University of Leeds
SCHOOL OF COMPUTING
RESEARCH REPORT SERIES
Report 2001.06

Building Treebanks using a Grammar Induction System

by

Menno van Zaanen[1]

March 2001

[1] School of Computing
menno@comp.leeds.ac.uk

1 Background

The increase of computer storage and processing power has opened the way for new, more resource intensive linguistic applications that used to be unreachable. The trend in increase of resources also creates new uses for structured corpora or treebanks. On the other hand, wider availability of treebanks will account for new types of applications. These new applications can already be found in several fields, for example:

- Natural language parsing [Bod98], [Cha97],
- Evaluation of natural language grammars [BAF$^+$91],
- Machine translation [MS99], [Pou00].

Even though the applications rely heavily on the availability of treebanks, in practice it is often hard to find one that is suitable for the specific task. The main reasons for this are that "the costs of annotation are prohibitively time and expertise intensive, and the resulting corpora may be too susceptible to restriction to a particular domain, application, or genre" [KS99].

Grammar induction systems may help to solve the above mentioned problems. These systems learn grammars, which can then be used to parse sentences. Parsing indicates possible structures or completely structures the corpus, making annotation less time and thus less expertise intensive. Furthermore, grammar induction systems can be reused on corpora in different domains.

2 Alignment-Based Learning

An example of a grammar induction system that can be used to build treebanks is Alignment-Based Learning (ABL). It learns structure using a corpus of plain (unstructured) sentences and outputs a treebank based on these sentences. Since ABL is an unsupervised system, it uses plain sentences only. ABL does not need a structured training set to initialise.

ABL has been tested on the ATIS (Air Traffic Information System) corpus from the Penn Treebank containing 716 sentences resulting in non-crossing brackets precision of 85.31 % and non-crossing brackets recall of 89.31 %. Applying the ABL system to the OVIS (Openbaar Vervoer Informatie Systeem) corpus, a Dutch corpus consisting of 6,797 sentences, resulted in 89.25 % non-crossing brackets precision and 85.04 % non-crossing brackets recall. (Non-crossing brackets precision denotes the percentage of *learned* constituents that do not overlap with any constituents in the *original* corpus and non-crossing brackets recall shows the percentage of constituents in the *original* corpus that do not overlap with any constituents in the *learned* corpus.) [vZ00b]

ABL consists of two distinct steps (for a more elaborate review see [vZ00a]):

Alignment Learning During alignment learning, the system builds a search space of possible constituents. What constitutes a constituent is described by Harris's idea which states that constituents of the same type can be substituted by each other [Har51]. The algorithm finds parts of sentences that can be replaced by other parts. This is done by aligning sentences from the unstructured corpus in pairs. Using the string-edit distance algorithm [WF74], it finds the parts of the sentences that are similar in both sentences and the parts of the sentences that are dissimilar. The dissimilar parts can be substituted in pairs and are thus stored as possible constituents.

Selection Learning The alignment learning phase may find possible constituents that overlap. The selection learning phase selects the best (non-overlapping) constituents by searching the space, generated by the alignment learning phase, directed by a probabilistic evaluation function.

3 Problems of unsupervised grammar induction systems

Unsupervised grammar induction systems like ABL do not have any knowledge about what the final treebank should look like, since unsupervised systems are not guided towards the *wanted* treebank. Although, they usually yield less than perfect results, these systems are still useful, for example when building a treebank of an unknown language, when no experts are available or when results are needed quickly.

On the other hand, when experts are available, when more precise results are needed or when there are no time restrictions, the resulting treebank generated by an unsupervised grammar induction system is generally not good enough. The quality of such a treebank can only be improved by post-processing done by experts. As an example, we mention the Penn Treebank which was annotated in two phases (automatic structure induction followed by manual post-processing) [MSM93]. It is probably the most widely used treebank to date.

Instead of choosing for one of the two possibilities of building a treebank (using an induction system or annotating the treebank by hand), we would like to combine the best of both methods. This should result in a system that suggests possible tree structures for the expert to choose from *and* it should learn from the choices made by the expert in parallel.

3.1 (Semi-)Supervised Alignment-Based Learning (SABL)

The ABL system can be easily adapted into a system that indicates reasonably good tree structures *and* learns from the expert's choices. All changes in the algorithm occur in the selection learning phase:

Select n-best constituents Instead of selecting the best constituent only, as in ABL, let the system select the n (say 5) best constituents. These constituents are presented to the expert, who chooses the correct one or, if the correct one is not present, adds it manually.

Learn from the expert's choice ABL's selection of the best constituents from the search space is guided by a probabilistic evaluation function. In order to learn from the choices made by the expert, the probabilities of the chosen constituents should be changed:

- If the correct constituent was already present in the search space, the probability of that constituent should be increased. When a constituent has a high probability, it will have a higher chance to be selected next time.
- If the correct constituent was *not* present in the search space, it should be inserted and the probabilities of the constituents should be adjusted. Since it was the preferred constituent, its probability should be increased as if the constituent was present in the search space already.

Note that increasing the probability is actually a shift in probability mass. The system should be a sound probabilistic model, so the increase of probability is effectively subtracted from the incorrect constituents.

Varying the amount of increase in probability changes the learning properties of the system. A small amount of increase makes the system a slow learner, while a large amount of increase may overfit the system.

Using an unsupervised grammar induction system and manual annotation are two opposing methods in building a treebank. When SABL is set to select only the best constituent and no editing by the expert takes place, it is equivalent to the unsupervised ABL system. On the other hand, when the constituents indicated by the system are completely ignored, the expert is hand tagging the treebank. Depending on the proportion of the two methods used (i.e. how many proposed constituents are used), SABL can be placed anywhere between the two extremes.

When the expert selects or corrects the constituents SABL proposes, SABL will learn from these choices and the quality of the proposed constituents (and thus the quality of the resulting treebank) will improve. Therefore, building a treebank will generally start out with manual annotation, but since the system learns, it will suggest increasingly precise constituents, resulting in a more unsupervised way of annotation.

4 Summary

Recent research in computational linguistics shows the need for more treebanks. Treebanks, however, are not widely available yet and building them is time and expertise intensive. SABL, a semi-supervised version of the unsupervised grammar induction system ABL, can be used to help build new treebanks. Because SABL learns through the expert's choices, building the treebank will progress from mostly manual to unsupervised annotation.

References

[BAF$^+$91] E. Black, S. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings of a Workshop—Speech and Natural Language*, pages 306–311, February 19–22 1991.

[Bod98] Rens Bod. *Beyond Grammar—An Experience-Based Theory of Language*, volume 88 of *CSLI Lecture Notes*. Center for Study of Language and Information (CSLI) Publications, Stanford:CA, USA, 1998.

[Cha97] Eugene Charniak. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 598–603, 1997.

[Har51] Zellig S. Harris. *Structural Linguistics*. University of Chicago Press, Chicago:IL, USA and London, UK, 7th (1966) edition, 1951. Formerly Entitled: Methods in Structural Linguistics.

[KS99] A. Kehler and A. Stolcke. Introduction. In A. Kehler and A. Stolcke, editors, *Proceedings of a Workshop—Unsupervised Learning in Natural Language Processing; Maryland:MD, USA*, June 1999.

[MS99] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. Massachusetts Institute of Technology Press, Cambridge:MA, USA and London, UK, 1999.

[MSM93] M. Marcus, B. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[Pou00] Arjen Poutsma. Data-Oriented Translation—using the Data-Oriented Parsing framework for machine translation. Master's thesis, University of Amsterdam, Amsterdam, the Netherlands, 2000.

[vZ00a] M. van Zaanen. ABL: Alignment-based learning. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING); Saarbrücken, Germany*, pages 961–967. Association for Computational Linguistics (ACL), July 31–August 4 2000.

[vZ00b] M. van Zaanen. Bootstrapping syntax and recursion using alignment-based learning. In Pat Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1063–1070, Stanford:CA, USA, June 29–July 2 2000. Stanford University.

[WF74] Robert A. Wagner and Michael J. Fischer. The string-to-string correction problem. *Communications of the Association for Computing Machinery*, 21(1):168–173, 1974.