Active DOP: A constituency treebank annotation tool with online learning

Andreas van Cranenburgh
Heinrich Heine University of Düsseldorf
Universitätsstraße 1, 40225 Düsseldorf, Germany
cranenburgh@phil.hhu.de

Abstract

We present a language-independent treebank annotation tool supporting rich annotations with discontinuous constituents and function tags. Candidate analyses are generated by an exemplar-based parsing model that immediately learns from each new annotated sentence during annotation. This makes it suitable for situations in which only a limited seed treebank is available, or a radically different domain is being annotated. The tool offers the possibility to experiment with and evaluate active learning methods to speed up annotation in a naturalistic setting, i.e., measuring actual annotation costs and tracking specific user interactions. The code is made available under the GNU GPL license at https://github.com/andreasvc/activedop.

1 Introduction

Treebank annotation is a labor-intensive manual task with various opportunities for automation. This is typically done with bespoke annotation tools (e.g., PTB, FTB, Negra, Tiger) that provide some form of semi-automatic annotation. The Penn treebank was annotated with the help of a rule-based deterministic parser (Marcus et al., 1993). This parser only provided a partial parse with constituents that it was certain about. A similar process was used for the French Treebank (Abeillé et al., 2003). The German Tiger treebank uses a more elaborate approach with two parsers providing candidate analyses (Brants et al., 2002). The first is a cascaded Markov model that provides interactive annotation and can be retrained on user feedback; the second is based on a precision grammar (HPSG) which is not retrained but has the advantage of always being consistent.

Compared to other treebank annotation tools, we believe our tool offers the following advantages:

- Applicable to any constituency treebank without feature engineering or handwritten rules. Discontinuous constituents and function tags are included in the annotation and suggested parses (ignored by most statistical parsers).
- Online learning: updating the grammar is fast and can therefore be done after every sentence instead of only after a larger batch, which makes the tool suitable for low resource settings and rapidly adapting the grammar to a new domain.
- The possibility to explore active learning methods in a naturalistic setting, i.e., measuring actual annotation cost instead of in synthetic simulations.

2 The Parser

Our system is based on the parser presented in van Cranenburgh et al. (2016), a constituency parser supporting discontinuous constituents and function tags. POS tagging and unknown word handling is integrated in the parser. The parser is based on the Data-Oriented Parsing framework (Scha, 1990; Bod, 1992), which views the treebank as a set of exemplars of which arbitrary fragments can be identified as

Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations, pages 38–42
Santa Fe, New Mexico, USA, August 20-26, 2018.

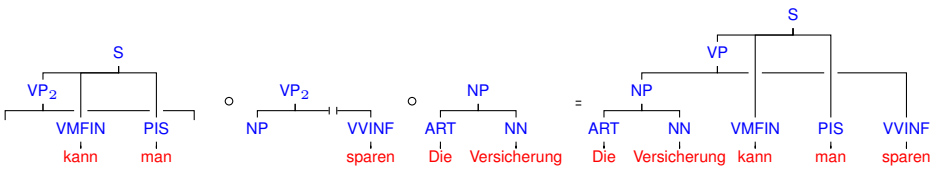

Figure 1: A DOP derivation with discontinuous constituents. Translation: *The insurance one can save.*

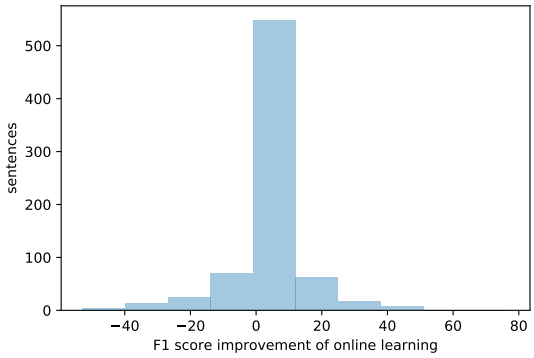

Figure 2: Histogram showing the difference in F1-score with and without online learning in a simulation of annotating the Tiger treebank (higher is better). The improvement is significant with $p < 0.01$.

productive units to analyze new sentences. The parser employs a Tree-Substitution Grammar (TSG) consisting of a set of elementary trees with associated frequencies. The elementary trees are automatically induced from training data in the form of tree fragments attested in two or more trees. Such recurring tree fragments can be efficiently extracted from sets of trees using a tree-kernel approach (Sangati et al., 2010; van Cranenburgh, 2014) which compares pairs of trees in search of common subgraphs. Through the use of indexes of the treebank, this step can be done exactly and exhaustively, instead of needing to resort to approximate methods. Data-driven parsing with discontinuous constituents is done using the grammar formalism of Linear Context-Free Rewriting Systems (LCFRS; Kallmeyer and Maier, 2013), extended to a tree-substitution grammar (van Cranenburgh et al., 2016). Figure 1 shows an example of a derivation with the grammar. Note how discontinuities in elementary trees are marked, specifying where the spans of other elementary trees go as they are combined into a full parse.

This parser is extended with the capability of adding trees to the grammar: online learning. Conceptually, this simply entails adding more exemplars to the model. Since the weights of the elementary trees are simple relative frequencies, there is no expensive parameter estimation involved (compared to, e.g., expectation maximization for latent variable grammars, stochastic gradient descent for deep learning, etc). The set of elementary trees in the grammar is extended with the fragments extracted from the new tree when it is compared to the existing training data. Apart from bookkeeping work such as re-normalizing the relative frequencies and re-indexing grammar rules, updating the grammar is computationally simple and takes less than 1 second. It is therefore feasible to continuously update the grammar during interactive annotation.

Figure 2 shows an evaluation of online learning using a synthetic experiment simulating the annotation of the Tiger treebank. Starting with an initial grammar based on 5000 sentences, candidate parses for new sentences are suggested, and compared to the gold annotation in the treebank. When online learning is enabled, the gold parse is added to the grammar after each sentence. Since both the initial grammar and the new sentences are from the same domain and treebank, the effect is limited, but still there is a clear improvement when online learning is enabled.

Another feature that was added is to improve the handling of sentences that cannot be parsed completely. When a sentence fails to parse, the longest, most probable partial parses are extracted from it in a greedy fashion, until the whole sentence is covered.

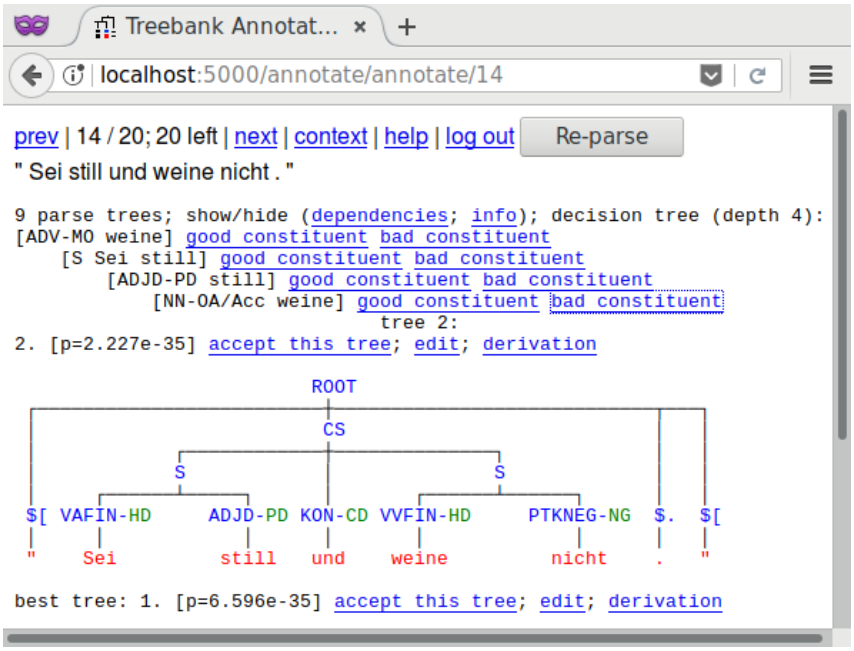

Figure 3: A candidate parse arrived at after following the decision tree of possible parses. The green labels are function tags. Translation: *Be quiet and stop crying.* (from Grimm's fairy tales)

3 User interface

The user interface presents possible candidate parses, which can be selected and edited interactively. Two mechanisms are provided to navigate the potentially long list of similar n-best parses: a decision tree and span constraints.

Upon annotating a new sentence, the user is presented with the most probable analysis. The remaining analyses can be accessed by navigating a decision tree where the nodes ask for the presence of particular constituents that differ between the analyses (Baldridge and Osborne, 2004). We use an entropy-based decision tree method, taking into account the probability distribution of the possible analyses, such that the most probable analyses will require the least number of discriminants. After each discriminant, an example of an analysis confirming to the currently selected discriminants is shown. See Figure 3 for an example.

The decision tree guides the user using the extracted discriminants. Span constraints allow the user to select discriminants. Clicking on a constituent will add a constraint to require or block a particular labeled span, which are then filtered from the list of candidate parses. Additionally, if the desired parse was pruned during parsing, the sentence can be parsed again, potentially producing more trees matching the constraints. See Figure 4 for an example.

In case the correct parse is not among the n-best candidates, the user can select any tree for manual post-editing, in a graphical interface where nodes can be re-attached by drag and drop and labels can be selected from drop down menus. Additionally, a subtree can be selected for re-parsing, after which a replacement can be picked from an n-best list.

4 Active Learning

Active learning is a form of machine learning in which the model takes the initiative of optimizing the selection of new training data to annotate in order to maximize training utility value (for an overview, cf. Settles, 2010). Concretely, this means manipulating the order of sentences to annotate as presented to the user.

A well established technique is uncertainty sampling, which selects sentences of which the model is most uncertain. Uncertainty is measured as the entropy of the probability distribution of possible analyses for a sentence. Using this heuristic, the most difficult sentences will be annotated first. While

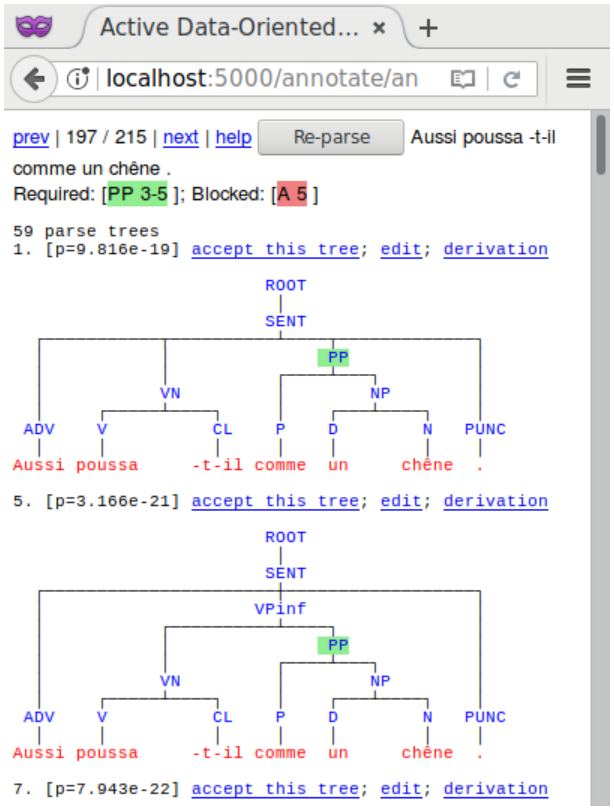

Figure 4: Filtering the list of candidates using span constraints. Here the PP is required, while *chêne* is blocked from being an adjective. Translation: *He also grew like an oak tree* (from Madame Bovary).

this reliably results in steep learning curves, it also means the annotation cost is high and the selected sentences may contain outliers that are difficult but not as useful with respect to the rest of the corpus.

Several works have explored active learning for statistical parsing. Tang et al. (2002) experiments with uncertainty sampling and representativeness ranking, evaluated on a simple treebank of airline reservations. Hwa (2004) presents results on uncertainty sampling with the Penn treebank. Reichart and Rappoport (2009) also adds a clustering method and applies more cognitively grounded cost metrics. Reductions of up to 30 % fewer annotated constituents necessary for a given level of accuracy are shown to be possible in simulations of annotating the Penn treebank. However, whether such reductions also obtain with human annotators has to our knowledge never been confirmed.

In future work we want to explore whether the information in tree fragment distributions and TSG derivations may enable the development of better active learning methods, and run an annotation experiment in which all user interactions are carefully measured.

Acknowledgements

The author is grateful to Laura Kallmeyer and three reviewers for comments. This work was supported by a grant from the German Research Foundation (DFG).

References

Anne Abeillé, Lionel Clément, and François Toussenel. 2003. Building a Treebank for French. In *Treebanks: Building and using parsed corpora*, pages 165–188. Springer.

Jason Baldridge and Miles Osborne. 2004. Active learning and the total cost of annotation. In *Proceedings of EMNLP*, pages 9–16.

Rens Bod. 1992. A computational model of language performance: Data-oriented parsing. In *Proceedings COLING*, pages 855–859.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The Tiger treebank. In *Proceedings of the workshop on treebanks and linguistic theories*, pages 24–41.

Andreas van Cranenburgh. 2014. Extraction of phrase-structure fragments with a linear average time tree kernel. *Computational Linguistics in the Netherlands Journal*, 4:3–16.

Andreas van Cranenburgh, Remko Scha, and Rens Bod. 2016. Data-oriented parsing with discontinuous constituents and function tags. *Journal of Language Modelling*, 4(1):57–111.

Rebecca Hwa. 2004. Sample selection for statistical parsing. *Computational linguistics*, 30(3):253–276.

Laura Kallmeyer and Wolfgang Maier. 2013. Data-driven parsing using probabilistic linear context-free rewriting systems. *Computational Linguistics*, 39(1):87–119.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational linguistics*, 19(2):313–330.

Roi Reichart and Ari Rappoport. 2009. Sample selection for statistical parsers: cognitively driven algorithms and evaluation measures. In *Proceedings of CoNLL*, pages 3–11.

Federico Sangati, Willem Zuidema, and Rens Bod. 2010. Efficiently extract recurring tree fragments from large treebanks. In *Proceedings of LREC*, pages 219–226.

Remko Scha. 1990. Language theory and language technology; competence and performance. In Q.A.M. de Kort and G.L.J. Leerdam, editors, *Computertoepassingen in de Neerlandistiek*, pages 7–22. LVVN, Almere, the Netherlands. Original title: Taaltheorie en taaltechnologie; competence en performance. English translation: `http://remkoscha.nl/LeerdamE.html`.

Burr Settles. 2010. Active learning literature survey. Tech report, `http://burrsettles.com/pub/settles.activelearning.pdf`.

Min Tang, Xiaoqiang Luo, and Salim Roukos. 2002. Active learning for statistical natural language parsing. In *Proceedings of ACL*, pages 120–127.