Annotate – An Efficient Interactive Annotation Tool

Oliver Plaehn, Thorsten Brants
Computational Linguistics
Saarland University
66041 Saarbrücken, Germany
{plaehn,brants}@coli.uni-sb.de

1 Introduction

During the creation of the NEGRA corpus[1] (Skut et al., 1997), we developed very efficient interactive annotation tools. An easy-to-use graphical tool, Annotate, is used to manipulate syntactic structures. Annotate interacts with a part-of-speech tagger and a parser running in the background (Brants, 1999; Brants, 2000), thus facilitating rapid semi-automatic corpus annotation.

Section 2 describes the graphical annotation tool and the annotation process. The parser is presented in section 3. Section 4 gives some conclusions.

2 The Annotation Tool

Annotate provides a comprehensive set of commands for efficient creation and manipulation of syntactic structures. The structures consist of trees with possibly crossing branches, labeled terminal nodes (part-of-speech tags and morphological information), labeled edges (grammatical functions), and labeled non-terminal nodes (phrase categories). In addition, so-called secondary links can be drawn between arbitrary nodes, forming directed graphs. These can be used for incorporating structure sharing information, e.g. for multiple PP attachment, to represent ellipses, or to resolve anaphors. Annotate provides immediate graphical feedback on all changes applied to the syntactic structure. Other features of Annotate include a search function, basic tokenization commands (for splitting or merging words, moving sentence boundaries, etc.), postscript output, an undo function, and online help for the different label sets.

Annotate's most notable feature is its interface to an external parser/tagger, which allows for a semi-automatic annotation procedure. First, the tagger determines a part-of-speech tag for each word of the current sentence. Based on the tags' probabilities, the tagger distinguishes reliable and unreliable assignments. Unreliable assignments are highlighted and the annotator is prompted for confirmation or correction. Second, the syntactic structure is built bottom-up. The parser incrementally suggests new phrases, based on the already constructed partial syntactic structure and the part-of-speech tags. If the suggested phrase is correct, the annotator accepts it. Otherwise, she rejects it and may call the parser for a new suggestion. Alternatively, she may decide to insert a new phrase manually. The internal structures used by the parser are updated, and it again suggests a new phrase based on the existing partial structure. In addition to phrase hypotheses, the parser suggests labels for nodes and edges. Reliable and unreliable assignments are distinguished, and the annotator is prompted for confirmation or correction of the unreliable assignments.

[1] For availability, please check
http://www.coli.uni-sb.de/sfb378/negra-corpus

3 The Parser

Annotate interacts with a parser running in the background to produce syntactic structures on-line. It sends a partial annotation to the parser, which returns one new element. The user can either accept or reject this new element.

The parser is based on the part-of-speech tagger TnT (Brants, 2000)[2] and *Cascaded Markov Models* (Brants, 1999). Each layer of the syntactic structure is represented by a separate Markov Model, starting with the part-of-speech layer. Higher layers are a generalization of the part-of-speech tagging layer. For the purpose of annotation, the tagger and the parser both run in interactive mode.

First, a sequence of part-of-speech tags is generated on-line for a given sentence. In order to facilitate error detection, the tagger generates alternative tags together with their probabilities $P(t_1)$ and $P(t_2)$, giving rise to a reliability measure. If the distance (their quotient) is large, the best tag is simply added to the annotation. If their distance is small, the human annotator is asked for confirmation. We choose a threshold θ and classify the best tag t_1:

$$\text{reliable, if } \frac{P(t_1)}{P(t_2)} \geq \theta\text{; unreliable, if } \frac{P(t_1)}{P(t_2)} < \theta.$$

θ is chosen empirically such that the expected accuracy of all reliable cases is above 99%.

[2] cf. http://www.coli.uni-sb.de/~thorsten/tnt

For part-of-speech tagging in the NEGRA corpus, approx. 85% of the tags are classified as reliable with $\theta = 100$. These have an accuracy of 99.2%. Such a level of accuracy is usually very hard to achieve even for human annotators. The remaining 15% are classified as unreliable with an accuracy of 83.0%. The human annotator mainly needs to concentrate on these 15%, which speeds up the process enormously.

Having finished the part-of-speech annotation, it is sent to the parser. The parser builds a lattice with all phrase hypotheses for layer 1 of the resulting structure. With the help of a Markov Model, the best phrase hypothesis is selected and presented to the annotator (please see (Brants, 1999) for details on the selection process). If the user accepts the phrase, all hypotheses that are not compatible with this phrase are removed from the lattice. If the user rejects it, this phrase is removed from the lattice. Probabilities are re-calculated, and again the best hypothesis is displayed to the annotator.

For a given annotation with n structural layers, $n+1$ lattices are generated and processed using Cascaded Markov Models. At each point, the phrase with the highest probability is displayed. The suggestion of the parser is correct in approx. 70% of the cases. This low number is partly due to crossing branches in the corpus which cannot be handled by the parser. Nevertheless, the human annotator needs to take care of less than a third of all phrases.

The program guides the human annotator through the syntactic structure. We prefer this incremental, bottom-up approach over the immediate generation of complete structures and subsequent error correction by a human because our approach is much faster to perform and much less error-prone.

Labels for grammatical functions (edge labels) are inserted using Markov Models (Brants et al., 1997). As in the case of part-of-speech tagging, a reliability measure is employed in order to significantly reduce the work by concentrating on unreliable tags. 91% of the assignments are classified as reliable, with an accuracy of 99.6%; the remaining 9% are classified as unreliable with an accuracy of 79.2%.

4 Conclusions

The semi-automatic process described above facilitates a very rapid annotation. A trained annotator needs on average 50 seconds/sentence (approx. 1,300 tokens/hour) for part-of-speech plus structural annotation in the NEGRA corpus. This is the fastest structural annotation reported in the literature.

The tools are not restricted to a particular tagset or annotation scheme. It is easy to train the tagger and parser on a small amount of annotated data and to subsequently apply them to annotating according to the new scheme. Using the newly annotated data for training improves the parser's performance and thereby constitutes a bootstrapping approach. In addition to their use for creating the NEGRA corpus, the tools are used for the annotation of the Verbmobil corpora, consisting of transcribed German (Stegmann et al., 1998), English, and Japanese dialogues. Furthermore, trained versions for the Penn Treebank (Marcus et al., 1993) and the Susanne corpus (Sampson, 1995) exist.

To sum up, our annotation tools facilitate comfortable, customizable and fast semi-automatic part-of-speech and syntactic annotation. The tools run under Solaris and Linux and are freely available for research purposes[3].

Acknowledgements

The work on the annotation tools was carried out in the DFG Sonderforschungsbereich 378 *Resource-Adaptive Cognitive Processes*, project C3 *Concurrent Grammar Processing*, and is now continued in the DFG Project TIGER *Linguistic Annotation of a German Corpus*.

We would like to thank the numerous annotators, who have used and are using these tools, for their work and for valuable suggestions for improvements.

References

Thorsten Brants, Wojciech Skut, and Brigitte Krenn. 1997. Tagging grammatical functions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing EMNLP-97*, Providence, RI, USA.

Thorsten Brants. 1999. Cascaded Markov models. In *Proceedings of 9th Conference of the European Chapter of the Association for Computational Linguistics EACL-99*, Bergen, Norway.

Thorsten Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing ANLP-2000*, Seattle, WA.

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

Geoffrey Sampson. 1995. *English for the Computer.* Oxford University Press, Oxford.

Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing ANLP-97*, Washington, DC.

Rosemary Stegmann, Heike Schulz, and Erhard Hinrichs. 1998. Stylebook for the German treebank in Verbmobil. Verbmobil report, Seminar für Sprachwissenschaft, Universität Tübingen, Germany.

[3] cf. `http://www.coli.uni-sb.de/sfb378/negra-corpus/annotate`