Springer. 2006, XI, ISBN: 978-1-4020-4888-3.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel and Deniz Yuret. 2007b. The CoNLL 2007 Shared Task on Dependency Parsing. *Proceedings of EMNLP-CoNL*L. Prague, Czech Republic.

Adwait Ratnaparkhi, Jeff Reynar, and Salim Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *HLT '94: Proceedings of the Workshop on Human Language Technology*, USA.

Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of International Conference on New Methods in Language Processing*. September 1994

Jun Suzuki, Hideki Isozaki, Xavier Carreras, and Michael Collins. 2009. An Empirical Study of Semi-supervised Structured Conditional Models for Dependency Parsing. In *Proceedings of EMNLP*, pages 551–560. Association for Computational Linguistics.

Deyi Xiong, Shuanglong Li, Qun Liu, Shouxun Lin, and Yueliang Qian. 2005. Parsing the Penn Chinese Treebank with semantic knowledge. In *Proc. of the 2nd International Joint Conference on Natural Language Processing* (IJCNLP-05), Korea.

Yue Zhang, and Joakim Nivre. 2011. Transition-Based Parsing with Rich Non-Local Features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.