Semantic Feature Engineering for Enhancing Disambiguation Performance in Deep Linguistic Processing

Danielle Ben-Gera, Yi Zhang, Valia Kordoni

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

Abstract

The task of parse disambiguation has gained in importance over the last decade as the complexity of grammars used in deep linguistic processing has been increasing. In this paper we propose to employ the fine-grained HPSG formalism in order to investigate the contribution of deeper linguistic knowledge to the task of ranking the different trees the parser outputs. In particular, we focus on the incorporation of semantic features in the disambiguation component and the stability of our model cross domains. Our work is carried out within DELPH-IN (http://www.delph-in.net), using the LinGo Redwoods and the WeScience corpora, parsed with the English Resource Grammar and the PET parser.

1. Introduction & Previous Work

Deep parsing of natural language is a difficult task. In order to gain wide meaningful coverage of a given language complex grammars are employed. Grammars such as the English Resource Grammar (henceforward ERG; (Copestake and Flickinger, 2000)) offer a broad coverage analysis of English, but are at the same time faced with challenges, as the number of correct structures that are licensed by the grammar is often vast. Parsers, such as the PET parser (Callmeier, 2000) which are mostly used in combination with the ERG for the processing of large-scale data, often return thousands of different readings for one single sentence. Those readings may differ from one another in significant ways. The task of automatically ranking the different readings and returning the correct one is hence an important one.

Different methods have been suggested in the literature for the task of parse selection, both for more traditional treebanks, such as the Penn Treebank ((Charniak, 1997), (Collins, 1997)), and for so-called dynamic treebanks, like the Redwoods treebank ((Toutanova et al., 2002), (Toutanova et al., 2005), and (Zhang et al., 2007)).

Both generative and discriminative models were employed for the task. In generative models the classifier derives the probability of a certain observation to belong to a certain class ($P(Y|X)$, where Y is the class and X is the observation) using the prior and the joint probability of X and Y. Discriminative models are generally considered a more direct approach, since the conditional probability $P(Y|X)$ is being calculated directly using log-linear models.

As a result of estimating the posterior probability, discriminative models do not assume independence of features. This characteristic makes log-linear models more appealing to our task since linguistics features are often dependent on each other.

The most extensive work using HPSG treebanks was done by (Toutanova et al., 2005). In their work both discriminative and generative models for parse disambiguation are investigated and up to 14% error reduction is achieved when using discriminative models over generative ones with the same features. This supports our choice of using a discriminative model.

In our work we are trying to extend this previous attempt to include more explicit semantic features directly from the full Minimal Recursion Semantics (MRS; (Copestake et al., 2005)) representation. (Toutanova et al., 2005) uses a simplified semantic graph that is extracted from the complete MRS feature structure.

In addition, we put a focus on investigating domain adaptability of our discriminative model. To the best of our knowledge, former approaches that made use of the HPSG formalism were carried out on the Redwoods treebank (Oepen et al., 2002) which is very well adjusted to the grammar and hence might not provide a clear insight to the difficulties of the task. Therefore testing our models on new independent corpora is valuable for the performance evaluation of the method.

The disambiguation of the different results typically takes place offline, on the output of the parser. This approach has the advantage that all the information from the parser was already collected and it is possible to achieve exact ranking of the different parses even with non-local features that are not available during parsing. A considerable disadvantage of this method is that the output of the parser can be partial. It is often the case that due to the complexity of the parsing task, most systems keep only the n-best hypotheses during parsing and potentially discard parses that might be significant to the training process, in particular a correct parse can be lost.

A related concern is the efficiency of the parse selection task. When the parser yields many results, re-ranking can be very complex, especially when taking into account more context and the set of features grows. To overcome this drawback, selective forest unpacking methods have been proposed. Using compact representations it is possible to encode many different derivations in an efficient way. Those forest representations then need to be unpacked for disambiguation to apply (cf., the selective unpacking meth-

ods proposed in ((Carroll and Oepen, 2005) and (Zhang et al., 2007), as well as the discriminative model of (Huang, 2008)).
This paper aims at improving the performance of re-ranking the output of the PET parser. We are using a maximum entropy model that incorporates different properties of each reading to learn a disambiguation model that will allow us to discard unlikely structures. We are mainly interested in the contribution of the semantic components of the sentence to this task. We investigate the performance and robustness of our models on different domains using two different corpora. We show that features extracted from the semantic component increase classification performance and that models which make extensive use of the semantics properties of a sentence are more robust and less domain specific.

2. Semantic Features for Disambiguation in Deep Linguistic Processing

2.1. General Setup

In the work we are reporting here we use a maximum entropy approach to create a model to classify our data. Maximum Entropy aims at creating a stochastic model that reflects the probability of a label given a certain event. It operates under the philosophy that a probability distribution should be as unified as possible given some constraints – if we have no reason to assume otherwise all observation should have the same probability.
We interpret the observed probability distribution of the different features as a set of constraints on the expected value of the model. Maximum entropy can be hence translated to a problem of constrained optimization; maximizing the entropy taking into account the features.
The events are described as a set of features and during training the model calculates weights for those features. After the model has been trained, the prediction of the label for a given event is a simple matter of taking the log of the sum of all the different features times their weights (see 1).

$$p(y|x) = \frac{1}{Z_\lambda(x)} exp(\sum_i \lambda_i f_i(x,y)) \quad (1)$$

The process of training is used to estimate the parameters of the model and in particular the weights vector λ. For training we use two different corpora, LOGON and WeScience, both analyzed with the same underlying grammar (July 2009 ERG) to test our models.
The first corpus is part of the Redwoods treebank which includes three data-sets that are substantially different. The first data-set is Ecom, which includes commercial-related sentences that were collected from users asking questions or giving comments about a product. Naturally, the sentences in this data-set are quite repetitive. The second data-set is the Verbmobil data-set, which includes spoken sentences from the Verbmobil project. Those are typically short sentences that contain fewer ambiguities. Finally, the third data-set is the LOGON data-set, which contains tourism-related sentences that were collected as part of , the Norwegian-English machine translation project. The July 2009 LOGON treebank has 8,927 sentences with an average of 228.65 different readings per sentence and median of 106 reading per sentence. Note that, Redwoods has been a part of the DELPH-IN collaboration for many years and as such it is often used both for development and evaluation of tasks within this framework. This creates a certain bias of the corpus towards the grammar that is used to create the treebank that is reflected in higher much higher results.
The second treebank corpus we are using is the WeScience[1]. This corpus contains articles that were collected from Wikipedia. As such they are diverse in length and number of ambiguities while still mostly keep structured grammatical form. Therefore this corpus is more useful than data randomly collected from the Internet that lacks structure consistency and may be ungrammatical.
WeScience treebank contains 10,146 sentences with an average sentence length of 16.05 words per sentence (about two words longer than the average sentence in) and contains more ambiguities with an average of 270.49 and median of 330 readings per sentence.
Those figures are influenced by the maximum readings that are allowed by the parser. In our setting the upper limit of readings has been set to 1000 after which the parser does not continue to process any more derivations. A gold standard tree is chosen for all sentences, meaning that in some cases a better tree was discarded and the tree that is annotated as the correct one might not be the optimal one that the parser outputs.
The grammar that we use to parse those sentences is the latest ERG release (July 2009). Since the ERG is an on-going developing project, it is necessary to decide and conduct all experiments with the same version of the grammar. of the model across different domains. The ERG contains unary operations that implement morphological changes on the lexicon and binary rules (or schemata) that encode the syntactic structure of the sentence.
One of the greatest advantages of HPSG is the ability to easily integrate different components to the grammar without the need to create a real different layer of analysis. Different levels of language processing can become part of the parse of the sentence simply by translating them into a feature-value matrix and adding the relevant constraints to the grammar. This allows for the same parser (in our case the PET parser) to collect the additional information without any further modification to the parsing method.
For the purpose of this paper we are specially interested in the semantics component of the grammar. The ERG provides, in addition to the typical syntactic layer, detailed semantic information for the sentence using MRS representations.
MRS is a flat semantic representations that can describe sets of semantic interpretations for a syntactic structure in an underspecified way. The framework is fully compositional and MRS structures can be represented as feature structures which can be integrated within the HPSG framework. These two aspects make it easy to describe a syntax-semantic interface for HPSG parsing with MRS.
The MRS language consists of an object language and a meta-language describing how fragments of the object lan-

[1]The corpus is available at http://wiki.delph-in.net/moin/WeScience

guage can be combined. The object language is a specific first order language with generalized quantifiers. The basic units of an MRS representation are called elementary predictions. An elementary predication has the form $label : relation(var_1, ..., var_n, hole_1, ..., hole_n)$, where *label* and $hole_{1n}$ are called handles and constitute part of the meta language. In addition, an MRS structure contains a set of constraints between handles, called qeq-constraints and a unique top-handle. In a configuration of an MRS, handles are identified pairwise such that elementary predictions can be substituted into the hole-handles identified with their labels. This way quantifier scoping is explicitly resolved. However such a structure is only called a configuration of the MRS if it satisfies all qeq-constraints. The qeq-constraint $f =_q g$ is satisfied if either f is identified with g or f outscopes g. The set of configurations for the MRS corresponds to the set of its readings.

Using these treebanks and the grammar we conduct experiments to investigate the influence of adding semantics to the disambiguation model. We conduct each experiment on two random splits and averaged the results. We designed the experiments such that in addition to the question of the influence of adding the semantics to the model, we will also examine the cross-domain stability.

2.2. Baseline

For the analysis of the mentioned data, we have thus created a reliable baseline that allows us to evaluate the contribution of different properties to the task of disambiguation. Following previous work, it is common practice to choose a set of features that includes the derivation trees rules that are being used. Adding a certain level of parenting improves results depending on the data-set. Since the ERG underlying the Redwoods corpus has a maximally binary structure, the result trees are not as flat as in other corpora such the Penn Treebank (Marcus et al., 1994) and do not contain much context within the span of a single rule. This characteristic makes it important to include non-local information such as parenting.

Since no previous baseline exists for the specific data-set, we approach the task by choosing a simple set of features consists of rules up to the third level of parenting. As can be observed in table 1, there is a clear improvement when using 2nd level grandparenting (up to 5% in the same domain). When raising the tree depth to 3-grandparenting level, we can no longer observe an obvious improvement of results. In some cases there is even a drop in the result of the best pick (up to 2% in the LOGON-LOGON experiment).

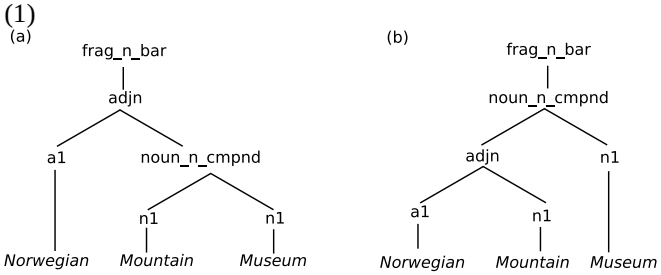
2.3. Deep semantic features

Deeper properties of the derivations must be extracted from the Redwoods treebank using the grammar as a reference. The treebank contains a basic derivation tree, where the nodes are annotated with the rules that have been used to derive its structure. From this schemata information we can reproduce the full feature structure that describes the sentence. The HPSG-based ERG provides MRS semantic representations. In order to have access to those features, we first extract this information from the basic derivation tree that is available in the treebank. Since the derivation tree is labeled with a unique rule that was used to create the node, we can use the grammar to trace back bottom-up all the feature-value pairs and create a full description of the parse including the MRS. This can be achieved using the [incr itsdb()] toolkit[2] which is also part of the DELPH-IN collaboration open source repository.

Looking at the semantics of sentences helps us disambiguate structure where the syntax cannot provide any hints, since more than one reading is syntactically correct. This is often the case with PP attachments or complex noun compounding.

Consider, for instance, the sentence in (1) below. This is an example from the Redwoods treebank. The phrase *Norwegian Mountain Museum* has 29 different derivation trees in the treebank. Here we look at two of them in order to exemplify the difference:

(1)



There is a clear difference of meaning between those two readings. In (2a) the modifier *Norwegian* modifies the noun phrase *Mountain Museum*, while in (b) the adjective modifies only the word *Mountain*. On the syntactic level both those solutions are possible, but looking at the MRS of those two parses we can see a clear difference.

(2) (a)

```
{e2:
 e2:unknown[ARG x4:_museum_n_of]
 e9:_norwegian_a_1[ARG1 x4:_museum_n_of]
 e11:compound[ARG1 x4:_museum_n_of,
              ARG2 x10:_mountain_n_1]
}
```

(b)

```
{e2:
 e2:unknown[ARG x4:_museum_n_of]
 e10:compound[ARG1 x4:_museum_n_of,
              ARG2 x9:_mountain_n_1]
 e15:_norwegian_a_1[ARG1 x9:_mountain_n_1]
}
```

Notice that the MRS representation here is a simplified one that can be viewed as a semantic dependency structure. From those structures we can extract features that model the dependencies. Although the dependency structures tend to be flatter than the syntax trees, one can still often observe some nesting in the structure. Taking this into account allows us to consider wider context for the semantics, in a similar way to the one we use for the syntactic structure, i.e., using various parenting levels. That allows a better

[2] http://wiki.delph-in.net/moin/ItsdbTop

	LO-LO		WS-WS		LO-WS		WS-LO	
	1-best	10-best	1-best	10-best	1-best	10-best	1-best	10-best
p0	56.3480	76.6048	42.9618	67.5953	31.0850	58.0645	38.5164	65.6205
p1	57.0641	80.4564	43.9882	73.3137	32.2580	61.1432	36.8045	67.1897
p2	62.1968	82.5962	46.7741	74.0469	32.2580	61.1432	36.3766	68.3300
p3	60.0570	83.8801	45.8944	74.1935	33.5777	62.3167	35.9486	69.9001

Table 1: Results of using different levels of parenting and grandparenting on different data-sets. In this table *LO* stands for LOGON and *WS* stands for WeScience. *LO-WS* for example means the model was trained on and tested WeScience.

comparison between the contribution of the syntax and the contribution of the semantics.

In addition to this simplified structure we also have access to the full MRS of the sentence. The full structure allows us to extract explicitly information such as the scope of the quantifiers in the sentence. This information is important specially since in the MRS formalism many ambiguities are created due to the fact that every NP is viewed as a scope baring element that can hence overscope, or be overscopped, by other elements. This creates many of possible configurations for a given sentence.

(3)

```
<h1,e2,
{h3:unknown_rel<0:24>(e2, x4),
h5:udef_q_rel<0:24>(x4, h6, h7),
h8:compound_rel<0:24>(e10, x4, x9),
h11:udef_q_rel<0:24>(x9, h12, h13),
h14:_norwegian_a_1_rel<0:8>(e15, x9),
h14:_mountain_n_1_rel<10:17>(x9),
h8:_museum_n_of_rel<19:24>(x4, i16)},
{h6 qeq h8,
h12 qeq h14}>
```

Example (3) is the full MRS of the same sentence (1). The structure includes the top-handle (`h1`), the set of predicates and lastly the set of qeq constraints. In addition, the quantifiers are explicitly mentioned. The quantifier `undef_q_rel` appears twice and takes scope over *mountain* (`x9`) and *museum* (`x4`), we use this information as well in our semantics model.

In order to build a less sparse model that reflects more general observations, we would like to abstract away from the specific semantic relations and use the categories of words which we extract from the lexicon. We are experimenting with different models both with lexical categories and with the word specific semantic relations.

2.4. Evaluation and Error Analysis

We conducted four experiments, two where training and testing is being done on the same domain, and two cross-domain experiments. We evaluated each experiment in terms of best pick, where only if the highest ranked reading is the optimal one it is considered a correct choice by the disambiguator. In addition we also evaluate how often the correct reading is in the top ten ranked parses. Top-10 evaluation is significant since the differences between the readings can be very small (due to the fine-grained analyzes that an HPSG-oriented grammar like the ERG provides), and often human annotators are not fully consistent with minor details. Furthermore one can benefit greatly from a tool that will narrow down the number of possible results from a maximum of 1000 to 10.

(Toutanova et al., 2005) report little improvement when combining semantic properties into the learner. One of the reasons for this is the fact that the results in this work were evaluated on the syntactic structure solely. The manually annotated gold standard was created with the correct syntactic structure as a guideline, mostly ignoring the semantic part. Evaluating the disambiguator on a test set that was annotated with the correct MRS structure in mind might be necessary in order to truly understand the contribution of the semantics.

Part of the goal of our work has also been to conduct a detailed error analysis in order to investigate the contribution of the semantics component to the disambiguation model. We show that using the semantic features we are able to disambiguate a range of phenomena that is beyond the syntactic level. (Toutanova et al., 2005) indicate that a significant portion of real errors, i.e., errors that are not caused by human wrong annotations, or by the grammar licensing impossible derivations, are errors where semantic knowledge is needed. PP-attachment, modifier attachment and even the correct choice of lexical item often require knowledge about the meaning of the sentence.

The semantic model is more stable when applying it to different domains and shows less drop of results when testing the model on a new corpus. The syntax-only model shows a drop of almost 10% between the WeScience-WeScience in domain experiment and the WeScience-LOGON experiment where the learner is being tested on more unseen structures from new corpus. In the model that uses the semantics, however, we see only 7.5% drop.

The semantics of a sentence is in a sense where the essence of sentence is being captured. When we utter a phrase we are really interested in the its meaning. The syntax, or the structure of the sentences, is being employed to serve this purpose of conveying the meaning. In that sense the semantic representation is more domain independent as it is not influenced by a specific structure that might be common in a certain corpus.

2.5. Results and Discussion

In table 2 we can observe the results of the experiments with the semantic models on both the corpora. We can see that introducing the semantic features results with some improvement in almost all of the experiments. From those experiments we can conclude that the semantic features do

	LO-LO		WS-WS		LO-WS		WS-LO	
	1-best	10-best	1-best	10-best	1-best	10-best	1-best	10-best
p3 (baseline)	60.0570	83.8801	45.8944	74.1935	33.5777	62.3167	35.9486	69.9001
sem-d2	40.7988	73.7517	26.0997	61.4369	20.5278	51.7595	23.9657	58.0599
syn+sem	61.7689	83.8801	46.0410	74.9266	35.7771	62.9032	38.5164	69.3295

Table 2: Results of the experiments with semantics-only model and a model that incorporate both the semantics and the syntactic features.

add new information to the model. This additional information is especially helpful where the syntactic features reach their limits.

In order to check the contribution of the semantics to the task of disambiguation, we have trained a model consisting of semantic features only. It is interesting to note that this model preforms much better on the LOGON corpus, with 40% precision in best pick (much higher then a random pick which is 27%). However it does not perform well in the other experiments. We suspect that this is due to the work that was done on LOGON. was used as a development set for building the ERG grammar and in particular the MRS part of the grammar. For that reason we expect higher results when training and testing on this treebank.

The syntax alone with level-three grandparenting gives us precision of about 43% on average. This is better than the semantic-only model, but one should note that the syntax model is able to take into account more of the context using the granparenting. Here too we can observe that the results on the data-set are much higher and exhibit a clear bias.

From table 2 one can draw some general observations about the different data-sets we have used. As mentioned above, the WeSicence corpus is much more diverse and represents data that was collected from the Internet. In addition, no fine tuning of the grammar was done on this corpus, and hence it is a more realistic scenario of using the disambiguator.

It is not surprising that in all the different configurations of the model cross domain testing yields lower results. Furthermore, we can see that training on and testing on WeScience gives the lowest results. Since LOGON is more specific and less diverse the trainer simply does not encounter enough examples to be able to cope with the more complex input of the Wikipedia articles. When we reverse the experiment and train the model on WeScience, we obtain better results (e.g. 38.5% instead of 35.7% in the syntax and semantics model).

The more significant improvement when adding the semantics is apparent in the best pick, where we see more than 1.5% better precision on average. On the other hand the top-10 is not influenced as much. It seems that the additional semantic information helps to disambiguate fine grained information that allows us to re-rank the top candidates in a more precise way.

Examining different domains by using different corpora gives us a much better understanding of the contribution of the various features to the models. The contribution of the semantic features can be observed most clearly in the cross domain experiments where adding the semantics gives up a visible improvement of 2.2% when training on LOGON and 2.6% when training on the WeScience in the best pick evaluation.

In some of the experiments, we have observed that adding more syntactic features (by increasing the parenting level) does not help improve the performance. These data-sets seem to have fully used the information that the syntax can provide us. On those sets, that naturally yield lower baseline results, we see a better performance after the integration of the semantic features into the model.

3. Conclusion and Outlook

From our experiments so far it is clear that adding information about the meaning of the sentence can be beneficial for disambiguating the parser's output. We have shown that the MRS structures that are provided by the grammar can be translated into features that contribute to the disambiguation model.

Moreover, that the same improvement can be observed in cross domain experiments as well. The semantic information consistently helps the model to re-rank the readings and increases the precision.

Another important result of this work is the observation that the LOGON corpus is a very artificial choice to evaluate systems developed using the ERG. We can see that in all our experiments working solely with the LOGON treebank results in very high figures that do not represent the true abilities of the disambiguation model.

4. References

U. Callmeier. 2000. PET–a platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering*, 6(01):99–107.

J. Carroll and S. Oepen. 2005. High efficiency realization for a wide-coverage unification grammar. *Lecture notes in computer science*, 3651:165.

E. Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the National Conference on Artificial Intelligence*, pages 598–603.

M Collins. 1997. Three generative, lexicalised models for statistical parsing. In Jonathan David Bobaljik and Tony Bures, editors, *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 16–23, NJ, USA. ACL.

A. Copestake and D. Flickinger. 2000. An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of LREC*, volume 2.

A. Copestake, D. Flickinger, C. Pollard, and I.A. Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language & Computation*, 3(4):281–332.

L. Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In *In Proceeding of Association for Computational Linguistics*. ACL.

M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz. 1994. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330.

S. Oepen, K. Toutanova, S. Shieber, C. Manning, D. Flickinger, and T. Brants. 2002. The LinGO Redwoods treebank motivation and preliminary applications. In *Proceedings of the 19th international conference on Computational linguistics-Volume 2*, pages 1–5. Association for Computational Linguistics Morristown, NJ, USA.

K. Toutanova, C.D. Manning, S. Shieber, D. Flickinger, and S. Oepen. 2002. Parse disambiguation for a rich HPSG grammar. In *Proceedings of the First Workshop on Treebanks and Linguistic Theories*, pages 253–263.

K. Toutanova, C.D. Manning, D. Flickinger, and S. Oepen. 2005. Stochastic HPSG parse disambiguation using the Redwoods corpus. *Research on Language & Computation*, 3(1):83–105.

Y. Zhang, S. Oepen, and J. Carroll. 2007. Efficiency in unification-based n-best parsing. *IWPT*, page 48.