Using Self-Trained Bilexical Preferences to Improve Disambiguation Accuracy

Gertjan van Noord
University of Groningen
vannoord@let.rug.nl

Abstract

A method is described to incorporate bilexical preferences between phrase heads, such as selection restrictions, in a Maximum-Entropy parser for Dutch. The bilexical preferences are modelled as association rates which are determined on the basis of a very large parsed corpus (about 500M words). We show that the incorporation of such self-trained preferences improves parsing accuracy significantly.

1 Motivation

In parse selection, the task is to select the correct syntactic analysis of a given sentence from a set of parses generated by some other mechanism. On the basis of correctly labelled examples, supervised parse selection techniques can be employed to obtain reasonable accuracy. Although parsing has improved enormously over the last few years, even the most successful parsers make very silly, sometimes embarassing, mistakes. In our experiments with a large wide-coverage stochastic attribute-value grammar of Dutch, we noted that the system sometimes is insensitive to the naturalness of the various lexical combinations it has to consider. Although parsers often employ lexical features which are in principle able to represent preferences with respect to word combinations, the size of the training data will be too small to be able to learn the relevance of such features successfully.

In maximum-entropy parsing, the supervised parsing technique that we use in our experiments, arbitrary features can be defined which are employed to characterize different parses. So it is possible to construct features for any property that is thought to be important for disambiguation. However, such features can be useful for disambiguation only in case the training set contains a sufficient number of occurrences of these features. This is problematic, in practice, for features that encode bilexical preferences such as selection restrictions, because typical training sets are much too small to estimate the relevance of features representing cooccurrences of two words. As a simple example consider the ambiguous Dutch sentence

(1) Melk drinkt de baby niet
Milk drinks the baby not

The standard model of the parser we experimented with employs a wide variety of features including syntactic features and lexical features. In particular, the model also includes features which encode whether or not the subject or the object is fronted in a parse. Since subjects, in general, are fronted much more frequently than objects, the model has learnt to prefer readings in which the fronted constituent is analysed as the subject. Although the model also contains features to distinguish whether e.g. `milk` occurs as the subject or the object of `drink`, the model has not learnt a preference for either of these features, since there were no sentences in the training data that involved both these two words.

To make this point more explicit, we found that in about 200 sentences of our parsed corpus of 27 million sentences `milk` is the head of the direct object of the verb `drink`. Suppose that we would need at least perhaps 5 to 10 sentences in our training corpus

Proceedings of the 10th Conference on Parsing Technologies, pages 1–10,
Prague, Czech Republic, June 2007. ©2007 Association for Computational Linguistics

in order to be able to learn the specific preference between `milk` and `drink`. The implication is that we would need a (manually labeled!) training corpus of approximately 1 million sentences (20 million words). In contrast, the disambiguation model of the Dutch parser we are reporting on in this paper is trained on a manually labeled corpus of slightly over 7,000 sentences (145,000 words). It appears that semi-supervised or un-supervised methods are required here.

Note that the problem not only occurs for artificial examples such as (1); here are a few mis-parsed examples actually encountered in a large parsed corpus:

(2) a. Campari moet **u** gedronken hebben
Campari must have drunk you
You must have drunk Campari

b. De wijn die **Elvis** zou hebben gedronken als hij wijn zou hebben gedronken
The wine Elvis would have drunk if he had drunk wine
The wine that would have drunk Elvis if he had drunk wine

c. De paus heeft **tweehonderd daklozen** te eten gehad
The pope had twohunderd homeless people for dinner

In this paper, we describe an alternative approach in which we employ pointwise mutual information association score in the maximum entropy disambiguation model. Pointwise mutual information (Fano, 1961) was used to measure strength of selection restrictions for instance by Church and Hanks (1990). The association scores used here are estimated using a very large parsed corpus of 500 million words (27 million sentences). We show that the incorporation of this additional knowledge source improves parsing accuracy. Because the association scores are estimated on the basis of a large corpus that is parsed by the parser that we aim to improve upon, this technique can be described as a somewhat particular instance of self-training. Self-training has been investigated for statistical parsing before. Although naively adding self-labeled material to extend training data is normally not succesfull, there have been successful variants of self-learning for parsing as well. For instance, in McClosky et al. (2006) self-learning is used to improve a two-phase parser reranker, with very good results for the classical Wall Street Journal parsing task.

Clearly, the idea that selection restrictions ought to be useful for parsing accuracy is not new. However, as far as we know this is the first time that automatically acquired selection restrictions have been shown to improve parsing accuracy results. Related research includes Abekawa and Okumura (2006) and Kawahara and Kurohashi (2006) where statistical information between verbs and case elements is collected on the basis of large automatically analysed corpora.

2 Background: Alpino parser

The experiments are performed using the Alpino parser for Dutch. In this section we briefly describe the parser, as well as the corpora that we have used in the experiments described later.

2.1 Grammar and Lexicon

The Alpino system is a linguistically motivated, wide-coverage grammar and parser for Dutch in the tradition of HPSG. It consists of over 600 grammar rules and a large lexicon of over 100,000 lexemes and various rules to recognize special constructs such as named entities, temporal expressions, etc. The grammar takes a ‘constructional’ approach, with rich lexical representations and a large number of detailed, construction specific rules. Both the lexicon and the rule component are organized in a multiple inheritance hierarchy. Heuristics have been implemented to deal with unknown words and word sequences, and ungrammatical or out-of-coverage sentences (which may nevertheless contain fragments that are analysable). The Alpino system includes a POS-tagger which greatly reduces lexical ambiguity, without an observable decrease in parsing accuracy (Prins, 2005).

2.2 Parser

Based on the categories assigned to words, and the set of grammar rules compiled from the HPSG grammar, a left-corner parser finds the set of all parses, and stores this set compactly in a packed parse forest. All parses are rooted by an instance of the top category, which is a category that generalizes over all maximal projections (S, NP, VP, ADVP, AP, PP and some others). If there is no parse covering the complete input, the parser finds all parses for each substring. In such cases, the robustness component will then select the best sequence of non-overlapping parses (i.e., maximal projections) from this set.

In order to select the best parse from the compact parse forest, a best-first search algorithm is applied. The algorithm consults a Maximum Entropy disambiguation model to judge the quality of (partial) parses. Since the disambiguation model includes inherently non-local features, efficient dynamic programming solutions are not directly applicable. Instead, a best-first beam-search algorithm is employed (van Noord and Malouf, 2005; van Noord, 2006).

2.3 Maximum Entropy disambiguation model

The maximum entropy model is a conditional model which assigns a probability to a parse t for a given sentence s. Furthermore, $f_i(t)$ are the feature functions which count the occurrence of each feature i in a parse t. Each feature i has an associated weight λ_i. The score ϕ of a parse t is defined as the sum of the weighted feature counts:

$$\phi(t) = \sum_i \lambda_i f_i(t)$$

If t is a parse of s, the actual conditional probability is given by the following, where $T(s)$ are all parses of s:

$$P(t|s) = \frac{\exp(\phi(t))}{\sum_{u \in T(s)} \exp(\phi(u))}$$

However, note that if we only want to select the best parse we can ignore the actual probability, and it suffices to use the score ϕ to rank competing parses.

The Maximum Entropy model employs a large set of features. The standard model uses about 42,000 different features. Features describe various properties of parses. For instance, the model includes features which signal the application of particular grammar rules, as well as local configurations of grammar rules. There are features signalling specific POS-tags and subcategorization frames. Other features signal local or non-local occurrences of extraction (WH-movement, relative clauses etc.), the grammatical role of the extracted element (subject vs. non-subject etc.), features to represent the distance of a relative clause and the noun it modifies, features describing the amount of parallelism between conjuncts in a coordination, etc. In addition, there are lexical features which represent the co-occurrence of two specific words in a specific dependency, and the occurrence of a specific word as a specific dependent for a given POS-tag. Each parse is characterized by its feature vector (the counts for each of the 42,000 features). Once the model is trained, each feature is associated with its weight λ (a positive or negative number, typically close to 0). To find out which parse is the best parse according to the model, it suffices to multiply the frequency of each feature with its corresponding weight, and sum these weighted frequencies. The parse with the highest sum is the best parse. Formal details of the disambiguation model are presented in van Noord and Malouf (2005).

2.4 Dependency structures

Although Alpino is not a dependency grammar in the traditional sense, dependency structures are generated by the lexicon and grammar rules as the value of a dedicated feature `dt`. The dependency structures are based on CGN (Corpus Gesproken Nederlands, Corpus of Spoken Dutch) (Hoekstra et al., 2003), D-Coi and LASSY (van Noord et al., 2006). Such dependency structures are somewhat idiosyncratic, as can be observed in the example in figure 1 for the sentence:

(3) waar en wanneer dronk Elvis wijn?
where and when did Elvis drink wine?

2.5 Evaluation

The output of the parser is evaluated by comparing the generated dependency structure for a corpus sentence to the gold standard dependency structure in a treebank. For this comparison, we represent the dependency structure (a directed acyclic graph) as a set of named dependency relations. The dependency graph in figure 1 is represented with the following set of dependencies:

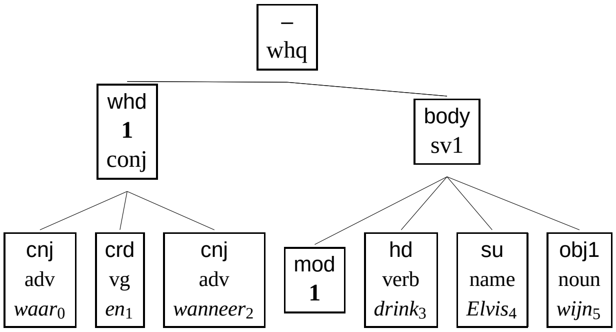

Figure 1: Dependency graph example. Reentrant nodes are visualized using a bold-face index. Root forms of head words are explicitly included in separate nodes, and different types of head receive a different relation label such as hd, crd (for coordination), whd (for WH-phrases) etc. In this case, the WH-phrase is both the *whd* element of the top-node, as well as a *mod* dependent of *drink*.

crd/cnj(en, waar)	crd/cnj(en, wanneer)
whd/body(en, drink)	hd/mod(drink, en)
hd/obj1(drink, wijn)	hd/su(drink, Elvis)

Comparing these sets, we count the number of dependencies that are identical in the generated parse and the stored structure, which is expressed traditionally using f-score (Briscoe et al., 2002). We prefer to express similarity between dependency structures by *concept accuracy*:

$$\mathrm{CA} = 1 - \frac{\sum_i D_f^i}{\max(\sum_i D_g^i, \sum_i D_p^i)}$$

where D_p^i is the number of dependencies produced by the parser for sentence i, D_g is the number of dependencies in the treebank parse, and D_f is the number of incorrect and missing dependencies produced by the parser.

The standard version of Alpino that we use here as baseline system is trained on the 145,000 word Alpino treebank, which contains dependency structures for the cdbl (newspaper) part of the Eindhoven corpus. The parameters for training the model are the same for the baseline model, as well as the model that includes the self-trained bilexical preferences (introduced below). These parameters include the Gaussian penalty, thresholds for feature selection, etc. Details of the training procedure are described in van Noord and Malouf (2005).

#sentences	100%	30,000,000
#words		500,000,000
#sentences without parse	0.2%	100,000
#sentences with fragments	8%	2,500,000
#single full parse	92%	27,500,000

Table 1: Approximate counts of the number of sentences and words in the parsed corpus. About 0,2% of the sentences did not get a parse, for computational reasons (out of memory, or maximum parse time exceeded).

2.6 Parsed Corpora

Over the course of about a year, Alpino has been used to parse most of the TwNC-02 (Twente Newspaper Corpus), Dutch Wikipedia, and the Duch part of Europarl. TwNC consists of Dutch newspaper texts from 1994 - 2004. We did not use the material from Trouw 2001, since part of that material is used in the test set used below. We used the 200 node Beowulf Linux cluster of the High-Performance Computing center of the University of Groningen. The dependency structures are stored in XML. The XML files can be processed and searched in various ways, for instance, using XPATH, XSLT and Xquery (Bouma and Kloosterman, 2002). Some quantitative information of this parsed corpus is listed in table 1. In the experiments described below, we do not distinguish between full and fragment parses; sentences without a parse are obviously ignored.

3 Bilexical preferences

3.1 Association Score

The parsed corpora described in the previous section have been used in order to compute association scores between lexical dependencies. The parses constructed by Alpino are dependency structures. In such dependency structures, the basic dependencies are of the form $r(w_1, w_2)$ where r is a relation such as *subject, object, modifier, prepositional complement, . . .*, and w_i are root forms of words.

Bilexical preference between two root forms w_1

tokens	480,000,000
types	100,000,000
types with frequency ≥ 20	2,000,000

Table 2: Number of lexical dependencies in parsed corpora (approximate counts)

bijltje	gooi_neer	13
duimschroef	draai_aan	13
peentje	zweet	13
traantje	pink_weg	13
boontje	dop	12
centje	verdien_bij	12
champagne_fles	ontkurk	12
dorst	les	12

Table 3: Pairs involving a direct object relationship with the highest pointwise mutual information score.

and w_2 is computed using an association score based on *pointwise mutual information*, as defined by Fano (1961) and used for a similar purpose in Church and Hanks (1990), as well as in many other studies in corpus linguistics. The association score is defined here as follows:

$$I(r(w_1, w_2) = \log \frac{f(r(w_1, w_2))}{f(r(w_1, \_))f(\_(\_, w_2))}$$

where $f(X)$ is the relative frequency of X. In the above formula, the underscore is a place holder for an arbitrary relation or an arbitrary word. The association score I compares the actual relative frequency of w_1 and w_2 with dependency r, with the relative frequency we would expect if the words were independent. For instance, to compute I(hd/obj1(`drink,melk`)) we lookup the number of times `drink` occurs with a direct object out of all 462,250,644 dependencies (15,713) and the number of times `melk` occurs as a dependent (10,172). If we multiply the two corresponding relative frequencies, we get the expected relative frequency (0.35) for hd/obj1(`drink,melk`), which is about 560 times as big as the actual frequence, 195. Taking the log of this gives us the association score (6.33) for this bi-lexical dependency. Note that pairs that we have seen fewer than 20 times are ignored. Mutual information scores are unreliable for low frequencies. An additional benefit of a frequency threshold is a manageable size of the resulting data-structures.

The pairs involving a direct object relationship with the highest scores are listed in table 3. The highest scoring nouns that occur as the direct object of `drink` are listed in table 4.

biertje	small glass of beer	8
borreltje	strong alcoholic drink	8
glaasje	small glass	8
pilsje	small glass of beer	8
pintje	small glass of beer	8
pint	glass of beer	8
wijntje	small glass of wine	8
alcohol	alcohol	7
bier	beer	7

Table 4: Pairs involving a direct object relationship with the highest pointwise mutual information score for the verb `drink`.

overlangs	snijd_door	12
welig	tier	12
dunnetjes	doe_over	11
stief_moederlijk	bedeel	11
on_zedelijk	betast	11
stierlijk	verveel	11
cum laude	studeer_af	10
hermetisch	grendel_af	10
ingespannen	tuur	10
instemmend	knik	10
kostelijk	amuseer	10

Table 5: Pairs involving a modifier relationship between a verb and an adverbial with the highest association score.

Selection restrictions are often associated only with direct objects. We include bilexical association scores for all types of dependencies. We found that association scores for other types of dependencies also captures both collocational preferences as well as weaker cooccurrence preferences. Some examples including modifiers are listed in table 5. Such preferences are useful for disambiguation as well. Consider the ambiguous Dutch sentence

(4) omdat we lauw bier dronken
because we drank warm beer
because we drank beer warmly

The adjective `lauw` (cold, lukewarm, warm) can be used to modify both nouns and verbs; this latter possibility is exemplified in:

(5) We hebben lauw gereageerd
We reacted indifferently

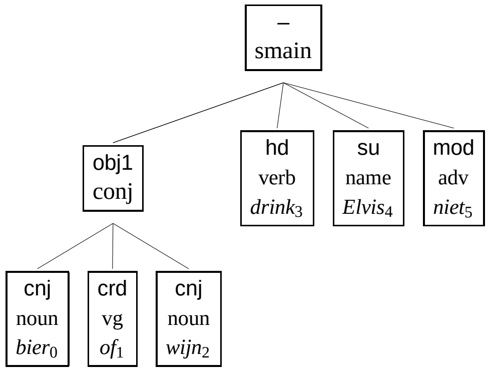

Figure 2: Dependency structure produced for coordination

3.2 Extending pairs

The CGN dependencies that we work with fail to relate pairs of words in certain syntactic constructions for which it can be reasonably assumed that bilexical preferences should be useful. We have identified two such constructions, namely relative clauses and coordination, and for these constructions we generalize our method, to take such dependencies into account too.

Consider coordinations such as:

(6) Bier of wijn drinkt Elvis niet
Beer or wine, Elvis does not drink

The dependency structure of the intended analysis is given in figure 2. The resulting set of dependencies for this example treats the coordinator as the head of the conjunction:

hd/obj1(`drink`, `of`) crd/cnj(`of`, `bier`)
crd/cnj(`of`, `wijn`) hd/su(`drink`, `elvis`)
hd/mod(`drink`, `niet`)

So there are no direct dependencies between the verb and the individual conjuncts. For this reason, we add additional dependencies $r(A, C)$ for every pair of dependency $r(A, B)$, crd/cnj(B, C).

Relative clauses are another syntactic phenomenon where we extend the set of dependencies. For a noun phrase such as:

(7) Wijn die Elvis niet dronk
Wine which Elvis did not drink

there is no direct dependency between `wijn` and `drink`, as can be seen in the dependency structure given in figure 3. Sets of dependencies are extended in such cases, to make the relation between the noun and the role it plays in the relative clause explicit.

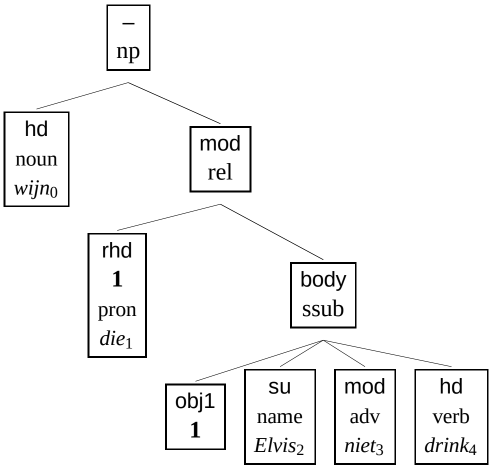

Figure 3: Dependency structure produced for relative clause

3.3 Using association scores as features

The association scores for all dependencies are used in our maximum entropy disambiguation model as follows. The technique is reminiscent of the inclusion of auxiliary distributions in stochastic attribute-value grammar (Johnson and Riezler, 2000).

Recall that a maximum entropy disambiguation model exploits features. Features are properties of parses, and we can use such features to describe any property of parses that we believe is of importance for disambiguation. For the disambiguation model, a parse is fully characterized by a vector of feature counts.

We introduce features $z(t, r)$ for each of the major POS labels t (verb, noun, adjective, adverb, ...) and each of the dependency relations r. The 'count' of such a feature is determined by the association scores for actually occuring dependency pairs. For example, if in a given parse a given verb v has a direct object dependent n, then we compute the association of this particular pair, and use the resulting number as the count of that feature. Of course, if there are multiple dependencies of this type in a single parse, the corresponding association scores are all summed.

To illustrate this technique, consider the dependency structure given earlier in figure 2. For this example, there are four of these new features with a non-zero count. The counts are given by the corresponding association scores as follows:

$$\begin{array}{lcl}
z(\text{verb}, \text{hd/su}) & = & I(\text{hd/su}(\texttt{drink}, \texttt{elvis})) \\
z(\text{verb}, \text{hd/mod}) & = & I(\text{hd/mod}(\texttt{drink}, \texttt{niet})) \\
z(\text{verb}, \text{hd/obj1}) & = & I(\text{hd/obj1}(\texttt{drink}, \texttt{of})) \\
 & + & I(\text{hd/obj1}(\texttt{drink}, \texttt{bier})) \\
 & + & I(\text{hd/obj1}(\texttt{drink}, \texttt{wijn})) \\
z(\text{conj}, \text{crd/cnj}) & = & I(\text{crd/cnj}(\texttt{of}, \texttt{bier})) \\
 & + & I(\text{crd/cnj}(\texttt{of}, \texttt{wijn}))
\end{array}$$

It is crucial to observe that the new features do not include any direct reference to actual words. This means that there will be only a fairly limited number of new features (depending on the number of tags t and relations r), and we can expect that these features are frequent enough to be able to estimate their weights in training material of limited size.

Association scores can be negative if two words in a lexical dependency occur less frequently than one would expect if the words were independent. However, since association scores are unreliable for low frequencies (including, often, frequencies of zero), and since such negative associations involve low frequencies by their nature, we only take into account positive association scores.

4 Experiments

We report on two experiments. In the first experiment, we report on the results of tenfold cross-validation on the Alpino treebank. This is the material that is standardly used for training and testing. For each of the sentences of this corpus, the system produces atmost the first 1000 parses. For every parse we compute the quality by comparing its dependency structure with the gold standard dependency structure in the treebank. For training, atmost 100 parses are selected randomly for each sentence. For (tenfold cross-validated) testing, we use all available parses for a given sentence. In order to test the quality of the model, we check for each given sentence which of its atmost 1000 parses is selected by the disambiguation model. The quality of that parse is used in the computation of the accuracy, as listed in table 6. The column labeled *exact* measures the proportion of sentences for which the model selected the best possible parse (there can be multiple best possible parses). The *baseline* row reports on the quality of a disambiguation model which simply selects the first parse for each sentence. The *oracle* row reports on the quality of the best-possible disambiguation model, which would (by magic) always select the best possible parse (some parses are outside the coverage of the system, and some parses are generated only after more than 1000 inferior parses). The *error reduction* column measures which part of the disambiguation problem (difference between the baseline and oracle scores) is solved by the model.[1]

	fscore	err.red.	exact	CA
	%	%	%	%
baseline	74.02	0.00	16.0	73.48
oracle	91.97	100.00	100.0	91.67
standard	87.41	74.60	52.0	87.02
+self-training	87.91	77.38	54.8	87.51

Table 6: Results with ten-fold cross-validation on the Eindhoven-cdbl part of the Alpino treebank. In these experiments, the models are used to select a parse from a given set of atmost 1000 parses per sentence.

The results show a small but clear increase in error reduction, if the standard model (without the association score features) is compared with a (retrained) model that includes the association score features. The relatively large improvement of the *exact* score suggests that the bilexical preference features are particularly good at choosing between very good parses.

For the second experiment, we evaluate how well the resulting model performs in the full system. First of all, this is the only really convincing evaluation which measures progress for the system as a whole by virtue of including bilexical preferences. The second motivation for this experiment is for methodological reasons: we now test on a truly unseen test-set. The first experiment can be criti-

[1] Note that the error reduction numbers presented in the table are lower than those presented in van Noord and Malouf (2005). The reason is, that we report here on experiments in which parses are generated with a version of Alpino with the POS-tagger switched on. The POS-tagger already reduces the number of ambiguities, and in particular solves many of the 'easy' cases. The resulting models, however, are more effective in practice (where the model also is applied after the POS-tagger).

	prec %	rec %	fscore %	CA %
standard	90.77	90.49	90.63	90.32
+self-training	91.19	90.89	91.01	90.73

Table 7: Results on the WR-P-P-H part of the D-Coi corpus (2267 sentences from the newspaper Trouw, from 2001). In these experiments, we report on the full system. In the full system, the disambiguation model is used to guide a best-first beam-search procedure which extracts a parse from the parse forest. Difference in CA was found to be significant (using paired T-test on the per sentence CA scores).

cized on methodological grounds as follows. The Alpino Treebank was used to train the disambiguation model which was used to construct the large parsed treebank from which we extracted the counts for the association scores. Those scores might somehow therefore indirectly reflect certain aspects of the Alpino Treebank training data. Testing on that data later (with the inclusion of the association scores) is therefore not sound.

For this second experiment we used the WR-P-P-H (newspaper) part of the D-Coi corpus. This part contains 2256 sentences from the newspaper Trouw (2001). In table 7 we show the resulting f-score and CA for a system with and without the inclusion of the $z(t, r)$ features. The improvement found in the previous experiment is confirmed.

5 Conclusion and Outlook

One might wonder why self-training works in the case of selection restrictions, at least in the set-up described above. One may argue that, in order to learn that *milk* is a good object for *drink*, the parser has to analyse examples of *drink milk* in the raw data correctly. But if the parser is capable of analysing these examples, why does it need selection restrictions? The answer appears to be that the parser (without selection restrictions) is able to analyse the large majority of cases correctly. These cases include the many easy occurrences where no (difficult) ambiguities arise (case marking, number agreement and other syntactic characteristics often force a single reading). The easy cases outnumber the misparsed difficult cases, and therefore the selection restrictions can be learned. Using these selection restrictions as additional features, the parser is then able to also get the difficult, ambiguous, cases right.

There are various aspects of our method that need further investigation. First of all, existing techniques that involve selection restrictions (e.g., Resnik (1993)) typically assume classes of nouns, rather than individual nouns. In future work we hope to generalize our method to take classes into account, where the aim is to learn class membership also on the basis of large parsed corpora.

Another aspect of the technique that needs further research involves the use of a threshold in establishing the association score, and perhaps related to this issue, the incorporation of negative association scores (for instance for cases where a large number of cooccurrences of a pair would be expected but where in fact none or very few were found).

There are also some more practical issues that perhaps had a negative impact on our results. First, the large parsed corpus was collected over a period of about a year, but during that period, the actual system was not stable. In particular, due to various improvements of the dictionary, the root form of words that was used by the system changed over time. Since we used root forms in the computation of the association scores, this could be harmful in some specific cases. A further practical issue concerns repeated sentences or even full paragraphs. This happens in typical newspaper material for instance in the case of short descriptions of movies that may be repeated weekly for as long as that movie is playing. Pairs of words that occur in such repeated sentences receive association scores that are much too high. The method should be adapted to take this into account, perhaps simply by removing duplicated sentences.

Clearly, the idea that selection restrictions ought to be useful for parsing is not new. However, as far as we know this is the first time that automatically acquired selection restrictions have been shown to improve parsing accuracy results.

Acknowledgements

This research was carried out in part in the context of the D-Coi and Lassy projects. The D-Coi and Lassy projects are carried out within the STEVIN programme which is funded by the Dutch and Flemish governments (http://taalunieversum.org/taal/technologie/stevin/).

References

Takeshi Abekawa and Manabu Okumura. 2006. Japanese dependency parsing using co-occurrence information and a combination of case elements. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 833–840, Sydney, Australia, July. Association for Computational Linguistics.

Gosse Bouma and Geert Kloosterman. 2002. Querying dependency treebanks in XML. In *Proceedings of the Third international conference on Language Resources and Evaluation (LREC)*, pages 1686–1691, Gran Canaria, Spain.

Ted Briscoe, John Carroll, Jonathan Graham, and Ann Copestake. 2002. Relational evaluation schemes. In *Proceedings of the Beyond PARSEVAL Workshop at the 3rd International Conference on Language Resources and Evaluation*, pages 4–8, Las Palmas, Gran Canaria.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information and lexicography. *Computational Linguistics*, 16(1):22–29.

Robert Mario Fano. 1961. *Transmission of Information: A Statistical Theory of Communications*. MIT Press, Cambridge, MA.

Heleen Hoekstra, Michael Moortgat, Bram Renmans, Machteld Schouppe, Ineke Schuurman, and Ton van der Wouden, 2003. *CGN Syntactische Annotatie*, December.

Mark Johnson and Stefan Riezler. 2000. Exploiting auxiliary distributions in stochastic unification-based grammars. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 154–161, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Daisuke Kawahara and Sadao Kurohashi. 2006. A fully-lexicalized probabilistic model for japanese syntactic and case structure analysis. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 176–183, Morristown, NJ, USA. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, New York City, USA, June. Association for Computational Linguistics.

Robbert Prins. 2005. *Finite-State Pre-Processing for Natural Language Analysis*. Ph.D. thesis, University of Groningen.

Philip Stuart Resnik. 1993. *Selection and information: a class-based approach to lexical relationships*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, USA.

Gertjan van Noord and Robert Malouf. 2005. Wide coverage parsing with stochastic attribute value grammars. Draft available from http://www.let.rug.nl/˜vannoord. A preliminary version of this paper was published in the Proceedings of the IJCNLP workshop Beyond Shallow Analyses, Hainan China, 2004.

Gertjan van Noord, Ineke Schuurman, and Vincent Vandeghinste. 2006. Syntactic annotation of large corpora in STEVIN. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, Genoa, Italy.

Gertjan van Noord. 2006. **A**t **L**ast **P**arsing **I**s **N**ow **O**perational. In *TALN 2006 Verbum Ex Machina, Actes De La 13e Conference sur Le Traitement Automatique des Langues naturelles*, pages 20–42, Leuven.

Examples

Here we list a number of examples, which suggest that selection restrictions can also be important for dependencies, other than direct objects.

High scoring pairs involving a subject relationship with a verb:

alarmbel	rinkel
champagnekurk	knal
gij	echtbreek
haan	kraai
kikker	kwaak
rups	verpop
vonk	overspring
zweet	parel
belletje	rinkel
brievenbus	klepper

High scoring pairs involving a modifier relationship with a noun:

in vitro	fertilisatie
Hubble	ruimtetelescoop
zelfrijzend	bakmeel
bezittelijk	voornaamwoord
ingegroeid	teennagel
knapperend	haardvuur
levendbarend	hagedis
onbevlekt	ontvangenis
ongeblust	kalk

High scoring pairs involving a predicative complement relationship with a verb:

beetgaar	kook
beuk	murw
schuimig	klop
suf	peins
suf	pieker
doormidden	scheur
ragfijn	hak
stuk	bijt
au serieux	neem
in duigen	val
lam	leg

High scoring pairs involving an apposition relationship with a noun:

jongensgroep	Boyzone
communicatiesysteem	C2000
blindeninstituut	De Steffenberg
haptonoom	Ted Troost
gebedsgenezeres	Greet Hofmans
rally	Parijs-Dakar
tovenaar	Gandalf
aartsengel	Gabriel
keeperstrainer	Joep Hiele
basketbalcoach	Ton Boot
partizaan	Tito

High scoring pairs involving a measure phrase relationship with an adjective:

graadje	erger
lichtjaar	verwijderd
mijlenver	verwijderd
niets	liever
eindje	verderop
graad	warmer
illusie	armer
kilogram	wegend
onsje	minder
maatje	te groot
knip	waard