Beyond Skeleton Parsing: Producing a Comprehensive Large–Scale General–English Treebank With Full Grammatical Analysis

Ezra Black, Stephen Eubank
Hideki Kashioka, David Magerman*
ATR Interpreting Telecommunications Laboratories
2-2 Hikaridai, Seika-cho
Soraku-gun, Kyoto, Japan 619-02
{black,eubank,kashioka}@atr.itl.co.jp

Roger Garside, Geoffrey Leech
Depts of Computing and Linguistics
University of Lancaster,
Bailrigg, Lancaster LA1 4YT, UK
rgg@comp.lancs.ac.uk
G.Leech@cent1.lancs.ac.uk

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 (Black et al., 1992; Brill, 1994; Merialdo, 1994; Weischedel et al., 1993) and for statistical parsers (Black et al., 1993; Brill, 1993; Jelinek et al., 1994; Magerman, 1995; Magerman and Marcus, 1991).

In this article, we present the *ATR/Lancaster Treebank of American English*, a new resource for natural–language–processing research, which has been prepared by Lancaster University (UK)'s Unit for Computer Research on the English Language, according to specifications provided by ATR (Japan)'s Statistical Parsing Group. First we provide a "static" description, with (a) a discussion of the mode of selection and initial processing of text for inclusion in the treebank, and (b) an explanation of the scheme of grammatical annotation we then apply to the text. Second, we supply a "process" description of the treebank, in which we detail the physical and computational mechanisms by which we have created it. Finally, we lay out plans for the further development of this new treebank.

All of the features of the ATR/Lancaster Treebank that are described below represent a radical departure from extant large–scale (Eyes and Leech, 1993; Garside and McEnery, 1993; Marcus et al., 1993) treebanks. We have chosen in this article to present our treebank in some detail, rather than to compare and contrast it with other treebanks. But the major differences between this and earlier treebanks can easily be grasped via a comparison of the descriptions below with those of the sources just cited.

*Current affiliation: Renaissance Technologies Corp., 25 East Loop Road, Suite 211, Stony Brook, NY 11776 USA; Consultant, ATR Interpreting Telecommunications Laboratories, 3–12/94

2 General Description of the Treebank

2.1 Document Selection and Preprocessing

The ATR/Lancaster Treebank consists of approximately 730,000 words of grammatically–analyzed text divided into roughly 950 documents ranging in length from about 30 to about 3600 words.

The idea informing the selection of documents for inclusion in this new treebank was to pack into it the maximum degree of document variation along many different scales—document length, subject area, style, point of view, etc.—but without establishing a single, predetermined classification of the included documents.[1] Differing purposes for which the treebank might be utilized may favor differing groupings or classifications of its component documents. Overall, the rationale for seeking to take as broad as possible a sample of current standard American English, is to support the parsing and tagging of unconstrained American English text by providing a training corpus which includes documents fairly similar to almost any input which might arise.

Documents were obtained from three sources: the Internet; optically–scanned hardcopy "occasional" documents (restaurant take–out menus; fundraising letters; utility bills); and purchase from commercial or academic vendors. To illustrate the diverse nature of the documents included in this treebank, we list, in Table 1, titles of nine typical documents.

In general, and as one might expect, the documents we have used were written in the early to mid 1990s, in the United States, in "Standard" American English. However, there are fairly many

[1] as was done, by contrast, in the Brown Corpus (Kucera and Francis, 1967).

Empire Szechuan Flier (Chinese take-out food)
Catalog of Guitar Dealer
UN Charter: Chapters 1-5
Airplane Exit-Row Seating: Passenger Information Sheet
Bicycles: How To Trackstand
Government: US Goals at G7
Shoe Store Sale Flier
Hair-Loss Remedy Brochure
Cancer: Ewing's Sarcoma Patient Information

Table 1: Nine Typical Documents From ATR/Lancaster Treebank

exceptions: documents written by Captain John Smith of Plymouth Plantation (1600s), by Benjamin Franklin (1700s), by Americans writing in periods throughout the 1800s and 1900s; documents written in Australian, British, Canadian, and Indian English; and documents featuring a range of dialects and regional varieties of current American English. A smattering of such documents is included because within standard English, these linguistic varieties are sometimes quoted or otherwise utilized, and so they should be represented.

As noted above, each document within the treebank is classified along many different axes, in order to support a large variety of different task-specific groupings of the documents. Each document is classifed according to tone, style, linguistic level, point of view, physical description of document, geographical background of author, etc. Sample values for these attributes are: "friendly", "dense", "literary", "technical", "how-to guide", and "American South", respectively. To convey domain information, one or more Dewey Decimal System three-digit classifiers are associated with each document. For instance, for the cv of a physiologist, Dewey 612 and 616 (Medical Sciences: Human Physiology; Diseases) were chosen. On a more mundane, "bookkeeping" level, values for text title, author, publication date, text source, etc. are recorded as well.

An SGML-like markup language is used to capture a variety of organizational-level facts about each document, such as LIST structure; TITLEs and CAPTIONs; and even more recondite events such as POEM and IMAGE. HIGHLIGHTing of words and phrases is recorded, along with the variety of highlighting: italics, boldface, large font, etc. Spelling errors and, where essential, other typographical lapses, are scrupulously recorded and then corrected.

Tokenization (i.e. word-splitting: Edward's $\longrightarrow$ Edward 's) and sentence-splitting (e.g. He said, "Hi there. Long time no see." $\longrightarrow$ (Sentence.1:) He said, (Sentence.2:) "Hi there. (Sentence.3:) Long time no see.") are performed by hand according to predetermined policies. Hence the treebank provides the resource of multifarious correct instances of word- and sentence-splitting.

2.2 Scheme of Grammatical Annotation

Heretofore, all existing large-scale treebanks have employed the grammatical analysis technique of *skeleton parsing* (Eyes and Leech, 1993; Garside and McEnery, 1993; Marcus et al., 1993),[2] in which only a partial, relatively sketchy, grammatical analysis of each sentence in the treebank is provided.[3] In contrast, the ATR/Lancaster Treebank assigns to each of its sentences a full and complete grammatical analysis with respect to a very detailed, very comprehensive broad-coverage grammar of English. Moreover, a very large, highly detailed part-of-speech tagset is used to label each word of each sentence with its syntactic *and* semantic categories. The result is an extremely specific and informative syntactic and semantic diagram of every sentence in the treebank.

This shift from skeleton-parsing-based treebanks to a treebank providing full, detailed grammatical analysis resolves a set of problems, detailed in (Black, 1994), involved in using skeleton-parsing-based treebanks as a means of initializing training statistics for probabilistic grammars (Black et al., 1993). Briefly, the first of these problems, which applies even where the grammar being trained has been induced from the training treebank (Sharman et al., 1990), is that the syntactic sketchiness of a skeleton-parsed treebank leads a statistical training algorithm to overcount, in some circumstances, and in other cases to un-

[2]The 1995-release Penn Treebank adds functional information to some nonterminals (Marcus et al., 1994), but with its rudimentary (roughly-45-tag) tagset, its non-detailed internal analysis of noun compounds and NPs more generally, its lack of semantic categorization of words and phrases, etc., it arguably remains a skeleton-parsed treebank, albeit an enriched one.

[3]A different kind of partial parse- crucially, one generated automatically and not by hand- characterizes the "treebank" produced by processing the 200-million-word Birmingham University (UK) Bank-of-English text corpus with the dependency-grammar-based ENGCG Helsinki Parser (Karlsson et al., 1995).

dercount instances of rule firings in training data (treebank) parses, and thus to incorrectly estimate rule probabilities. The second problem is that where the grammar being trained is more detailed syntactically than the skeleton parsing based training treebank, the training corpus radically underperforms in its crucial job of specifying correct parses for training purposes (Black, 1994).

In addition to resolving grammar training problems, our Treebank provides a means of training non grammar based parsing procedures (Brill, 1993; Jelinek et al., 1994; Magerman, 1995) at new, higher levels of grammatical detail and comprehensiveness.

Treebank sentences are parsed in terms of the *ATR English Grammar*, whose characteristics we will briefly describe.

The Grammar's part of speech tagset resembles the 179 tag Claws tagset developed by UCREL (Eyes and Leech, 1993), but with numerous major and minor differences. One major difference, for instance, is that the ATR tagset captures the difference between e.g. "wall covering", where "covering" is a lexicalized noun ending in -ing, and "the covering of all bets", where "covering" is a verbal noun. In Claws practice, both are NN1 (singular common noun). The ATR tagset innovates the tag type NVVG for verbal nouns. Another major difference is the (limited) use of "subcategorization", e.g. VDBLOBJ for double object verbs (teach Bill Latin, etc.).

Each verb, noun, adjective and adverb in the ATR tagset includes a semantic label, chosen from 42 noun/adjective/adverb categories and 29 verb/verbal categories, some overlap existing between these category sets. These semantic categories are intended for *any "Standard American English" text, in any domain.* Sample categories include: "physical.attribute" (nouns/adjectives/adverbs), "alter" (verbs/verbals), and "interpersonal.act" (nouns/adjectives/adverbs/verbs/verbals). They were developed by the ATR grammarian and then proven and refined via day in day out tagging for six months at ATR by two human "treebankers", then for four months at Lancaster by five treebankers, with daily interactions among treebankers, and between the treebankers and the ATR grammarian.

If we ignore the semantic portion of ATR tags, the tagset contains 165 different tags. Including the semantic categories in the tags, there are roughly 2200 tags. As is the case in the Claws tagset, so called "ditto tags" can be created based on almost any tag of the tagset, for the purpose of labelling multiword expressions. For instance, "will o' the wisp" is labelled as a 4 word singular common noun. This process can add considerable numbers of tags to the above totals.

Sentences in the Treebank are parsed with respect to the *ATR English Grammar*. The Grammar, a feature based context-free phrase structure grammar, is related to the IBM English Grammar as published in (Black et al., 1993), but differs more from the IBM Grammar than our tagset does from the Claws tagset. For instance, the notion of "mnemonic" has no application to the ATR Grammar; the ATR Grammar has 67 features and 1100 rules, whereas the IBM Grammar had 40 features and 750 rules, etc.

The precisely correct parse (as pre-specified by a human "treebanker") figures *among* the parses produced for any given sentence by the ATR Grammar, roughly 90% of the time, for text of the unconstrained, wide open sort that the Treebank is composed of. The job of the treebankers is to locate this exact parse, for each sentence, and add it to the Treebank.

Figure 1 shows two sample parsed sentences from the ATR Treebank (and originally from a Chinese take out food flier). Because it is informative to know which of the 1100 rules is used at a given tree node, and since the particular "nonterminal category" associated with any node of the tree is always recoverable,[4] nodes are labelled with ATR Grammar rule names rather than, as is more usual, with nonterminal names.

3 Producing the Treebank

In this part of the article, we turn from "what" to "how", and discuss the mechanisms by which the ATR/Lancaster Treebank was produced.

3.1 The Software Backbone GWBTool: A Treebanker's Workstation

GWBTool is a Motif-based X-Windows application which allows the treebanker to interact with the ATR English Grammar in order to produce the most accurate treebank in the shortest amount of time.

The treebanking process begins in the Treebank Editor screen of the treebanker's workstation with a list of sentences tagged with part-of-speech categories. The treebanker selects a sentence from the list, for processing. Next, with the click of a button, the Treebank Editor graphically displays the parse forest for the sentence in a mouse-sensitive Parse Tree window (Figure 2). Each node displayed represents a constituent in the parse forest. A shaded constituent node 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. Clicking the left mouse button on a constituent node pops up a window listing the feature values for that constituent.

[4] It is contained in the rule name itself.

```
<S id="39" count=8>
<HIGH rendition="italic">
[start [quo (_( [sprpd23 [sprime2 [ibbar2 [r2 Please_RRCONCESSIVE r2] ibbar2]
[sc3 [v4 Mention_VVIVERBAL-ACT [nbar4 [d1 this_DD1 d1]
[n1a coupon_NN1DOCUMENT n1a] nbar4] [fa1 when_CSWHEN
[v1 ordering_VVGINTER-ACT v1] fa1] v4] sc3] sprime2] sprpd23] )_) quo] start]
</HIGH>
</S>

<S id="48" count=5>
<HIGH rendition ="large">
[start [sprpd22 [coord3 [cc3 [cc1 OR_CCOR cc1] cc3]
[nbar13 [d3 ONE_MC1WORD d3] [j1 FREE_JJSTATUS j1] [n4 [n1a FANTAIL_NN1ANIMAL n1a]
[n1a SHRIMPS_NN1FOOD n1a] n4] nbar13] coord3] sprpd22] start]
</HIGH>
</S>
```

Figure 1: *Two ATR/Lancaster Treebank* Sentences (8 words, italicized; 5 words, large font) from Chinese Take–Out Food Flier

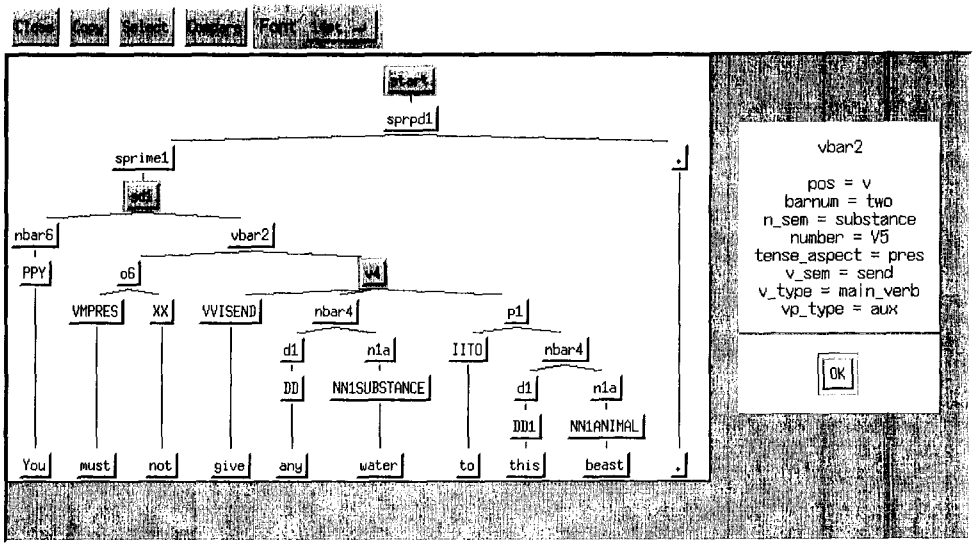

Figure 2: The GWBTool Treebanker's Workstation Parse Window display, showing the parse forest for an example sentence. On the far right, the feature values of the VBAR2 constituent, indicating that the constituent is an auxiliary verb phrase (bar level 2) containing a present-tense verb phrase with noun semantics SUBSTANCE and verb semantics SEND. The fact that the number feature is variable (NUMBER=V5) indicates that the number of the verb phrase is not specified by the sentence. The shaded nodes indicate where there are alternative parses.

The Treebank Editor also displays the number of parses in the parse forest. If the parse forest is unmanageably large, the treebanker can partially bracket the sentence and, again with the click of a button, see the parse forest containing only those parses which are consistent with the partial bracketing (i.e. which do not have any constituents which violate the constituent boundaries in the partial bracketing). 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.

3.2 Two–Stage 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 (?); (b) the tags are corrected by a 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 (see 2.1 above). However, the Claws tagger 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.

3.3 The Annotation Process

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–like mark-up. As with the tagging process, this is done by an automatic procedure with manual correction, using microemacs with a special set of macros.

Third, the tagging process described in section 3.2 is carried out. The tagged text is then extracted into a file for parsing via GWBTool (See 3.1.1).

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

3.4 Staff Training; Output Accuracy

Even though all Treebank parses are guaranteed to be acceptable to the ATR Grammar, insuring 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 manifold devices of the ATR Grammar expertly 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 conjointly 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 parsing error rates (percentage of sentences parsed incorrectly) of 7%, 10%, and 14% respectively, while the other two treebankers' error rates were 30% and 36% respectively. Tagging error rates (percentage of all tags that were incorrect), similarly, were 2.3%, 1.7%, 4.0%, 7.3% and 3.6%. Expected parsing error rate worked out to 11.9% for the first three, but 32.0% for the other two treebankers; while expected tagging error rates were 2.9% and 6.1% respectively.[5]

[5] Almost all tagging errors were semantic.

What is fortunate about this clustering of abilities is that the less able treebankers were also much less prolific than the others, producing only 25% of the total treebank. Therefore, we are provisionally excluding this 25% of the treebank (about 180,000 words) from use for parser training, though we are experimenting with the use of the entire treebank (expected tagging error rate: 3.9%) for tagger training. Finally, parsing and tagging consistency among the first three treebankers appears high.

4 Conclusion

Within the next two years, we intend to produce Version 2 of our Treebank, in which the 25% of the treebank that is currently suitable for training taggers but not parsers, is fully corrected.[6]

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.

References

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.

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

E. Black. 1994. An experiment in customizing the Lancaster Treebank. In Oostdijk and de Haan, 1994, pages 159-168.

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.

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.

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

[6]Seven-tenths of this 25% is already correct, so that the task involved is re-parsing 30% of 25% (= 7.5%) of the treebank.

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

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.

F. Karlsson, A. Voutilainen, J. Heikkila, and A. Anttila. 1995. Constraint Grammar: A Language Independent System for Parsing Unrestricted Text. Mouton de Gruyter: Berlin and New York.

H. Kucera and W. N. Francis. 1967. *Computational Analysis of Present-Day American English.* Brown University Press. Providence, RI.

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

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.

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.

M. Marcus, G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. 1994. The Penn Treebank: Annotating Predicate Argument Structure. *Proceedings, ARPA Human Language Technology Workshop*, Morgan Kaufmann Publishers Inc., San Francisco.

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

N. Oostdijk and P. de Haan, Editors. 1994. *Corpus-Based Research Into Language: In honour of Jan Aarts.* Rodopi Editions. Amsterdam.

R. A. Sharman, F. Jelinek, and R. Mercer. 1990. Generating a Grammar for Statistical Training. In *Proceedings, DARPA Speech and Natural Language Workshop*, Hidden Valley, Pennsylvania.

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.