A New Approach To Treebank Creation

Stephen G. Eubank , 柏岡 秀紀 , Ezra W. Black
ATR 音声翻訳通信研究所

1 Introduction

A *treebank* is a body of natural-language text which has been grammatically annotated by hand, in terms of some previously-established scheme of grammatical analysis. Treebanks have been used within the field of natural-language processing as a source of training data for statistical part-of-speech taggers [1, 4, 6, 12, 16] and for statistical parsers [2, 3, 6, 8, 9, 10, 14].

All large-scale treebanks of English produced to date have been based on the technique of "skeleton parsing", in which only an outline or high-level approximation of the syntactic structure of each sentence in the treebank is supplied. The broad coverage and level of detail of the ATR/Lancaster Treebank represent a radical departure from extant large-scale [5, 7, 11] and smaller-scale [13, 15] treebanks. We describe an efficient technique for producing such large-scale treebanks of English in which each sentence of the treebank is given a full and highly detailed parse with respect to a comprehensive broad-coverage grammar of English.

2 The Annotation Process

The annotation process is sketched in Figure 1. Initially a file consists of a header detailing the file name, text title, author, etc., and the text itself, which may be in a variety of formats; it may contain HTML mark-up, and files vary in the way in which, for example, emphasis is represented. The first stage of processing is a scan of the text to establish its format and, for large files, to delimit a sample to be annotated.

The second stage is the insertion of SGML (Standard Generalized Mark-up Language) markup. As with the tagging process, this is done by an automatic procedure with manual correction.

Third, the tagging process described in Section 3 is carried out. The tagged text is then extracted into a file for parsing as described in Section 4.

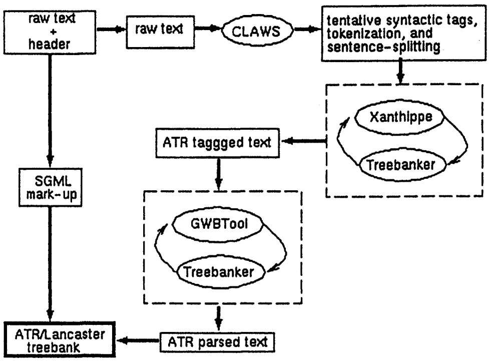


Figure 1: The annotation process for the ATR/Lancaster treebank. Portions inside dashed boxes represent interaction between the human treebanker and the software tools described in this paper.

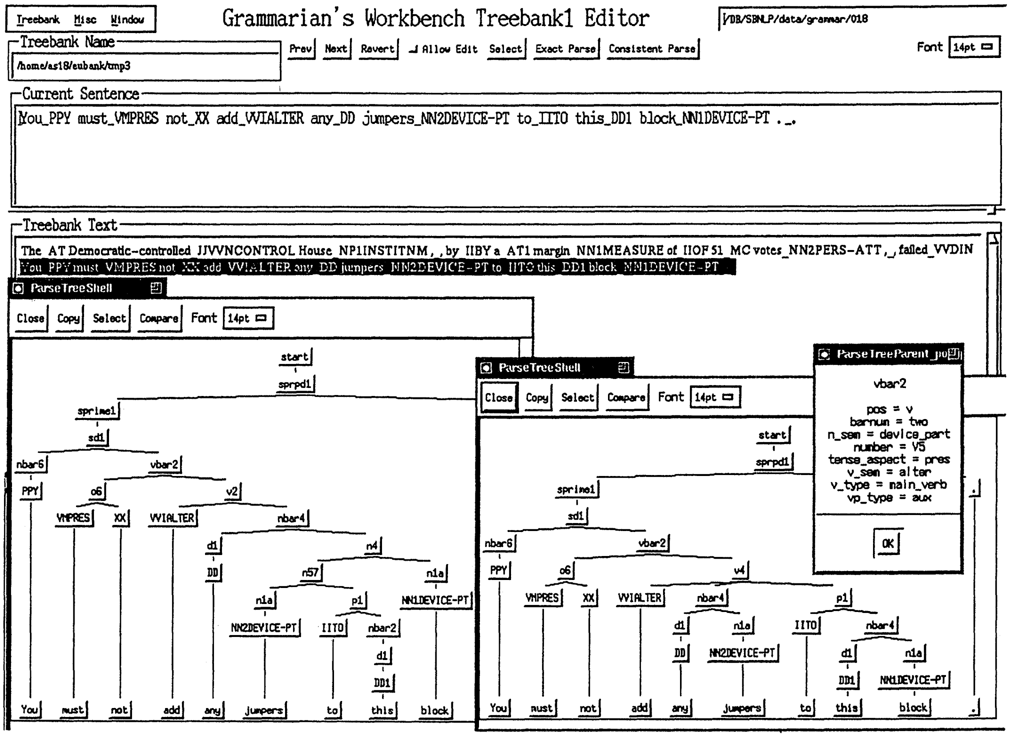
The final stage is merging the parsed and tagged text with all the annotation (SGML mark-up, header information) for return to ATR.

3 Part-Of-Speech Tagging

Part-of-speech tags are assigned in a two-stage process: (a) one or more potential tags are assigned automatically using the Claws HMM tagger [6]; (b) the tags are corrected by a human treebanker using a special-purpose X-windows-based editor, Xanthippe. This displays a text segment and, for each word contained therein, a ranked list of suggested tags. The analyst can choose among these tags or, by clicking on a panel of all possible tags, insert a tag not in the ranked list.

The automatic tagger inserts only the syntactic part of the tag. To insert the semantic part of the tag, Xanthippe presents a panel representing all possible semantic continuations of the syntactic part of the tag selected.

Tokenization, sentence-splitting, and spell-checking are carried out according to rule by the treebankers themselves. However, the Claws tag-



/DB/SBNLP/data/grammar/018

Treebank Name
/home/as18/eubank/tmp3

Current Sentence
You_PPY must_VMPRES not_XX add_VVIALTER any_DD jumpers_NN2DEVICE-PT to_IITO this_DD1 block_NN1DEVICE-PT ._.

Treebank Text
The AT Democratic-controlled JJVVNCONTROL House NP1INSTITNM, , by IIBY a AT1 margin NN1MEASURE of IIOF 51 MC votes_NN2PERS-ATT,_, failed_VVDIN
You PPY must VMPRES not XX add VV1ALTER any DD jumpers NN2DEVICE-PT to IITO this DD1 block NN1DEVICE-PT

vbar2

pos = v
barnum = two
n_sem = device_part
number = V5
tense_aspect = pres
v_sem = alter
v_type = main_verb
vp_type = aux

Figure 2: The GWBTool interface. This picture shows the display after a sentence has been selected (the highlighted sentence in the "Treebank Text" window); its parse forest generated by clicking the "Consistent Parse" button; and two possible trees chosen for comparison. The feature values of the "vbar2" node are also displayed in a pop-up window.

ger performs basic and preliminary tokenization and sentence-splitting, for optional correction using the Xanthippe editor. Xanthippe retains control at all times during the tag correction process, for instance allowing the insertion only of tags valid according to the ATR Grammar.

4 A Treebanker's Workbench

The Grammarian's WorkBench Tool (GWBTool) is a Motif-based X-Windows application which allows a treebanker to interact with the ATR English Grammar in order to produce the most accurate treebank in the shortest amount of time.

The parsing process begins in the Treebank Editor screen of GWBTool with a list of sentences tagged with part-of-speech categories. The treebanker selects a sentence from the list for processing. For example, consider the tagged sentence

```
You  must   not add      any
PPY  VMPRES XX  VV1ALTER DD

jumpers       to   this block        .
NN2DEVICE-PT  IITO DD1  NN1DEVICE-PT .
```

(The correct tag from the ATR Grammar is indicated below each word of the sentence.)

With the click of a button, the Treebank Editor graphically displays the number of parses in the parse forest and the parse forest itself for the sentence in a window. Figure 2 shows two trees from the parse forest for the example sentence. Each node displayed represents a constituent in the parse forest. A shaded constituent node (shading not visible in the figure) indicates that there are alternative analyses of that constituent, only one of which is displayed. By clicking the right mouse button on a shaded node, the treebanker can display a popup menu listing the alternative analyses, any of which can be displayed by selecting the appropriate menu item.

The treebanker can quickly review details of any analysis assigned to a constituent. Clicking the left mouse button on a constituent node pops up a window listing the feature values for that constituent, as shown in Figure 2. In the example, the feature values indicate that the "vbar2" constituent is an auxiliary verb phrase (bar level 2) containing a present-tense verb phrase with noun semantics "device_part" and verb semantics "alter". The fact that the number feature is variable (number=V5) indicates that the number of the verb phrase is not specified by the sentence.

If the parse forest is unmanageably large, the treebanker can easily constrain the possible parses by partially bracketing the sentence. The treebanker selects a range of text with a mouse and hits a key to define a constituent. GWBTool then displays the parse forest containing only those parses which are consistent with the partial bracketing (i.e. those with no constituents which violate the constituent boundaries in the partial bracketing).

The grammar contains 17 possible parses for the example sentence. By constraining the parses to be consistent with the partial bracketing

```
You must not add any
jumpers [to this block],
```

the number of consistent parses is reduced to 9. The parse tree shown on the left in Figure 2 no longer appears in the parse forest generated by GWBTool because it implies the partial bracketing "[[jumpers [to this]] block]", which is inconsistent with "[to this block]". Instead, only parses such as that shown on the right in Figure 2, which contain "to this block" as a constituent, appear. Note that the treebanker need not specify any labels in the partial bracketing, only constituent boundaries.

The process described above is repeated until the treebanker can narrow the parse forest down to a single correct parse. Crucially, for experienced Lancaster treebankers, the number of such iterations is, by now, normally none or one.

5 Output Accuracy

Even though all GWBTool parses are guaranteed to be acceptable to the ATR Grammar, ensuring consistency and accuracy of output has required considerable planning and effort. Of all the parses output for a sentence being treebanked, only a small subset are appropriate choices, given the sentence's meaning in the document in which it occurs. The five Lancaster treebankers had to undergo extensive training over a long period, to understand the ATR Grammar well enough to make the requisite choices.

This training was effected in three ways: a week of classroom training was followed by four months of daily email interaction between the treebankers and the creator of the ATR Grammar; and once this training period ended, daily Lancaster/ATR email interaction continued, as well as constant consultation among the treebankers themselves. A body of documentation and lore was developed and frequently referred to, concerning how all semantic and certain syntactic aspects of the tagset, as well as various grammar rules, are to be applied and interpreted. (This material is organized via a menu system, and updated at least weekly.) A searchable version of files annotated to date, and a list of past tagging decisions, ordered by word and by tag, are at the treebankers' disposal.

In addition to the constant dialogue between the treebankers and the ATR grammarian, Lancaster output was sampled periodically at ATR, hand-corrected, and sent back to the treebankers. In this way, quality control, determination of output accuracy, and consistency control were handled via the twin methods of sample correction and constant treebanker/grammarian dialogue.

With regard both to accuracy and consistency of output analyses, individual treebanker abilities clustered in a fortunate manner. Scoring of thousands of words of sample data over time revealed that three of the five treebankers had low parsing and tagging error rates,

What is fortunate about this clustering of abilities is that the less able treebankers were also much less prolific than the others, producing only 30% of the total treebank. Therefore, we are provisionally excluding this 30% of the treebank (currently about 150,000 words) from use for parser training, though we are experimenting with the use of the entire treebank for tagger training. Finally, parsing and tagging consistency among the first three treebankers appears high.

6 Conclusion

Over the next several years, the ATR/Lancaster Treebank of American English will form the basis for the research of ATR's Statistical Parsing Group in statistical parsing, part-of-speech tagging, and related fields.

However, the techniques and tools for computer-aided parsing described in this paper are not specific to the ATR grammar and treebank. A similar approach could be applied to build large treebanks for other mature, detailed, broad-coverage grammars.

7 Acknowledgements

The authors would like to thank David Magerman for his help in the design and implementation of GWBTool.

References

[1] E. Black, F. Jelinek, J. Lafferty, R. Mercer, S. Roukos. 1992. Decision tree models applied to the labelling of text with parts-of-speech. In *Proceedings, DARPA Speech and Natural Language Workshop*, Arden House, Morgan Kaufman Publishers.

[2] E. Black, R. Garside, and G. Leech, Editors. 1993. *Statistically-Driven Computer Grammars Of English: The IBM/Lancaster Approach.* Rodopi Editions. Amsterdam.

[3] E. Brill. 1993. Automatic grammar induction and parsing free text: A Transformation-based approach. In *Proceedings, 31st Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics.

[4] E. Brill. 1994. Some Advances in Transformation-Based Part of Speech Tagging. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 722-727, Seattle, Washington. American Association for Artificial Intelligence.

[5] E. Eyes and G. Leech. 1993. Syntactic Annotation: Linguistic Aspects of Grammatical Tagging and Skeleton Parsing. Chapter 3 of Black et. al. 1993.

[6] R. Garside, G. Leech, G. Sampson, Editors. 1987. *The Computational Analysis of English.* London, Longman.

[7] R. Garside and A. McEnery. 1993. Treebanking: The Compilation of a Corpus of Skeleton-Parsed Sentences. Chapter 2 of Black et. al. 1993.

[8] F. Jelinek, J. Lafferty, D. Magerman, R. Mercer, A. Ratnaparkhi, S. Roukos. 1994. Decision Tree Parsing using a Hidden Derivation Model. In *Proceedings, ARPA Workshop on Human Language Technology*, pages 260-265, Plainsboro, New Jersey, ARPA.

[9] D. M. Magerman and M. P. Marcus. 1991. Pearl: A Probabilistic Chart Parser. In *Proceedings, European ACL Conference*, March 1991, Berlin, Germany.

[10] D. M. Magerman. 1995. Statistical Decision-Tree Models for Parsing. In *Proceedings, 33rd Annual Meeting of the Association for Computational Linguistics*, pages 276–283, Cambridge, Massachusetts, Association for Computational Linguistics.

[11] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19.2:313-330.

[12] B. Merialdo. 1994. Tagging English Text with a Probabilistic Model. *Computational Linguistics*, 20.2:155-171.

[13] G. Sampson. 1994. *English for the Computer.* Oxford, Oxford University Press.

[14] S. Sekine and R. Grishman. 1995. A Corpus-based Probabilistic Grammar with Only Two Non-terminals. In *Proceedings, International Workshop on Parsing Technologies*, 1995.

[15] H. van Halteren and T. van den Heuvel. 1990. *Linguistic Exploitation of Syntactic Databases.* Rodopi Editions. Amsterdam.

[16] R. Weischedel, M. Meteer, R. Schwartz, L. Ramshaw, and J. Palmucci. 1993. Coping with Ambiguity and Unknown Words through Probabilistic Models. *Computational Linguistics*, 19.2:359-382.