Linguistic Resources for Speech Parsing

Ann Bies[a], Stephanie Strassel[a], Haejoong Lee[a], Kazuaki Maeda[a], Seth Kulick[a], Yang Liu[b,c], Mary Harper[d,e], Matthew Lease[f]

[a]Linguistic Data Consortium, University of Pennsylvania; [b]ICSI, Berkeley; [c]University of Texas at Dallas;
[d]Purdue University; [e]University of Maryland; [f]Brown University

{bies,strassel,haejoong,,maeda,skulick}@ldc.upenn.edu,
harper@purdue.edu, yangl@hlt.utdallas.edu, mlease@cs.brown.edu

Abstract

We report on the success of a two-pass approach to annotating metadata, speech effects and syntactic structure in English conversational speech: separately annotating transcribed speech for structural metadata, or structural events, (fillers, speech repairs (or edit dysfluencies) and SUs, or syntactic/semantic units) and for syntactic structure (treebanking constituent structure and shallow argument structure). The two annotations were then combined into a single representation. Certain alignment issues between the two types of annotation led to the discovery and correction of annotation errors in each, resulting in a more accurate and useful resource. The development of this corpus was motivated by the need to have both metadata and syntactic structure annotated in order to support synergistic work on speech parsing and structural event detection. Automatic detection of these speech phenomena would simultaneously improve parsing accuracy and provide a mechanism for cleaning up transcriptions for downstream text processing. Similarly, constraints imposed by text processing systems such as parsers can be used to help improve identification of disfluencies and sentence boundaries. This paper reports on our efforts to develop a linguistic resource providing both spoken metadata and syntactic structure information, and describes the resulting corpus of English conversational speech.

1. Motivation for the Creation of this Corpus

In order to apply language processing techniques to speech that have been traditionally applied to text, it is important to address the inherent differences between these two types of inputs. Textual input typically involves words that are broken into sentences and clauses using punctuation that are further organized into chunks such as paragraphs, sections, chapters, articles, books, and so on. Although speech is similar in many ways to text (e.g., it is comprised of words that have the same meaning as in text), it also has many differences, some stemming from the fact that people use different modalities/cognitive processes when processing/producing these inputs/outputs, and others stemming from the different ways in which these two methods of communication are conventionally expressed.

State-of-the-art automatic speech recognition (ASR) systems tend to focus on getting the words correct given that word error rate (WER) has been the metric minimized by such systems. Although WER is dropping, ASR systems do not currently generate/model structural information of the kinds that are available to someone who is reading text. In fact, spontaneous speech is typically not as highly organized as textual material and often contains phenomena such a speech repairs that do not appear in text. These aspects of spoken language present a challenge for systems attempting to bridge the gap between speech processing and natural language processing techniques.

Because automatic detection of sentence breaks and speech repairs is important for bridging between speech and text processing systems, there has been a growing interest in automatically enriching speech recognition output with structural information, further spurred by the structural metadata effort in the DARPA EARS program[1].

Most current state-of-the-art parsing techniques (Charniak, 2000; Collins, 1999) assume that sentence boundaries are given a priori and parse at the sentence level; however, speech recognizers produce only words as output. Although recognizers do work on segments of speech, these rarely correspond to a sentence in text. Speech repairs also pose a serious challenge for accurate parsing (e.g., “I went I mean I left the store” where “I went” is the reparandum, “I mean” is an editing phrase, and “I left” is the alteration in a content replacement speech repair). Recent efforts (e.g., Johnson and Charniak (2004); Charniak and Johnson (2001)) have demonstrated that the presence of speech repairs in the input to a parser hurts overall parse accuracy, and that improved methods to detect these repairs for removal prior to parsing helps alleviate the problem. Hence, effective detection and utilization of sentence boundary and disfluency hypotheses appears to be an important avenue of investigation for parsing spontaneous speech.

The data resources developed for the EARS program have enabled a wide range of research efforts to automatically label speech with metadata events[2] (e.g., Liu et al. (2004), Johnson et al. (2004)). Also, Kahn et al. (2004) demonstrated the impact of automatic sentence boundary detection on improved parsing accuracy, and in the other direction, Johnson et al. (2004) demonstrated that incorporation of syntactic knowledge helps increase the accuracy of automatic metadata detection systems, especially with respect to disfluency detection. These studies suggest there is a

[1] Effective, Affordable, Reusable Speech-to-Text, http://www.darpa.mil/ipto/programs/ears/

[2] The latest SimpleMDE specifications can be found at http://projects.ldc.upenn.edu/MDE/Guidelines/SimpleMDE_V6.2.pdf

good opportunity for exploiting synergy between parsing and structural metadata systems. The 2005 Johns Hopkins University (JHU) Summer Workshop on Parsing and Structural Event Detection (Harper et al., 2005) investigated this synergy given the unified metadata/treebank resource described in this paper. This paper reports our efforts to develop the linguistic resources in support of this workshop effort, and describes the resulting corpus of English conversational speech, annotated for both spoken metadata and syntactic structure.

1.1. Rationale for Developing the Resource

Due to the dual goals of investigating factors impacting the parse accuracy of conversational speech and the effect of syntactic and other knowledge sources on improving MDE (structural metadata extraction) detection accuracy, it was very important for the Hopkins team to have access to a unified conversational speech resource with consistent metadata markups and parse trees. Although the Switchboard Penn Treebank (LDC99T42) is a very useful resource for our experiments (especially for parser training), there were a number of reasons for developing a new resource:

- Metadata and treebanking annotation specifications have been refined since the Switchboard Penn Treebank.
- Metadata annotation in Switchboard Penn Treebank was done largely without reference to audio (Graff & Bird, 2000).
- Metadata annotated SU boundaries were revised during treebanking in Switchboard Penn Treebank without specific guidelines.
- Fisher (and some Switchboard) transcripts had recently been metadata annotated according to the latest SimpleMDE specifications for EARS (LDC2005T24) and so could be reused (SimpleMDE annotated Switchboard data does not overlap that found in the Penn Treebank).

A recent community evaluation (RT-04F[3]) had previously generated ASR and automatic metadata markup, so having a treebank reference would facilitate the evaluation of parse accuracy in a fully-automated system (i.e., parsing ASR words with automatic metadata markup). Given all of these elements, the new data resource would enable the evaluation of the impact of parsing information on MDE and a comparison with the state-of-the-art MDE system performance.

1.2. Data

The JHU Speech Parsing Corpus (LDC2005E15) described here was drawn from transcribed English conversational telephone speech originally developed for the DARPA EARS (Efficient, Affordable, Reusable Speech-To-Text) Program. Some of the phone calls come from Switchboard (LDC97S62) but the majority were newly collected for EARS under the Fisher Protocol (LDC2004E16, LDC2004E29, LDC2005E73). The Fisher data was first carefully transcribed by LDC staff using RT-04 Transcription Specification, Version 3.1[4] (Cieri et al., 2004); for the Switchboard data, ISIP transcripts were used[5]. The data comprised a total of 144 conversations or 140,000 words from the EARS RT04 data, representing 21,000 SUs (or syntactic/semantic units). The following table shows the data size of the dev1, dev2, and eval data sets in number of conversations, number of sentences (i.e., SUs), and number of words. Note that eval is the RT04 evaluation data set, dev1 is the RT03 MDE development and evaluation sets that was used as a development set for RT04, dev2 is an additional development set created for RT04.

	#Conversations	**#SUs**	**#words**
dev1	72	11k	71k
dev2	36	5k	35k
eval	36	5k	34k

Table 1: Sizes of data sets

In the next two sections, we describe the two-pass approach that was used to produce a metadata annotated treebank to support the workshop experiments. We first describe our efforts for separately annotating transcribed speech for structural metadata (structural events, fillers, speech repairs (or edit disfluencies) and SUs, or syntactic/semantic units) in Section 2. We then discuss how these annotations were leveraged to pre-parse the corpus prior to annotation of the corpus for syntactic structure (treebanking constituent structure and shallow argument structure) in Section 3. Finally, we discuss how the two annotations were combined into a single representation and some of the issues we faced in Section 4.

2. Structural Metadata Extraction (MDE) Data and Annotation

Given the transcriptions, the data was annotated for Metadata Extraction (MDE). As “metadata” corresponds to structural events, the information is directly relevant to structural event detection. The goal of MDE is to enable technology that can take raw speech-to-text output and refine it into forms that are more useful to humans and to downstream automatic processes. LDC defined several versions of an MDE annotation task for EARS; the JHU data was annotated to SimpleMDE V6.2[6], which contains the following elements: Fillers (including, e.g., filled pauses and discourse markers), Edit Disfluencies (repetitions, revisions and restarts) and SUs, or syntactic/semantic units. We describe each of these elements below:

Fillers: While the term *filler* has traditionally been synonymous with *filled pause* (Taylor, 1996), SimpleMDE uses the term to encompass a broad set of vocalized space-fillers: filled pauses (FPs), discourse markers (DMs), explicit editing terms (EETs), and

[3] RT Fall 2004 Evaluation Plan, v14, 8/30/04, http://www.nist.gov/speech/tests/rt/rt2004/fall/docs/rt04f-eval-plan-v14.pdf

[4] The RT-04 transcription can be found at http://www.ldc.upenn.edu/Projects/Transcription/rt-04/RT-04-guidelines-V3.1.pdf

[5] http://www.cavs.msstate.edu/hse/ies/projects/switchboard/

[6] The latest SimpleMDE specifications can be found at http://projects.ldc.upenn.edu/MDE/Guidelines/SimpleMDE_V6.2.pdf

asides/parentheticals (A/Ps). Excepting the last category, fillers can be understood as words that do not alter the propositional content of the material into which they are inserted. For example, FPs include non-lexemes, such as *um* or *ah*, that speakers use to indicate hesitation or to maintain control of a conversation. A DM is a word or phrase that functions primarily as a structuring unit of spoken language, such as *actually, now, anyway, see, basically, so, I mean, well, let's see, you know, like, you see*. DMs often signal the speaker's intention to mark a boundary in discourse, like a change in speaker or the beginning of a new topic. There is no exhaustive list of DMs for a given language due to their wide range of functions, colloquial variations, and the difficulty of defining them precisely. Furthermore, words that are used as discourse markers can be used for other purposes. EETs occur during an edit disfluency and consist of an overt statement (e.g., *I'm sorry*) from the speaker recognizing the disfluency. Asides and parentheticals (A/Ps) are different from the other filler types in that they do convey semantic information in the form of a short side comment before returning to the main topic. This may be either on a new topic (asides) or on the same topic of the larger utterance (parentheticals). Both break up the stream of discourse and are often accompanied by noticeable prosodic features.

Edit Disfluencies: Edit disfluencies, or speech repairs, occur when a speaker corrects or alters his utterance, or abandons it entirely and starts over. Edit disfluencies have a more complex internal structure than fillers, consisting of the original utterance (reparandum), an interruption point, an optional editing phase and a correction. There are four types of disfluencies annotated in SimpleMDE: repetitions; revisions; restarts; and complex disfluencies, which consist of multiple or nested edits. In Simple MDE, annotators label only the deletable region (DELREG) of the disfluency, which corresponds to the reparandum. In cases where the reparandum contains multiple disfluent utterances, annotators identify the maximal extent of the disfluent portion, starting with the left edge of the first disfluency and continuing to the right edge (IP) of the final disfluency. (Note that this means that the original MDE annotation does not include the extent information or internal IPs for the multiple edit disfluencies that Treebank treats as nested EDITED nodes; see discussion in Section 4.)

SUs: One of the goals of MDE annotation is the identification of all units within the discourse that function to express a complete thought or idea on the part of the speaker. Within MDE these elements are called SUs (Syntactic, Semantic or Slash Units). As with disfluency annotation, the goal of SU labeling is to improve transcript readability by presenting information in small, structured, coherent chunks. There are four sentence-level SUs. Statements are complete SUs that function as a declarative statement and are marked with /.; questions are complete SUs that function as an interrogative and are marked with /?. Backchannels are an open class of words uttered by the non-dominant speaker to indicate engagement in the conversation and are marked with /@. Incomplete SUs occur when an utterance does not constitute a grammatically complete sentence, phrase or continuer, and does not express a complete thought; these are marked with /-. To enhance inter-annotator consistency, there are also sentence-internal clausal and coordinating SUs (/, and /&).

3. Syntactic (Treebank) Parsing and Annotation Process

Using the existing the MDE annotations for guidance, the data were next annotated for syntactic structure. Treebank annotation was performed in accordance with existing guidelines for treebanking conversational telephone speech (Bies et al. 1995; Taylor 1995), in addition to more recent revisions to guidelines for treebanking (Bies et al., 2005).

Manual treebanking was preceded by the generation of automatic parse trees. Prior to automatic parsing, the first challenge was to convert the existing MDE annotations in RTTM format (a format developed by NIST for the EARS Program that labels each token in the reference transcript according to the properties it displays (e.g., lexeme versus non-lexeme, edit, filler, SU)) into a format appropriate for the parser such that it would generate accurate parses in a form that would require as little hand modification by the Treebank team as possible.

To provide high quality parses, we created scripts[7] to separate the edited material from the fluent part of each SU prior to parsing it using the MDE annotations, and then we parsed the edits and reinserted them into the tree for presentation to the annotators. Some important issues are listed below:

- We tokenized the words in SUs using LDC's script[8]. We later found that this was incomplete due to the interaction between partial words and contractions. This issue was addressed during the summer workshop.
- We chose to maintain all of the punctuation provided in the markup in the SU for parsing because it was likely to enhance parse accuracy and was expected to appear in the final tree annotations.
- For parsing complex edits, we concatenated the contiguous edits into one unit for parsing. In a few cases, edits occur across SUs in MDE annotations.
- Special treatment was required in our scripts for regions unannotated for MDE, complex edits, and SUs that were comprised solely of edited material.
- We used "EDITED" as the non-terminal tag for edit regions inserted into the fluent parse trees. Additionally we added a terminal node for the IP ((DISFL-IP +) at the end of the edits in an attempt to make the tree follow the conventions used in the Switchboard Treebank.

The initial parse trees were produced using Charniak's parser (Charniak, 2000), which was trained on Switchboard and supplemented with Wall Street Journal data (with a 4-1 ratio). The choice of training materials was based on two considerations: Fisher data was known to differ from Switchboard data, and we

[7] We would like to acknowledge Jeremy Kahn for providing advice and scripts that were adapted to generate the parse trees provided to the LDC treebanking team.

[8] http://www.cis.upenn.edu/~treebank/tokenizer.sed

would be largely parsing cleaned up sentences given the high quality metadata markups. Because the input to the parser was cleaned of edits, we also cleaned up the Switchboard trees used to train the parser. By hand, we fixed a few treebanking errors that caused problems for training the parser (e.g., missing top level parentheses, a pre-terminal used as a non-terminal). We also developed a perl script to clean up the parse trees used for training. In particular, we:

- removed CODE lines,
- promoted children of TYPO and then removed the TYPO bracket,
- removed XX constituents and edited constituents,
- removed DFL, IP, RM, and RS constituents,
- for A|B and A^B non-terminal constructions, we kept the A and discarded the B alternative,
- removed remaining carets on non-terminals (e.g., ^A became A).

We used the same parser as was used for the fluent portion of each SU to parse the edits, largely due to the fact that edits in Switchboard comprise a relatively small training resource. We also evaluated the quality of the edit parses produced for several conversation sides and found them to be adequate.

After the fluent portions of the SUs and the edits were parsed, the edits were reinserted by script. The parse trees were then output for each conversation side into a separate file. We generated a parse tree for each SU in each conversation side, using the following format:

SU-ID word-transcript-with-metadata-tags parse-tree

Note that SU-ID was made up of conversation-side_t1_t2_subtype, where t1 was the starting time for this SU, and t2 is the ending time for this SU, both shown in milliseconds; subtype was the SU subtype. Note that in some cases the SU subtype was unannotated. In the transcripts, the metadata tags for filled pauses and discourse markers (<FL_ST> and <FL_END>) and edit disfluencies (<EDIT_ST> and <EDIT_END>) were maintained.

We then augmented the resulting parses with function tags using the Bikel parsing engine[9] (Bikel, 2004) as modified by Kulick (Gabbard et al., 2006), to add semantic function information without altering the parse provided in a novel use of the constraint system built into the parser. This improved the parses given to the annotators and decreased the number of manual corrections necessary.

Treebank annotation was performed in accordance with existing guidelines for treebanking conversational telephone speech (Bies et al. 1995; Taylor 1995), in addition to more recent revisions to guidelines for treebanking (Bies et al., 2005). The treebank annotation included nested EDITED regions for restarts and other repairs, as specified in the existing guidelines for treebanking conversational telephone speech. The contrasts markedly with the MDE annotation of Edit Disfluencies, where only the maximal deletable region of the disfluency is marked and multiple disfluent utterances are not nested.

Treebank annotators had access to the MDE markup during the annotation process, and MDE disfluency annotations were followed whenever possible in the treebank annotation. However, when the MDE annotation was found to be in error, the treebank annotation took precedence and the MDE annotation was automatically corrected.

To establish correspondence between Treebank tokens and MDE tokens, unique tokens IDs from the original MDE annotation files were mapped to the Treebank files. The two token sequences were aligned using a simple algorithm to obtain an N-to-1 mapping. Using this mapping, each Treebank token of the form (*part-of-speech* word) was extended to (*part-of-speech*:[MDE_ID] word), thus creating a consistent mapping.

4. Ensuring Agreement between MDE and Syntactic Annotation Levels

One additional challenge is the existence of discrepancies between the MDE and Treebank treatments of structural metadata. Combining the MDE and Treebank annotations required aligning the two types of annotation at a variety of levels. For the most part, however, the MDE and Treebank annotations agreed:

- MDE SUs were 100% preserved as top level syntactic nodes (primarily S) in treebanking.
- Tokenization followed Treebank requirements.
- MDE edit disfluency spans were mostly preserved in the Treebank, with minor modifications (which were copied back to MDE). Nested EDITEDs and their associated IPs were annotated in the Treebank and copied back to MDE.
- MDE filler annotation was considered during treebanking, and followed if possible.
- Other small orthographic inconsistencies (capitalization and spelling) were not resolved but were documented.

4.1. Tokenization

When tokenization differences emerged it was necessary to follow Treebank guidelines so that the correct constituents were produced, but we were also careful to maintain alignment between MDE and the original parses. For example, a word that includes a clitic, such as *don't* or *it's*, is a single token under MDE annotation unless the clitic is part of an edit dysfluency, but must always be two tokens (*do* and *n't*; *it* and *'s*) for Treebank annotation, since the two tokens receive different part-of-speech tags. In addition, for *it* and *'s*, *it* must be marked as the subject and *'s* as the head of the verb phrase, so there is also a syntactic phrase boundary between the two tokens, e.g.:

```
(S (NP-SBJ it)
   (VP ’s
       (NP-PRD a book)))
```

In these cases, we followed Treebank tokenization and adjusted the MDE tokenization.

In order to align the timestamps in the MDE annotations with the Treebank annotation in these cases, we simply split the time interval of the original token in

[9] Publicly available at http://www.cis.upenn.edu/~dbikel/software.html

half (for something needing to be tokenized into two pieces).

4.2. Orthographic Differences

The following differences were discovered and documented, but as we decided not to change either MDE or Treebank annotation, these differences remain unresolved in the corpus. There were some differences between the treebank and MDE tokens having to do primarily with the word form of abbreviations in MDE. The word form of an abbreviation in MDE included the period marking abbreviation – so, "b." would have the word form "b." in MDE annotations. However, if an abbreviation is a sentence final word, the period could be split off as final punctuation for the Treebank. Hence, the treebank "word" in these cases would be simply "b" (separated from the period).

Capitalization used during transcription was not updated to agree with MDE annotation. In particular, the transcribers capitalized the word they thought started an SU. When the SUs were annotated under MDE guidelines, the original capitalization remained, even if a given word did not actually start an SU any longer.

4.3. Alignment of Nested Disfluencies

Differences also emerged in the treatment of nested disfluencies, which are permitted by Treebank but were not annotated as part of the SimpleMDE task. These are cases where, for example, there are multiple restarts or repairs of a single phrase. SimpleMDE annotates such multiple restarts all together as a single deletable region (DELREG). Treebank, on the other hand, annotates each restart as an EDITED node, and multiple restarts are annotated as nested EDITED nodes. Since each EDITED node or DELREG ends with a marked disfluency insertion point (DISFL-IP), Treebank annotation requires more IPs than MDE annotation does. These nested elements were added during Treebank annotation, and the additional IPs were automatically inserted into the MDE annotation based on the EDITED node annotation in the Treebank.

4.4. Agreement of Filler Annotations

One significant difference emerged in inspecting annotation agreement of fillers. In the case of filled pauses, Treebank and MDE guidelines are well aligned, though some mismatches resulted due to annotation or transcription error. More significant differences were found, however, reflecting differences in annotation standards and ambiguity between them. In addition to ensuring data is given the same interpretation at all levels of annotation, resolving inconsistencies supports establishment of canonical vocabulary for the community, and developing consistent resources for training and evaluating speech processing systems. To this end, we have studied guidelines and data for two corpora that have been both disfluency annotated and treebanked: the Switchboard Penn Treebank and the JHU Speech Parsing corpus. This has led us to develop a preliminary extension to existing treebanking guidelines.

Existing treebanking guidelines are as follows:

- **FPs** are bracketed INTJ and include *uh, um*, etc.
- **DMs** are usually bracketed INTJ. Examples include *well, like, now, see, say, actually*, etc. The DM *you know* is labelled PRN.
- **EETs** are bracketed PRN. Examples include *I mean, excuse me*, etc.
- **Asides** are bracketed PRN.
- **Continuers and assessors** are bracketed INTJ. Examples include *uh-huh, huh, really, exactly, right, yeah, oh, okay*, etc.

Note that the INTJ and PRN constituent types are reused across multiple SimpleMDE categories. This means a bijective mapping between SimpleMDE and syntactic annotations is not possible. While syntactic conventions with regard to INTJ and PRN could certainly be revised, this could require correction of a large amount of existing data. Instead, forgo the desired bijective mapping and accept a more practical compromise for correcting existing treebanks.

1. A word should descend from INTJ in the tree if it is SimpleMDE annotated as a single-word filler.
2. A phrase (i.e., group of contiguous words) should descend from PRN in the tree if it is SimpleMDE annotated as a phrasal filler.
3. Common single-word fillers, such as *like, so, well, actually*, and *now*, etc., should rarely descend from INTJ if they are not SimpleMDE annotated as single-word fillers.
4. Common phrasal fillers, such as *you know* and *I mean*, etc., should rarely descend from PRN if they are not SimpleMDE annotated as phrasal fillers.

The first two rules give a simple unidirectional mapping from SimpleMDE to tree annotation. The latter two rules provide an admittedly weak safety net for catching annotation inconsistency with common fillers. These rules are not completely implemented in the current corpus, but could be in future versions.

5. Analysis of the Two-Pass Process Case Study

The development of this corpus was motivated by the need to have both metadata and treebank annotation in a single representation. The two pass annotation process came about as the result of quickly adding new treebank annotations for this purpose to a corpus that had previously been annotated for metadata. Nonetheless, the two pass approach of marking metadata and then doing the treebanking was found to have certain advantages, in spite of the alignment issues discussed in the previous section.

- Separating the MDE and treebank annotation allows annotators to focus on a single level and type of annotation. This simplifies each annotation task.
- Using the MDE annotations (from the first of the two annotation passes) to separate fluent from disfluent speech allows for improved automatic parsing of each. Higher quality automatic parses greatly reduce the difficulty of treebank annotation and noticeably improve the speed of treebank hand correction (improving the second of the two annotation passes).
- Alignment issues between MDE and treebank annotation led to the discovery and correction of annotation errors in both MDE and treebank,

resulting in more accurate overall combined annotation. This corpus contains far fewer typos, for example, than the Switchboard Penn Treebank (using grep in the respective treebanks, there are six versus 986 TYPO constituents, and 43 “typo” comments versus 743 “^” preterminals). Hence all in all, the new treebank is fitter than Switchboard, with more accurate transcripts since they were double checked during MDE annotation, although the transcripts still had some error that was spotted during treebanking.

The advantages of improved speed and accuracy may outweigh the alignment difficulties, especially with revised guidelines to improve alignment in future metadata and treebank annotation. As a result, this kind of two pass annotation process may be the preferred annotation process for future efforts that combine metadata and treebank annotation, or annotation efforts combining such differing levels of annotation.

6. Conclusion

The combination of dysfluency and syntactic annotation in this corpus represents the first effort of its kind since the Switchboard Penn Treebank. The guidelines of both MDE and Treebank annotation were refined in the meantime, annotation made greater use of audio, and metadata annotations were more tightly followed. As a result, this corpus represents greater agreement across annotation levels than was seen with Switchboard. In addition, by preceding the treebanking effort with MDE annotation, we were able to leverage the MDE annotations to provide high quality parses that reduced the number of manual corrections required by our team. There were challenges associated with reconciling the representations used by MDE and Treebank annotations, but the lessons learned will go far to make future efforts both consistent and efficient. This treebank has already supported a variety of empirical studies on the synergy between parsing and structural metadata (Harper et al. 2005), and we believe that future work in processing and annotating conversational speech data will benefit from the availability of this resource.

7. Acknowledgments

This report is based upon work supported by DARPA under contract number MDA972-02-C-0038, by the National Science Foundation (NSF) under grant numbers 0121285, and by ARDA under contract number MDA904-03-C-1788. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF, DARPA, or ARDA. We would like to thank DARPA and NSA for providing additional support for this treebanking effort. Special thanks are also due to Colin Warner and Justin Mott, without whom this treebank could not have been annotated.

8. References

Bies, A., Ferguson, M., Katz, K. and MacIntyre, R. (1995). *Bracketing Guidelines for Treebank II Style*, Penn Treebank Project, University of Pennsylvania, CIS Technical Report MS-CIS-95-06.

Bies, A., Mott, J. and Warner, C. (2005). *Addendum to the Switchboard Treebank Guidelines*. Linguistic Data Consortium.

Bikel, D. (2004). On the Parameter Space of Lexicalized Statistical Parsing Models. Ph.D. Dissertation. University of Pennsylvania.

Charniak, E. (2000). A Maximum-Entropy-Inspired Parser. In *Proceedings of NAACL'00*, pp. 132-139.

Charniak, E. and Johnson, M. (2001). Edit Detection and Parsing for Transcribed Speech. In *Proceedings of NAACL'01*, pp 118-126.

Cieri, C., Miller, D. and Walker, K. (2004). The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text. In *Proceedings of LREC 2004*.

Collins, M. (1999). Head-Driven Statistical Models for Natural Language Parsing. PhD thesis, University of Pennsylvania.

Engel, D., Charniak, E. and Johnson, M. (2002). Parsing and dysfluency placement. In *Proceedings EMNLP*, pp. 49–54.

Gabbard, R., Kulick, S. and Marcus, M. (2006). Fully Parsing the Penn Treebank. In *Proceedings HLT-NAACL 2006*, New York.

Graff, D. and Bird, S. (2000). Many Uses, Many Annotations for Large Speech Corpora: Switchboard and TDT as Case Studies. In *Proceedings of LREC 2000*.

Harper, M., Dorr, B., Hale, J., Roark, B., Shafran, I., Lease, M., Liu, Y., Snover, M., Yung, L., Krasnyanskaya, A., and Stewart, R. (2005). *Parsing and Spoken Structural Event Detection*. Technical Report, The Johns-Hopkins University, 2005 Summer Research Workshop.

Johnson, M. and Charniak, E. (2004). A TAG-based Noisy Channel Model of Speech Repairs. In *Proceedings of ACL'04*, pp 33-39.

Johnson, M., Charniak, E. and Lease. M. (2004). An improved model for recognizing disfluencies in conversational speech. In *Proceedings of the Rich Text 2004 Fall Workshop*.

Kahn, J., Ostendorf, M. and Chelba, C. (2004). Parsing Conversational Speech Using Enhanced Segmentation. In *Proceedings of HLT-NAACL*.

Lease, M., Charniak, E. and Johnson, M. (2005). Parsing and its Applications for Conversational Speech. In *Proceedings 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*.

Liu, Y., Stolcke, A., Shriberg, E., Hillard, D., Ostendorf, M., Peskin, B. and Harper, M. (2004). The ICSI-SRI-UW Metadata Extraction System. In *Proceedings of the International Conference on Spoken Language Processing*, Jeju, South Korea, October 5-8, 2004.

Taylor, A. (1995). Revision of Meteer, Marie et al., 1995. *Dysfluency Annotation Stylebook for the Switchboard Corpus*. Revised by Ann Taylor, 1995. Ms., University of Pennsylvania.

Taylor, A. (1996). *Bracketing Switchboard: An Addendum to the Treebank II Bracketing Guidelines*. Linguistic Data Consortium.