Learning Constraint Grammar-style disambiguation rules using Inductive Logic Programming

Nikolaj Lindberg
Centre for Speech Technology
Royal Institute of Technology
SE-100 44 Stockholm, Sweden
nikolaj@speech.kth.se

Martin Eineborg
Telia Research AB
Spoken Language Processing
SE-136 80 Haninge, Sweden
Martin.E.Eineborg@telia.se

Abstract

This paper reports a pilot study, in which Constraint Grammar inspired rules were learnt using the Progol machine-learning system. Rules discarding faulty readings of ambiguously tagged words were learnt for the part of speech tags of the Stockholm-Umeå Corpus. Several thousand disambiguation rules were induced. When tested on unseen data, 98% of the words retained the correct reading after tagging. However, there were ambiguities pending after tagging, on an average 1.13 tags per word. The results suggest that the Progol system can be useful for learning tagging rules of good quality.

1 Introduction

The success of the Constraint Grammar (CG) (Karlsson et al., 1995) approach to part of speech tagging and surface syntactic dependency parsing is due to the minutely handcrafted grammar and two-level morphology lexicon, developed over several years.

In the study reported here, the Progol machine-learning system was used to induce CG-style tag eliminating rules from a one million word part of speech tagged corpus of Swedish. Some 7 000 rules were induced. When tested on unseen data, 98% of the words retained the correct tag. There were still ambiguities left in the output, on an average 1.13 readings per word.

In the following sections, the CG framework and the Progol machine learning system will be presented very briefly.

1.1 Constraint Grammar POS tagging

Constraint Grammar is a system for part of speech tagging and (shallow) syntactic dependency analysis of unrestricted text. In the following, only the part of speech tagging step will be discussed.

The following as a typical 'reductionistic' example of a CG rule which discards a verbal reading of a word following a word unambiguously tagged as determiner (Tapanainen, 1996, page 12):

```
REMOVE (V) IF (-1C DET) ;
```

where V is the target tag to be discarded and -1C DET denotes the word immediately to the left (-1), unambiguously (C) tagged as determiner (DET). There are several types of rules, not only 'reductionistic' ones, making the CG formalism quite powerful. A full-scale CG has hundreds of rules. The developers of English CG report that 99.7% of the words retain their correct reading, and that 93-97% of the words are unambiguous after tagging (Karlsson et al., 1995, page 186). A parser applying the constraints is described in Tapanainen (1996).

1.2 Inductive Logic Programming

Inductive Logic Programming (ILP) is a combination of machine learning and logic programming, where the goal is to find a hypothesis, H, given examples, E, and background knowledge, B, such that the hypothesis along with the background knowledge logically implies the examples (Muggleton, 1995, page 2):

$$B \wedge H \models E$$

The examples are usually split into a positive, E^+, and a negative, E^-, subset.

The ILP system used in this paper, CProgol Version 4.2, uses Horn clauses as the representational language. Progol creates, for each E^+, a most specific clause $\bot_i$ and then searches through the lattice of hypotheses, from specific to more general, bounded by

$$\Box \preceq H \preceq \bot_i$$

to find the clause that maximally compresses the data where $\preceq$ (θ-subsumption) is defined as

$$c_1 \preceq c_2 \iff \exists\theta : c_1\theta \subseteq c_2$$

and $\Box$ is the empty clause. As an example, consider the two clauses:

c_1: `p(X,Y) :- q(X,Y).`
c_2: `p(a,b) :- q(a,b), r(Z).`

where $c_1 \preceq c_2$ under the substitution $\theta = \{X/a, Y/b\}$.

When Progol has found the clause that compresses the data the most, it is added to the background knowledge and all examples that are redundant with respect to this new background knowledge are removed.

More informally, Progol builds the most specific clause for each positive example. It then tries to find a more general version of the clause (with respect to the background knowledge and mode declarations, see below) that explains as many positive and as few negative examples as possible.

Mode declarations specifying the properties of the rules have to be given by the user. A **modeh** declaration specifies the head of the rules, while **modeb** declarations specify what the bodies of the rules to induce might contain. The user also declares the types of arguments, and whether they are input or output arguments, or if an argument should be instantiated by Progol. Progol is freely available and documented in Muggleton (1995) and Roberts (1997).

1.3 The Stockholm-Umeå Corpus

The training material in the experiments reported here is sampled from a pre-release of the Stockholm-Umeå Corpus (SUC). SUC covers just over one million words of part of speech tagged Swedish text, sampled from different text genres (largely following the Brown corpus text categories). The first official release is now available on CD-ROM.

The SUC tagset has 146 different tags, and the tags consist of a part of speech tag, e.g. VB (the verb) followed by a (possibly empty) set of morphological features, such as PRS (the present tense) and AKT (the active voice), etc. There are 25 different part of speech tags. Thus, many of the 146 tags represent different inflected forms. Examples of the tags are found in Table 1. The SUC tagging scheme is presented in Ejerhed et al. (1992).

2 Previous work

Two previous studies on the induction of rules for part of speech tagging are presented in this section.

Samuelsson et al. (1996) describe experiments of inducing English CG rules, intended more as a help for the grammarian, rather than as an attempt to induce a full-scale CG. The training corpus consisted of some 55 000 words of English text, morphologically and syntactically tagged according to the EngCG tagset.

Constraints of the form presented in Section 1.1 were induced based on bigram statistics. Also lexical rules, discarding unlikely readings for certain word forms, were induced. In addition to these, 'barrier' rules were learnt. While the induced 'remove' rules were based on bigrams, the barrier rules utilized longer contexts.

When tested on a 10 000 word test corpus, the recall of the induced grammar was 98.2% with a precision of 87.3%, which means that some of the ambiguities were left pending after tagging (1.12 readings per word).

Cussens (1997) describes a project in which CG inspired rules for tagging English text were induced using the Progol machine-learning system. To its help the Progol system had a small hand-crafted syntactic grammar. The grammar was used as background knowledge to the Progol system only, and was not used for producing any syntactic structure in the final output. The examples consisted of the tags of all of the words on each side of the word to be disambiguated (the target word). Given no unknown words and a tag set of 43 different tags, the system tagged 96.4% of the words correctly.

3 Present work

The current work was inspired by Cussens (1997) as well as Samuelsson et al. (1996), but departs from both in several respects. It also follows up an initial experiment conducted by the current authors (Eineborg and Lindberg, 1998).

Following Samuelsson et al. (1996) local-context and lexical rules were induced. In the present work, no barrier rules were induced. In contrast to their study, a TWOL lexicon and an annotated training text using the same tagset were not available. Instead, a lexicon was created from the training corpus.

Just as in Cussens work, Progol was used to induce tag elimination rules from an annotated corpus. In contrast to his study, no grammatical background knowledge is given to the learner and also word tokens, and not only part of speech tags, are in the training data.

In order to induce the new rules, the context has been limited to a window of maximally five words, with the target word to disambiguate in the middle. A motivation for using a rather small window size can be found in Karlsson et al. (1995, page 59) where it is pointed out that sensible constraints referring to a position relative to the target word utilize close context, typically 1-3 words.

Some further restrictions on how the learning system may use the information in the window have been applied in order to reduce the complexity of the problem. This is described in Section 3.2.

A pre-release of the Stockholm-Umeå Corpus was used. Some 10% of the corpus was put aside to be used as test data, and the rest of the corpus made up the training data. The test data files were evenly distributed over the different text genres.

3.1 Preprocessing

Before starting the learning of constraints, the training data was preprocessed in different ways. Following Cussens (1997), a lexicon was produced from the training corpus. All different word forms in the corpus were represented in the lexicon by one look-up word and an ambiguity class, the set of different tags which occurred in the corpus for the word form. The lexicon ended up just over 86 000 entries big.

Similar to Karlsson et al. (1995), the first step of the tagging process was to identify 'idioms', although the term is used somewhat differently in this study; bi- and trigrams which were always tagged with one specific tag sequence (unambiguously tagged, i.e.) were extracted from the training text. Example 'idioms' are given in Table 1. 1 530 such bi- and trigrams were used.

Following Samuelsson et al. (1996), a list of very unlikely readings for certain words was produced ('lexical rules'). For a word form plus tag to qualify as a lexical rule, the word form should have a frequency of at least 100 occurrences in the training data, and the word should occur with the tag to discard in no more than 1% of the cases. 355 lexical rules were produced this way. The role of lexical rules and 'idioms' is to remove the simple cases of ambiguities, making it possible for the induced rules to fire, since these rules are all 'careful', meaning that they can refer to unambiguous contexts only (if they refer to tag features, and not word forms only, i.e.).

3.2 Rule induction

Rules were induced for all part of speech categories. Allowing the rules to refer to specific morphological features (and not necessarily a complete specification) has increased the expressive power of the rules, compared to the initial experiments (Eineborg and Lindberg, 1998). The rules can look at word form, part of speech, morphological features, and whether a word has an upper or lower case initial character. Although we used a window of size 5, the rules can look at maximally four positions at the same time within the window. Another restriction has been put on which combination of features the system may select from a context word. The closer a context word is to the target the more features it may use. This is done in order to reduce the search space. Each context word is represented as a prolog term with arguments for word form, upper/lower case character and part of speech tag along with a set of morphological features (if any).

A different set of training data was produced for each of the 24 part speech categories. The training data was pre-processed by applying the bi- and trigrams and the lexical rules, described above (Section 3.1). This step was taken in order to reduce the amount of training data — rules should not be learnt for ambiguities which would be taken care of anyway.

Progol is able to induce a hypothesis using only positive examples, or using both positive and negative examples. Since we are inducing tag eliminating rules, an example is considered

Bi- and Trigrams	POS Readings (unambiguous tag sequence)
ett par	ett/DT NEU SIN IND par/NN NEU SIN IND NOM
det är	det/PN NEU SIN DEF SUB/OBJ är/VB PRS AKT
i samband med	i/PP samband/NN NEU SIN IND NOM med/PP
på grund av	på/PP grund/NN UTR SIN IND NOM av/PP
...	...

Table 1: *'Idioms'. Unambiguous word sequences found in the training data.*

positive when a word is incorrectly tagged and the reading should be discarded. A negative example is a correctly tagged word where the reading should be retained. The training data for each part of speech tag consisted of between 4000 and 6000 positive examples with an equivalent number of negative examples. The examples for each part of speech category were randomly drawn from all examples available in the training data.

A noise level of 1% was tolerated to make sure that Progol could find important rules despite the fact that some examples could be incorrect.

3.3 Rule format

The induced rules code two types of information: Firstly, the rules state the number and positions of the context words relative to the target word (the word to disambiguate). Secondly, for each context word referred to by a rule, and possibly also for the target word, the rule states under what conditions the rule is applicable. These conditions can be the word form, morphological features or whether a word is spellt with an initial capital letter or not, and combinations of these things. Examples of induced rules are

```
remove(vb,A) :-
        constr(A,left(feats([dt]))).
remove(ie,A) :-
        constr(A,right_right(feats([def]),
        feats([vb]))).
remove(vb, A) :-
        context(A,left_target(word(att),
        featlist([imp,akt]))).
```

where the first rule eliminates all verbal (vb) readings of a word immediately preceded by a word tagged as determiner (dt). The second rule deletes the infinitive marker (ie) reading of a word followed by any word which has the feature 'definite' (def), followed by a verb (vb). The third rule deletes verb tags which have the features 'imperative' (imp) and 'active voice' (akt) if the preceding word is *att* (word(att)).

As alredy been mentioned, the scope of the rules has been limited to a window of five words, the target word included. In an earlier attempt, the window was seven words, but these rules were less expressive in other respects (Eineborg and Lindberg, 1998).

4 Results

Just under 7 000 rules were induced. The tagger was tested on a subset of the unseen data. Only sentences in which all words were in the lexicon were allowed. Sentences including words tagged as UO were discarded. The UO tag is a peculiarity of the SUC tagset, and conveys no grammatical information; it stands for 'foreign word' and is used e.g. for the words in passages quoting text which is not in Swedish.

The test data consisted of 42 925 words, including punctuation marks. After lexicon lookup the words were assigned 93 810 readings, i.e., on average 2.19 readings per word. 41 926 words retained the correct reading after disambiguation, which means that the correct tag survived for 97.7% of the words. After tagging, there were 48 691 readings left, 1.13 readings per word.

As a comparison to these results, a preliminary test of the Brill tagger also trained on the Stockholm-Umeå Corpus, tagged 96.9% of the words correctly, and Oliver Mason's QTag got 96.3% on the same data (Ridings, 1998). Neither of these two taggers leave ambiguities pending and both handles unknown words, which makes a direct comparison of the figures given above hard.

The processing times were quite long for most of the rule sets — few of them were actually allowed to continue until all examples were exhausted.

5 Discussion and future work

The figures of the experimental tagger are not optimal, but promising, considering that the rules induced is a limited subset of possible rule types.

Part of the explanation for the figure of ambiguities pending after tagging is that there are some ambiguity classes which are very hard to deal with. For example, there is a tag for the adverb, AB, and one tag for the verbal particle, PL. In the lexicon built from the corpus, there are 83 word forms which can have at least both these readings. Thus, turning a corpus into a lexicon might lead to the introduction of ambiguities hard to solve. A lexicon better tailored to the task would be of much use. Another important issue is that of handling unknown words.

To reduce the error rate, the bad rules should be identified by testing all rules against the training data. To tackle the residual ambiguities, the next step will be to learn also different kinds of rules, for example 'select' rules which retain a given reading, but discard all others. Also rules scoping longer contexts than a window of 5–7 words must be considered.

6 Conclusions

Using the Progol ILP system, some 7 000 tag eliminating rules were induced from the Stockholm-Umeå Corpus. A lexicon was built from the corpus, and after lexicon look-up, test data (including only known words) was disambiguated with the help of the induced rules. Of 42 925 known words, 41 926 (98%) retained the correct reading after disambiguation. Some ambiguities remained in output: on an average 1.13 readings per word. Considering the experimental status of the tagger, we find the results encouraging.

Acknowledgments

Britt Hartmann (Stockholm University) answered many corpus related questions. Henrik Boström (Stockholm University/Royal Institute of Technology) helped us untangle a few ILP mysteries.

References

Eric Brill. 1994. Some advances in transformation-based part of speech tagging. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*.

James Cussens. 1997. Part of speech tagging using Progol. In *Proceedings of the 7th International Workshop on Inductive Logic Programming (ILP-97)*, pages 93–108.

Martin Eineborg and Nikolaj Lindberg. 1998. Induction of Constraint Grammar-rules using Progol. In *Proceedings of The Eighth International Conference on Inductive Logic Programming (ILP'98)*, Madison, Wisconsin.

Eva Ejerhed, Gunnel Källgren, Wennstedt Ola, and Magnus Åström. 1992. *The Linguistic Annotation System of the Stockholm-Umeå Project*. Department of General Linguistics, University of Umeå.

Fred Karlsson, Atro Voutilainen, Juha Heikkilä, and Arto Anttila, editors. 1995. *Constraint Grammar: A language-independent system for parsing unrestricted text.* Mouton de Gruyter, Berlin and New York.

Oliver Manson, 1997. *QTAG—A portable probabilistic tagger.* Corpus Research, The University of Birmingham, U.K.

Stephen Muggleton. 1995. Inverse entailment and Progol. *New Generation Computing Journal*, 13:245–286.

Daniel Ridings. 1998. SUC and the Brill tagger. GU-ISS-98-1 (Research Reports from the Department of Swedish, Göteborg University).

Sam Roberts, 1997. *An introduction to Progol.*

Christer Samuelsson, Pasi Tapanainen, and Atro Voutilainen. 1996. Inducing Constraint Grammars. In Miclet Laurent and de la Higuera Colin, editors, *Grammatical Inference: Learning Syntax from Sentences*, pages 146–155. Springer Verlag.

Pasi Tapanainen. 1996. *The Constraint Grammar Parser CG-2.* Department of General Linguistics, University of Helsinki.