Adaptability of Lexical Acquisition for Large-scale Grammars

Kostadin Cholakov[†], **Gertjan van Noord**[†], **Valia Kordoni**[‡], **Yi Zhang**[‡]

† University of Groningen, The Netherlands
‡ Saarland University and DFKI GmbH, Germany
{k.cholakov,g.j.m.van.noord}@rug.nl
{kordoni,yzhang}@coli.uni-saarland.de

Abstract

In this paper, we demonstrate the portability of the lexical acquisition (LA) method proposed in Cholakov and van Noord (2010a). Here, LA refers to the acquisition of linguistic descriptions for words which are not listed in the lexicon of a given computational grammar, i.e., words which are unknown to this grammar. The method we discuss was originally developed for the Dutch Alpino system, and the paper shows that the method also applies to the GG (Crysmann, 2003), a computational HPSG grammar of German. The LA method obtains very similar results for German (84% F-measure on learning unknown words). Extending the GG with the lexical entries proposed by the LA method causes an important improvement in parsing accuracy for a test set of sentences containing unknown words. Furthermore, in a smaller experiment, we show that the linguistic knowledge the LA method provides can also be used for sentence generation.

1 Introduction

Computational grammars of natural language lie at the heart of various wide-coverage symbolic parsing systems. At present, such systems have been integrated into real-world NLP applications, such as IE, QA, grammar checking, MT and intelligent IR. This integration, though, has reminded us of some of the problems which the aforementioned grammars encounter when applied to naturally occurring text, in particular lack of lexical coverage. Since such grammars usually rely on hand-crafted lexicons containing elaborate linguistic descriptions, words not listed in the lexicon, i.e. words unknown to the grammar, pose a major issue in the employment of the grammars for real-life applications. In this context, *lexical acquisition* refers to the acquisition of correct lexical descriptions for unknown words.

Various LA techniques for computational grammars have been proposed in the past. Cussens and Pulman (2000) used a symbolic approach employing *inductive logic programming*, while Erbach (1990), Barg and Walther (1998) and Fouvry (2003) followed a unification-based approach. Other approaches have treated LA as a classification task where the unknown word is mapped to a finite set of labels. Baldwin (2005) has extracted features from various linguistic resources (POS taggers, chunkers, etc.) and used a set of binary classifiers to learn lexical entries for a large-scale grammar of English (ERG; (Copestake and Flickinger, 2000)). Zhang and Kordoni (2006) and Cholakov et al. (2008), on the other hand, have trained a maximum entropy (ME) classifier with features extracted from the grammar in order to acquire new lexical entries for the ERG and the GG (Crysmann, 2003), respectively. Extending this line of research, Cholakov and van Noord (2010a) have proposed a technique for learning unknown words for the Dutch Alpino grammar (van Noord, 2006) which takes into account the morphology of the unknown word and various contexts which it occurs in. In each case, however, LA is performed within a single parsing system, in a single framework, and mostly for a single language. It is unclear to what extent the various techniques can be used for a different language or parsing architecture.

The main motivation for the current work is

Proceedings of Recent Advances in Natural Language Processing, pages 355–362,
Hissar, Bulgaria, 12-14 September 2011.

to explore the challenging task of employing one such LA technique, the one proposed in Cholakov and van Noord (2010a) – henceforth C&vN– for another system and another language. The C&vN technique is an obvious candidate for such a generalisation challenge, since Cholakov and van Noord (2010a) claim explicitly that the method should apply to other systems and languages provided some conditions are met. The conditions listed in Cholakov and van Noord (2010a) are: a finite set of labels which unknown words are mapped onto, a syntactic parser, and a morphological component which generates the paradigm(s) of a given unknown word. As a further motivation for our choice we note that the method of C&vN can be extended to deal with wrong and incomplete lexical descriptions of words which are already in the lexicon (Cholakov and van Noord, 2010b). However, this extension is beyond the scope of the current paper.

The choice of German and the GG (Crysmann, 2003) as the target for our case study lies in the fact that German is a language with somewhat richer morphology than Dutch, which affects the design of the grammar and makes LA more challenging. A further challenge is posed by the fact that the GG, unlike Alpino, does not have a full form lexicon. Instead, lexical entries define only the stem of the word and all other forms are derived by applying various morphological rules defined in the grammar. In the case of the GG, the LA method has the additional task of mapping unknown words to their stems and, at the same time, the descriptions it acquires should be detailed enough to allow for the proper application of the morphological rules.

Naturally, one could employ other techniques for LA with the GG but our purpose is to show that we can avoid implementing system specific solutions by adapting an existing LA method.

The remainder of the paper is organised as follows. Section 2 describes the adoption of the discussed LA method to the GG. Section 3 presents the experiments conducted with the grammar and evaluates the performance of the LA algorithm. Section 4 investigates how the LA method affects parsing accuracy on sentences containing unknown words and explores the possibility of using newly acquired lexical entries in a small sentence realisation task. Section 5 concludes the paper.

2 Lexical Acquisition for German

In this section, we explain the main steps in the method presented in C&vN and we focus on issues which arise from porting it to the GG.

2.1 The Parsing Setup

The GG is a stochastic attribute-value grammar based on typed feature structures. The GG types are strictly defined within a type hierarchy. The grammar contains constructional and lexical rules, as well as a lexicon where words are assigned lexical types. Currently, it consists of 5K types, 115 rules and the lexicon contains approximately 55K entries. There are 411 distinct lexical types which words can be mapped onto.

We employ the PET system (Callmeier, 2000) to parse with the GG. PET is a system for efficient processing of unification-based grammars. It is an industrial strength implementation of a typed-feature structure formalism (Carpenter, 1992). The system comprises a sophisticated preprocessor, a bottom-up chart parser and a grammar compiler.

2.2 Constructing a Set of Labels for Learning

In C&vN the unknown words are mapped onto a finite set of labels, namely the linguistic descriptions contained in the Alpino lexicon. In the case of the GG, the unknown words have to be mapped onto lexical type(s) from the GG lexicon. We consider only open-class lexical types: nouns, adjectives, verbs and adverbs. In the case of Alpino, C&vN do not consider adverbs because adjectives which are used adverbially are listed as adjectives in the lexicon. The remaining adverbs are a closed class. In the GG, such adjectives are listed as adverbs and therefore the adverbs are also a target for lexical acquisition.

A further difference with Alpino is that the definitions of the lexical types in the GG are not explicit enough for the purposes of LA. Consider the lexical entry for *Abfahrten* (departures):

```
abfahrt-n := count-noun-le &
 [ MORPH.LIST.FIRST.STEM < "Abfahrt" >,
   SYNSEM.LKEYS [ --SUBJOPT -,
                  KEYAGR c-n-f,
                  KEYREL "_abfahrt_n_rel",
                  KEYSORT temp_move_poly,
                  MCLASS nclass-9 ] ].
```

The lexical type 'count-noun-le' shows that the word is a countable noun[1]. The KEYAGR feature

[1] *le* stands for lexeme.

indicates case, number and gender. In the example above, case and number are left underspecified while the gender is set to feminine. The value of SUBJOPT shows that this noun is always used with an article and MCLASS indicates its morphological paradigm. The KEYREL and KEYSORT features define the semantics of the word.

When performing LA with the GG, we need to learn not only the lexical type but also the information encoded in the various type features. For this purpose, we include the values of features which we consider relevant for LA into the type definitions. In the case of *Abfahrten* we include the value of the gender from the KEYAGR feature turning the lexical type into *count-noun-le_f*. Only features designating morphosyntactic agreement are considered. For all noun types and predicative adjectives this is the KEYAGR feature. For verb types allowing for prepositional complements, we consider the COMPAGR and the OCOMPAGR features which indicate the case of the the (oblique) complement. By creating such *expanded* lexical types, we give the LA method access to the information contained in the selected features.

The remaining features do not contribute to LA and they are also likely to cause data sparseness. When adding words to the lexicon, some of those features can safely be left underspecified while others (e.g., KEYREL) can be assigned default values. Experiments have shown that such mildly less constrained lexical entries do not affect the parsing accuracy since the ambiguity they create usually dissolves in the context of the unknown word.

2.3 Paradigm Generation and Its Importance

C&VN use the paradigm of the unknown word as an important source of morphological features for the classification process. However, as stated above, unlike Alpino, the GG does not have a full form lexicon. We see in the lexical entry of *Abfahrten* that the STEM feature defines only the stem of the word. All other morphological forms are derived by applying various morphological rules defined in the GG to the word stem. For this reason, we employ the paradigm not only as a source of features for the classifier but also as a way to map the unknown word to its stem.

The stem for nouns is the singular nominative noun form, for adjectives it is the base noninflected form and for verbs it is the root form. Adverbs in German have a single form which is used as the value of the STEM feature in adverb entries. Some nouns (e.g., *Baukosten* (building costs)) do not have all forms typical for German nouns. In such cases, the word itself is set as the value of the STEM feature.

Due to the GG design, it is not straightforward to use the morphological rules of the grammar for paradigm generation. Following a technique developed for generating the paradigms of Dutch words (Cholakov and van Noord, 2009), we created a German finite state morphology. The morphology does not have access to any linguistic information and thus, it generates all possible paradigms allowed by the word orthography. Then, the number of search hits Yahoo returns for each form in a given paradigm is combined with some simple heuristics to disambiguate the output of the morphology and to determine the correct paradigm(s). For words predicted to be nouns, we also apply heuristics to guess the gender.

One could argue that there is a simpler approach for mapping the various forms of the unknown word to its stem. For instance, the TreeTagger POS tagger (Schmid, 1994) could provide both POS and stem information with high accuracy. However, the generation of the paradigms allows us to extract contexts in which other forms of a given unknown word occur and thus, we have access to much more and linguistically diverse data. For example, C&VN show the benefits of having access to other forms of a word predicted to be a verb for learning subcategorization frames.

2.4 Classifier and Features

We employ the maximum entropy based classifier[2] and the features used for unknown word prediction as described in C&VN. The probability of a lexical type *t*, given an unknown word and its context *c* is:

$$(1) \qquad p(t|c) = \frac{exp(\sum_i \Theta_i f_i(t,c))}{\sum_{t' \in T} exp(\sum_i \Theta_i f_i(t',c))}$$

where $f_i(t, c)$ may encode arbitrary characteristics of the context and $< \Theta_1, \Theta_2, ... >$ can be evaluated by maximising the pseudo-likelihood on a training corpus (Malouf, 2002).

Table 1 shows the features for *Abfahrten*. Row **(i)** contains 4 separate features derived from the prefix of the word and 4 other suffix features are

[2]TADM; `http://tadm.sourceforge.net/`

given in row **(ii)**. The two features in rows **(iii)** and **(iv)** indicate whether the word starts with a separable particle and if it contains a hyphen, respectively. Since it is the stem of the unknown word we add to the lexicon, we also experimented with prefix and suffix features extracted from the stem. We assumed that those could allow for a better generalization of morphological properties but they proved to be less informative for the classifier.

Further, the paradigm generation method outputs a single paradigm for *Abfahrten* indicating that this word is a singular feminine noun. This information is explicitly used as a feature in the classifier which is shown in row **(v)** of Table 1.

Features
i) A, Ab, Abf, Abfa
ii) n, en, ten, rten
iii) particle_yes #in this case *Ab*
iv) hyphen_no
v) noun_feminine
vi) count-noun-le_f, mass-noun-le_f
vii) noun⟨f⟩

Table 1: Features for *Abfahrten*

Rows **(vi)** and **(vii)** show syntactic features obtained from what C&VN refer to as 'parsing with universal types'. Each unknown word is assigned the target types belonging to the POS of the paradigm(s) generated for this word. For example, *Abfahrten* is assigned all noun types from the set of types we want to learn. Sentences containing the unknown word and other of its forms are parsed with PET in best-only mode. For each sentence only the best parse selected by the disambiguation model of the parser is preserved. Then, the lexical type that has been assigned to the form of *Abfahrten* occurring in this parse is stored.

We employ the most frequently used type(s) (based on an empirical threshold) as features in the classifier (row **vi**). Further, as illustrated in row **(vii)**, each feature value we have attached to the type definition of the considered types (the part after the underscore) is also taken as a separate feature.

3 Experiments with Development Data

3.1 Experiment Setup

In our experiments with the GG, an open-class lexical type is considered if it has at least 10 lexical entries in the lexicon mapped onto it and it is assigned to at least 15 distinct words occurring in large corpora parsed with PET and the GG. The parsed corpus we use consists of roughly 2.5M sentences randomly selected from the German part of the Wacky project (Kilgarriff and Grefenstette, 2003). The Wacky project aims at the creation of large corpora for different languages, including German, from various web sources, such as online newspapers and magazines, legal texts, internet fora, etc.

Following these criteria, we have selected 39 open-class types out of the 411 lexical types defined in the GG. As described in Section 2.2, we re-defined the type definitions of the 39 types which resulted in the creation of 68 *expanded* types. This number is smaller than the 611 types used in the experiments with Alpino because the GG does not have a full form lexicon. Table 2 gives more details about the type distribution.

	Original types	Expanded types
Total	39	68
-nouns	5	15
-verbs	28	45
-adjectives	4	6
-adverbs	2	2

Table 2: Distribution of the target lexical types

In order to train and test the classifier, 2400 less frequent words are temporarily removed from the lexicon of the GG. Of these, 2000 are used for training, and 400 words are used for testing. We assume that less frequent words are typically unknown and, in order to simulate their behaviour, all 2400 words we removed from the lexicon have between 40 and 100 occurrences in the parsed corpus. Experiments with a minimum lower than 40 occurrences have shown that this is a reasonable threshold to filter out typos, tokenization errors, etc. The distribution of the parts-of-speech for the 2400 words is listed in Table 3 (some words have more than a single part-of-speech).

3.2 Evaluation of the Paradigm Generation Component

Since paradigms play such a crucial role in the experiments with the GG, we first evaluate the performance of the paradigm generation component.

Table 3 shows the overall results and the results for each POS. *Accuracy* indicates how many

of the generated paradigms are correct. In the

	overall	nouns	adj	verbs
total	2954	1196	651	694
accuracy(%)	96.45	91.09	100	99.54

Table 3: Paradigm generation results

paradigms generated for verbs there were three mistakes. However, the generated verb stems were all correct. Similarly, the stems for all nouns were correct, including the stems of 98 nouns which contained a mistake in their paradigm. In 91 cases the singular genitive form was incorrect, in another 12 cases the predicted gender was wrong. The mapping of the words to their correct stems is correct in all cases.

3.3 Evaluation of the Classifier

Let us now investigate the performance of the classifier. We allow prediction of multiple types per word but we discard the types accounting together for less than 5% of probability mass. Additionally, there are three baseline methods:

- *Naive*– each unknown word is assigned the most frequent expanded type in the lexicon: *count-noun-le_f*
- *Naive POS*– the word is given the most frequent expanded type for the POS of each paradigm generated for it
- *GG*– the unknown word is assigned the most frequently used type in the parsing stage (e.g., for *Abfahrten*, this is *count-noun-le_f* from row vi) in Table 1)

The overall results are given in Table 4 together with the result C&VN reported for Alpino. Table 5 breaks down the results for each POS. Precision indicates how many types found by the method are correct and recall indicates how many of the lexical types of a given word are actually found. The presented results are the average precision and recall for the 400 test words. The original lexical types which the words had before they were removed from the GG lexicon are used as a gold standard for comparison.

The LA model improves upon the baselines, and performs very similar to the results reported for Dutch. The German model achieves somewhat better recall which is balanced by lower precision. Figure 1 shows that the F-measure reaches 70% already at 100 training words. It goes up to 80% when 300 words are used for training the curve flattens out at 1600 training words. The results indicate that the method of C&VN can be successfully applied outside the environment which it was primarily developed for.

Model	Prec(%)	Rec(%)	F-meas(%)
Naive	21.75	21.07	21.41
Naive POS	58.96	47.65	52.7
GG	67	48.96	56.58
LA with the GG	82.04	86.5	84.21
LA with Alpino	89.08	80.52	84.58

Table 4: Overall experiment results

POS	Prec(%)	Rec(%)	F-meas(%)
Nouns	91	93.85	92.4
Adj	88.89	93.07	90.93
Verbs	65.02	69.64	67.25
Adverbs	75.32	76.32	75.82

Table 5: Detailed results for the LA model

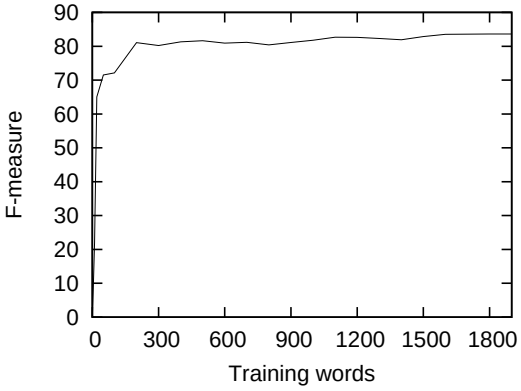

Figure 1: Learning curve

Predicting lexical entries for verbs is the hardest task for the LA model. The classifier has a strong bias towards assigning transitive and intransitive verb types. It either fails to predict infrequent frames or it wrongly predicts a transitive type for intransitive verbs and vice versa. Another difficulty for the model is the distinction which the GG makes between ergative and non-ergative verbs.

The main issue with adverbs is that many of them can be used as adjectives as well. As a consequence, the classifier has a strong bias towards predicting an adverb type for words for which an adjective type has also been predicted. Further, it also has a bias towards assigning one of the two adverb types, namely, *intersect-adv-le*. Finally, no

pattern in the errors for nouns and adjectives can be identified.

4 Tests with Real Unknown Words

4.1 LA and Parsing Accuracy

Once we have a trained model, we want to investigate how LA affects parsing accuracy.

We conducted an experiment with a test set of 450 sentences which all contain unknown words. The sentences are randomly selected from a German newspaper corpus containing 614K sentences. The articles in the corpus deal with various domains. For this experiment, we parse the 450 sentences with PET, under two conditions. In the first case, the standard lexicon of the GG is used, whereas in the second case, we add to the GG lexical entries acquired offline by the LA method. The standard GG model includes a guesser which assigns generic types to the unknown words. Some of the morphosyntactic features in these types are left underspecified and the semantic features receive default values. The experiment therefore compares the difference in parsing accuracy of the built-in guesser with the LA model.

From the 450 sentences, we selected the 113 sentences which PET/GG was able to parse with the standard lexicon as well as with the extended lexicon (for this reason, the accuracy figures below are relatively high). For 100 out of the 113 sentences a correct parse is produced (among the set of parses) by at least one of the methods. In the standard setup, a correct parse can be produced for 89 sentences. For the setup with LA, this number increases to 99 sentences. The correct parses for the 100 sentences were used as our gold standard, to be able to report the accuracy numbers below, for the best parse. These 100 sentences have an average sentence length of 17.72 words, and contain 106 distinct unknown words. Accuracy is measured in terms of labelled brackets. The results are listed in Table 6.

Model	Accuracy	msec/sentence
GG-standard	92.80	9824
GG + LA	94.51	9911

Table 6: Results with real unknown words

Adding the lexical entries proposed by the LA model leads to an increase in parsing accuracy. This result is consistent with the one reported for C&VN for Dutch.

The increase in parsing accuracy has to do mainly with the fact that the built-in guesser assigns noun types to the vast majority of the unknown words. Many of the features in those entries are left underspecified which creates a lot of ambiguity and which makes it harder for the parser disambiguation model to select the correct analysis. As mentioned in Section 2.2, the LA model also leaves some of the features underspecified or assigns default values to them. Still, the information it provides is much more linguistically accurate which helps for ambiguity resolution and the production of the correct parse.

4.2 LA for Sentence Realisation

As a further evaluation, extending the evaluation methodology of C&VN, we also investigate if the acquired lexical entries affect sentence realisation.

The GG adopts Minimal Recursion Semantics (MRS, Copestake et al. (2005)) as semantic representation. This, together with the fine-grained linguistic information in the GG lexical types, allows for finding the textual realisations for a given input semantic representation. Sentence realisation with the GG is performed within the LKB grammar engineering platform which provides an efficient generation engine. This engine is essentially a chart-based generator (Kay, 1996) with various optimisations for MRS and packed parse forest (Carroll and Oepen, 2005).

As there are less ordering constraints in the semantic representation (comparing to the word sequence in parsing inputs), the computation is intrinsically more expensive. While in parsing the ambiguity in the less constrained lexical entries acquired with LA dissolves quickly in its context, there is a potential risk of overgeneration in sentence realisation.

We conduct an indicative experiment with 14 unknown words from the test set used in Section 4.1. These words have been assigned verb types by the classifier. The focus of the experiment is on verbs because of the large number of possible subcategorization frames, which is a major source for overgeneration and can severely damage the quality of the sentence realisations.

We have extracted a test set of 64 sentences from the Wacky web corpus we used in Section 3.1, each of which contains one of the 14 selected words. We parse those sentences with the GG using the verb lexical entries acquired for the 14

unknown words with LA. Some of the sentences are edited to make sure that there are no other unknown words in them. The best MRS is recorded, and sent back to the generation engine. The generated realisations are recorded and compared with the original input sentence. The average sentence length of the selected 64 sentences is 7.66 tokens.

We construct manually another sentence set where the 14 unknown words are replaced by verbs from the GG lexicon. Each replacement verb belongs to the same lexical type and has the same type features as the lexical entry acquired for the unknown word it replaces. This comparison set indicates what the performance of the GG would be with fully constrained, but otherwise similar lexical entries.

There were 3.28 realisations per sentence for the test set versus 3.16 for the comparison one. As for accuracy, a realisation is considered correct if it is an exact match of the original sentence (excluding punctuation). Despite the higher number for realisations per sentence for the test set, the quality of the realisations is the same for both sets– for 60 sentences a correct realisation is produced. Thus, the entries acquired with LA can be employed for both parsing and realisation.

5 Conclusion

We addressed the challenging issue of generalising LA techniques for computational grammars by applying the method of C&VN, originally developed for the Dutch Alpino grammar, to the GG, an HPSG grammar for German. This resulted in improved parsing accuracy. The modifications we made to adopt the method to the linguistic properties of German and the design of the GG did not change its fundamental principles and the basic steps of the algorithm it implements.

Moreover, we have also shown that the lexicon acquired with this method may also be used for generation, something that to our knowledge has not been tried so far in similar linguistic processing architectures. The successful adaptation of the discussed LA method for the GG also suggests that such architectures share common design principles which makes it possible for common solutions to be developed.

References

Tim Baldwin. 2005. Bootstrapping deep lexical resources: Resources for courses. In *Proceedings of the ACL-SIGLEX 2005 Workshop on Deep Lexical Acquisition*, Ann Arbor, USA.

Petra Barg and Markus Walther. 1998. Processing unknown words in HPSG. In *Proceedings of the 36th Conference of the ACL*, Montreal, Quebec, Canada.

Ulrich Callmeier. 2000. PET– a platform for experimentation with efficient HPSG processing techniques. In *Journal of Natural Language Engineering*, volume 6, pages 99–107. Cambridge University Press.

Bob Carpenter. 1992. *The Logic of Typed Feature Structures*. Cambridge University Press.

John Carroll and Stephan Oepen. 2005. High efficiency realization for a wide-coverage unification grammar. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP 2005)*, pages 165–176, Jeju Island, Korea.

Kostadin Cholakov and Gertjan van Noord. 2009. Combining finite state and corpus-based techniques for unknown word prediction. In *Proceedings of the 7th Recent Advances in Natural Language Processing (RANLP) conference*, Borovets, Bulgaria.

Kostadin Cholakov and Gertjan van Noord. 2010a. Acquisition of unknown word paradigms for large-scale grammars. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING-2010)*, Beijing, China.

Kostadin Cholakov and Gertjan van Noord. 2010b. Using unknown word techniques to learn known words. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, Cambridge, MA.

Kostadin Cholakov, Valia Kordoni, and Yi Zhang. 2008. Towards domain-independent deep linguistic processing: Ensuring portability and re-usability of lexicalised grammars. In *Proceedings of COLING 2008 Workshop on Grammar Engineering Across Frameworks (GEAF08)*, Manchester, UK.

Ann Copestake and Dan Flickinger. 2000. An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the 2nd International Conference on Language Resource and Evaluation (LREC 2000)*, Athens, Greece.

Ann Copestake, Dan Flickinger, Carl J. Pollard, and Ivan A. Sag. 2005. Minimal recursion semantics: an introduction. *Research on Language and Computation*, 3(4):281–332.

Berthold Crysmann. 2003. On the efficient implementation of German verb placement in HPSG. In *Proceedings of RANLP 2003*, Borovets, Bulgaria.

James Cussens and Stephen Pulman. 2000. Incorporating linguistic constraints into inductive logic programming. In *Proceedings of the Fourth Conference on Computational Natural Language Learning*.

Gregor Erbach. 1990. Syntactic processing of unknown words. IWBS report 131. Technical report, IBM, Stuttgart.

Frederik Fouvry. 2003. Lexicon acquisition with a large-coverage unification-based grammar. In *Companion to the 10th Conference of EACL*, pages 87–90, Budapest, Hungary.

Martin Kay. 1996. Chart generation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 200–204, Santa Cruz, California, USA, June. Association for Computational Linguistics.

Adam Kilgarriff and G Grefenstette. 2003. Introduction to the special issue on the web as a corpus. In *Computational Linguistics*, volume 29, pages 333–347.

Robert Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th conference on Natural Language Learning (CoNLL-2002)*, pages 49–55, Taipei, Taiwan.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK.

Gertjan van Noord. 2006. At last parsing is now operational. In *Proceedings of TALN*, Leuven, Belgium.

Yi Zhang and Valia Kordoni. 2006. Automated deep lexical acquisition for robust open text processing. In *Proceedings of the Fifth International Conference on Language Resourses and Evaluation (LREC 2006)*, Genoa, Italy.