Lexicon acquisition with a large-coverage unification-based grammar

Frederik Fouvry
Computational Linguistics
Saarland University
PO Box 15 11 50
D-66041 Saarbrücken, Germany
fouvry@coli.uni-sb.de

Abstract

We describe how unknown lexical entries are processed in a unification-based framework with large-coverage grammars and how from their usage lexical entries are extracted. To keep the time and space usage during parsing within bounds, information from external sources like Part of Speech (PoS) taggers and morphological analysers is taken into account when information is constructed for unknown words.

1 Introduction

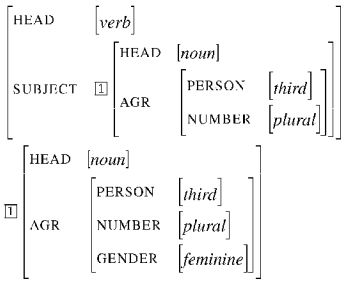
For Natural Language Processing (NLP) in general, and processing with linguistically rich frameworks more specifically, unknown words are a problem. The following gives an idea of the extent of the problem. In an evaluation of a large-scale grammar for unrestricted text on a newspaper corpus, we found that the number of failed parses due to unknown words accounted for around 89% of the total number of unsuccessful analyses. Even though this figure does not say anything about the grammar (these failures may be hiding many others), it shows the importance of the problem.

For unification-based implementations, which often refer to linguistic theories and are therefore rich in information, one approach to deal with unknown words has been proposed several times: to exploit the syntactic context of completed analyses to collect information about a new word. A few implementations have been developed to demonstrate the feasibility of the technique, but to our knowledge it has not been applied yet to large-coverage grammars. In this note we discuss how we are applying it to such a grammar for unrestricted text. Starting from this "standard" technique, we extend it and integrate PoS and morphological information, originating from external resources.

We will first describe the method of learning information from the syntactic context. Then we discuss the current results of our implementation, and how the external resources are put to use. Finally an evaluation scheme is presented and some issues we intend to investigate next.

2 Acquiring new information

In a unification-based framework, information is percolated throughout the parse tree via the reentrancies. Information that is underspecified in a lexical entry very often becomes more specific when it is used in a parse tree. Take the following example. The lexical entry for the French verb form *étaient* ("were") specifies that its subject should be a plural noun phrase (and in the third person). When it is combined with the feminine plural noun phrase *les conditions* ("the conditions") via $\boxed{1}$ in (1), the information about the subject of *étaient* will also include the gender value.

(1)


Normally this increase in information is not used for anything outside the current analysis. With unknown words however, this property can be used to find out how they can be used. When a word is encountered that cannot be found in the lexicon, a generic underspecified lexical entry is used, and for the rest parsing proceeds as usual. The result is one or more analyses where the information of the unknown entry will have been filled in as described above by the surrounding words. If instead of *les conditions* an unknown NP had been used, we would know from the specifications on *étaient* that it should be a plural noun. The feature structures thus specified are candidate lexical entries for the unknown words. This technique is described by e.g. Erbach (1990) and Walther and Barg (1998).

As pointed out by all of these authors, these feature structures will be partly too general and partly too specific. For instance, case information for nouns or gender information for verb complements are in most cases unwanted. On the other hand, only very little semantic information, if any, will be found in this way, and it will need to be supplied by other means. Furthermore, not all features have the same status. Some are lexical, others are syntactic, semantic and still others are bookkeeping features. What should happen with the acquired information depends on the status of the features. Barg and Walther (1998) talk about generalisable and revisable information. The former are values that are too specific (e.g. case), while the latter are values that should be changeable. They work with a formalism that allows value overwriting, and specify in the grammar what values belong to which class.

3 Implementation

The system we used for our implementation is the Linguistic Knowledge Base (LKB) (Copestake, 2002). It processes unification grammars efficiently (Oepen and Carroll, 2000), and there are large scale HPSG-style grammars available for it (e.g. LinGO (2001)[1]). We implemented the method for acquisition of new entries that is described in the previous section. The generic entry for unknown words should satisfy some minimal requirements: it should prohibit the application of lexical rules (see below), it should restrict the number of complements, and it should help maintain the presence of information (e.g. semantics), such that it is not lost only because an unknown word occurred in the sentence.

In the framework, lexical rules behave like unary phrase structure rules. If the input to such a rule is underspecified, the output might not have a sufficient amount of information filled in to prevent another application of the same rule, and so on. Therefore, lexical rules should not be applied to the unknown words at parse time. At this stage we want to collect the syntactic information of a string as it is used in the given sentence. Afterwards, the lexical rules can be applied (inversely) to the structures that were found, so as to generate the appropriate lexical entries.

Although preliminary tests showed encouraging results, obtaining analyses became quickly harder when the sentences got longer, due to the number of rule applications that was spawned on the underspecified entries. In Head-driven Phrase Structure Grammar (HPSG), the notion of the *head* plays an important role. The constituent which is the head selects for one or more dependents. If the unknown word is a head, then the selection of the dependents is underspecified, which leads to an increased number of solutions. In the current setup, multiple unknown words in a sentence can almost never be treated due to the compounded ambiguity.

A second observation was that the amount of information that is added to the underspecified entry is surprisingly high. We obtained those results in the following way. After the feature structure for the unknown words are extracted from the chart, we *unfill* them. This consists of removing the fea-

[1] Where we refer to the grammar or quote figures relating to it, we assume the current version of the grammar (October 2002).

tures from the feature structures whose value can be inferred from the type hierarchy and the constraints (Götz, 1994). About 30% of the feature structure nodes can be removed (this figure can vary greatly among feature structures and grammars, but this is a typical figure in our experiments). These feature structures are not totally well-typed, but can be made so. This is the information that has been unified in by the context.

The value co-occurrence in these feature structures is in principle unlimited. Horiguchi et al. (1995) specify certain feature co-occurrences in lexical templates to limit the underspecification, with the goal of making the search space smaller, and the lexical entries more specific. The co-occurrence constraints cannot be acquired with the methods here described. A way out is the following. The English LinGO grammar defines for the lexicon a set of special types. These types contain all information for a class of words. A lexical entry consists of nothing but the definition of the string and the semantic relation for the word in addition to the appropriate lexical type. The lexicon thus relies on a highly structured hierarchy of relations. A strategy to increase the specificity of the lexical entries is to collect these types, and use them as input for the unknown word entry. The obvious advantage is that it makes the search space much more restricted than would be the case with one underspecified entry.

There is however also a disadvantage: the number of these types is quite high (463). The amount of ambiguity here is not caused by the rule applications, but by the initial number of lexical entries. To work around that problem we decided to integrate knowledge from external sources. We chose for a statistical PoS tagger, *i.c.* Trigrams'n'Tags (TnT) (Brants, 2000). These taggers return a number of alternatives each associated with a probability, so that the parser can decide what will be used in the analysis. Even when the range of alternatives is left wide open (currently in our experiments the least likely tags that are allowed are 10,000 times less likely than the most likely one), the number of alternatives remains far below the number of lexical types.

The information that can be derived from the tagger output varies with the tag set, but it usually also contains some morphological information. Even though the lexical types are already highly specified, still more value can be filled when it is known that certain morphological rules applied to them. For instance the Penn Treebank tag NNS (Santorini, 1990) indicates a plural noun. While the fact that it is a noun is present in the lexical entry — and can therefore be realised by a type — the fact that it is a plural will restrict it further.

4 Evaluation

There are two aspects that are relevant to be measured: the quality of the newly acquired lexical entries, and the efficiency with which parsing with unknown words takes place.

We have already discussed where the ambiguity arises with unknown words. One of the goals that we will pursue further is to reduce this ambiguity. Obviously, long sentences, with several unknown words should be processable. We have not been able yet to fully assess the impact of the PoS tagger because the mapping from tags to types does not limit the initial number of entries for the unknown word sufficiently yet.

The quality of the acquired lexical entries can be measured as follows. A known entry is removed from the lexicon, and parse trees are constructed for sentences containing the word. The resulting entries are compared to the hand-written entry. The minimum requirement is that a feature structure compatible with the hand-written entry should be found among the results.

5 Outlook

There are a number of issues that we should consider before the newly acquired lexical entries can be used. Among these are the problem of homonyms and the question how long and how many feature structures should be collected for a string.

This approach does not seem to be able to deal with homonyms. The criterion to distinguish known words from unknown ones is whether the string occurs in the lexicon. If of two words that are homonyms one occurs in the lexicon, then that

one will always be chosen to provide the feature structures for the corresponding string. A naive solution would be to reprocess the input considering one of the words as an unknown word, but that is not feasible: how that word can be chosen, without having to analyse the sentence as many times as it contains words? Here as well, external resources like a PoS tagger might provide useful information: the probabilities will be higher if it knows about the homonym.

We also intend to look at how long entries should be collected. Currently new entries are stored in a temporary lexicon. It is a question how long they should stay there and how many feature structures should be collected for a given string. Some words, for instance spelling errors, should (probably) not be stored in the final lexicon. When should they be removed? We expect that these values will have to be determined experimentally.

It seems that it will also be important to have a way to deal with conflicting information. This can be beneficial to deal with information from different sources, for instance from a PoS tagger and from a morphological analyser, or from two feature structures for the same string. Even if we limit ourselves to a tagger, there is still the problem of the high number of solutions that is found when a sentence contains an unknown word. We should be able to generalise over the entries to reduce the number of resulting entries.

6 Summary

We have discussed how new words can be acquired in a large-scale grammar. The basic method has been proposed before, but not with a grammar of a similar coverage. We have described a way how the information concerning unknown words can be restricted in a grammatically sound way, by the definition of lexical types and the use of external knowledge sources. We have discussed evaluation techniques and mentioned a number of issues we will have to deal with.

Acknowledgements

The material presented in this note has benefited from discussions with Ulrich Callmeier, Ann Copestake, Dan Flickinger, Bernd Kiefer, Stephan Oepen and Melanie Siegel. We also thank three anonymous reviewers for their comments. The research was funded by the German Research Fund DFG in the Collaborative Research Centre SFB 378 *Resource-Adaptive Cognitive Processes*, subproject Performance Modelling for Declarative Grammar Models (MI1 PERFORM).

References

Petra Barg and Markus Walther. 1998. Processing unknown words in HPSG. In *Proceedings of the 17th International Conference on Computational Linguistics*, volume I, pages 91–95, Université de Montréal, Montreal, Quebec, Canada, August 10–14. Also as cs.CL/9809106 from `http://xxx.lanl.gov/`.

Thorsten Brants. 2000. TnT: A statistical part-of-speech tagger. In *Proceedings of ANLP-2000*, Seattle, WA, 29 April–3 May. Association for Computational Linguistics.

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. Stanford, CA.

Gregor Erbach. 1990. Syntactic processing of unknown words. In P. Jorrand and V. Sgurev, editors, *Artificial Intelligence IV: Methodology, Systems, Applications*, pages 371–381. Amsterdam.

Thilo Götz. 1994. A normal form for typed feature structures. Master's thesis, Seminar für Sprachwissenschaft, Eberhard-Karls-Universität, Tübingen, April.

Keiko Horiguchi, Kentaro Torisawa, and Jun ichi Tsujii. 1995. Automatic acquisition of content words using an HPSG-based parser. In *Proceedings of the Natural Language Processing Pacific Rim Symposium*, pages 320–325, Seoul, Korea, December.

LinGO. 2001. English Resource Grammar. Available on-line. `http://lingo.stanford.edu/ftp/erg.tgz` (26 July 2002).

Stephan Oepen and John Carroll. 2000. Parser engineering and performance profiling. *Natural Language Engineering*, 6(1):81–97, March.

Beatrice Santorini, 1990. *Part-of-speech tagging guidelines for the Penn Treebank project*, June. Third revision, second printing (February 1995).

Markus Walther and Petra Barg. 1998. Towards incremental lexical acquisition in HPSG. In Gosse Bouma, Geert-Jan Kruijff, and Richard Oehrle, editors, *Proceedings of FHCG'98*, pages 289–297, Saarbrücken, August.