Skeletons in the parser: Using a shallow parser to improve deep parsing

Mary Swift, James Allen, and Daniel Gildea
Department of Computer Science
University of Rochester
Rochester, NY 14627
{swift,allen,gildea}@cs.rochester.edu

Abstract

We describe a simple approach for integrating shallow and deep parsing. We use phrase structure bracketing obtained from the Collins parser as filters to guide deep parsing. Our experiments demonstrate that our technique yields substantial gains in speed along with modest improvements in accuracy.

1 Introduction

The detailed linguistic analyses generated by deep parsing are an essential component of spoken dialog systems that collaboratively perform tasks with users (e.g., (Allen et al., 2001)). For example, interpretation in the TRIPS collaborative dialog assistant relies on the representation produced by its parser for word sense disambiguation, constituent dependencies, and semantic roles such as agent, theme, goal, etc. Broad coverage unification-based deep parsers, however, unavoidably have problems meeting the very high accuracy and efficiency requirements needed for real-time dialog. On the other hand, parsers based on lexicalized probabilistic context free grammars such those of Collins (1999) and Charniak (1997), which we call shallow parsers[1], are robust and efficient, but the structural representations obtained with such parsers are insufficient as input for intelligent reasoning. In addition, they are not accurate when exact match is considered as opposed to constituent recall and precision and bracket crossing. For example, the standard Collins parser yields an exact match on only 36% on the standard test set (section 23) of the Wall Street Journal Corpus.

In this paper we explore the question of whether preprocessing with a shallow parser can produce analyses that are good enough to help improve the speed and accuracy of deep parsing. Previous work on German (Frank et al., 2002) pursued a similar strategy and showed promising results after considerable effort transforming the output of the shallow parser into useful guidance to the deep parser. We were interested in seeing if we could take a shallow parser off the shelf, namely the Collins parser, and use its output fairly directly to improve the performance of the TRIPS parser. It has been reported that stochastic parsers degrade in performance on domains different than what they were trained on (Hwa, 1999; Gildea, 2001), so there really was an issue whether the output would be good enough. In particular, we are taking the Collins parser trained on the Wall Street Journal and applying it unchanged to spontaneous human-human dialog in an emergency rescue task domain. We have found that there are islands of reliability in the results from the Collins parser that can be used to substantially improve the performance of the TRIPS parser.

The remainder of the paper is organized as follows. Section 2.1 provides background on the Monroe corpus, a set of task-oriented dialogs that is the basis for the parser evaluations. In section 2.2 we describe the TRIPS parser and the representation it produces for reasoning. In section 3 we describe the preliminary evaluations we carried out by running the Collins parser over the Monroe corpus. We then describe our experiments in combining the parsers under different conditions. We look at different conditions, first seeing how this method can improve overall parsing of our corpus, and then with real-time parsing conditions, as required for spoken dialog systems. We find we can get substantial efficiency improvements on the corpus parsing, which mostly disappear when we look at the semi-real-time case. In the latter, however, we do see some improvement in coverage.

2 Background

2.1 The Monroe Corpus

Our data consists of transcribed dialogs between two humans engaged in carefully designed tasks in simulated emergency management situations in Monroe County, New York (Stent, 2001). The scenario was designed to encourage collaborative prob-

[1] We do not intend the "chunking" sense of *shallow parsing* — all our parsers return tree structures.

U We also have to send a road crew there as well
S So we probably can't actually send the ambulance over the bridge
U You're probably right
U Because it's going to take another two hours
U So we'll actually run out of time if we wait for that
U So I guess we'll need to send them
U Actually could we send them up fifteen across two fifty two down three eighty three
U Take that way around
S Wait
S The generator's going downtown
S Right
U The generator is going to two fifty two
S Oh oh I see the problem
U So if we go up fifteen or go south on fifteen
S And then go up three eighty three
U Two fifty two
S Three eighty three
U And that'll get us all the way over to the person with pneumonia or the person who needs the generator
U Say at the most it takes an hour
U It should take no more than an hour to get the generator over to that person
S Okay
S So we have the people taken care of

Figure 1: Excerpt from Monroe dialog

lem solving and mixed initiative interaction involving complex planning and coordination between the participants, so the communication is very spontaneous and interactive. The corpus is split into utterances, and the speech repairs are marked and automatically removed for these tests. Utterances that are incomplete or uninterpretable (by humans) are also marked and eliminated from the corpus. The remaining utterances form the set on which we have been developing and testing the grammar. Figure 1 shows an excerpt from one of the dialogs.

The entire Monroe corpus consists of 20 dialogs ranging from about 7 minutes up to 40 minutes in length. Our tests here focus on a subset of five dialogs that have been used to drive the grammar development: s2, s4, s12, s16 and s17 (henceforth dialogs 1, 2, 3, 4 and 5), constituting 1556 parseable utterances.[2]

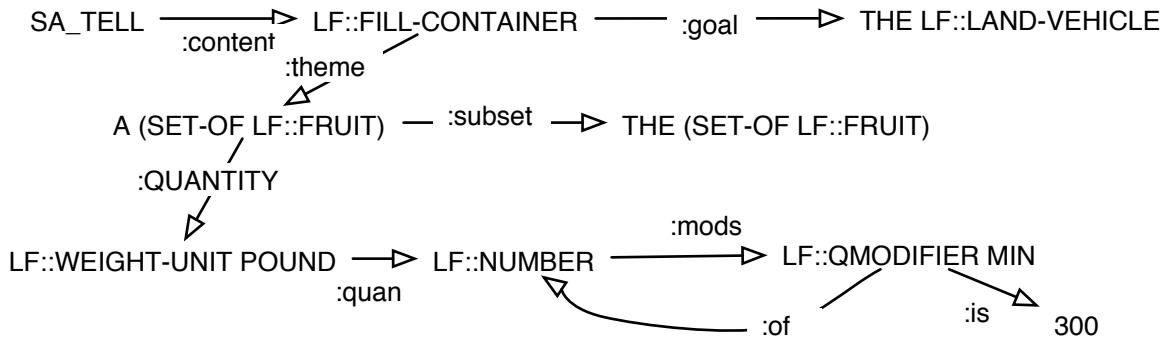
2.2 The TRIPS Parser

The deep parser we used is a robust parsing system developed in the TRIPS system over the past five years being driven from five different domains. The grammatical formalism and parsing framework is essentially a lexicalized version of the formalism described in (Allen, 1995). It is a GPSG/HPSG (Pollard and Sag, 1994) inspired unification grammar of approximately 1300 rules with a rich model of semantic features (Dzikovska, 2004). The parser is an agenda-driven best-first chart parser that supports experimentation with different parsing strategies, although in practice we almost always use a straightforward bi-directional bottom-up algorithm. As an illustration of its flexibility, the modifications required to perform this experiment required adding only one function of ten lines of code. The grammar used for these experiments is the same TRIPS grammar used in all our applications, and the rules have hand-tuned weights. The weights of newly derived constituents are computed exactly as in a PCFG algorithm, the only difference being that the weights don't necessarily add to 1 and so are not probabilities.[3] The TRIPS parser does not use a maximum entropy model (cf. the XLE system (Kaplan et al., 2004)) because there is insufficient training data and it is as yet unclear how such as model would perform at the detailed level of semantic representation produced by the TRIPS parser (see Figure 2 and discussion below).

The rules, lexicon, and semantic ontology are independent of any specific domain but tailored to human-computer practical dialog. The grammar is fairly extensive in coverage (and still growing), and has quite good coverage of a corpus of human-human dialogs in the Monroe domain, an emergency management domain (Swift et al., 2004). The

[2]Parseable utterances exclude utterances that are incomplete or ungrammatical (see (Tetreault et al., 2004).)

[3]We have a version of the grammar that uses a non-lexicalized PCFG model, but it was not used here as it does not perform as well. Thus we are using our best model, making it the most challenging to show improvement.




```
(SPEECHACT V38109 SA_TELL :CONTENT V37618)
  (F V37618 (LF::FILL-CONTAINER LOAD) :GOAL V37800 :THEME V38041
   :TMA ((TENSE PAST) (PASSIVE +)))
    (THE V37800 (LF::LAND-VEHICLE TRUCK))
    (A V38041 (SET-OF (LF::FRUIT ORANGE)) :QUANTITY V37526 :SUBSET V37539)
      (QUANTITY-TERM V37526 (LF::WEIGHT-UNIT POUND) :QUAN V37479)
        (QUANTITY-TERM V37479 LF::NUMBER :MODS (V38268))
          (F V38268 (LF::QMODIFIER MIN) :OF V37479 :IS V37523)
            (QUANTITY-TERM V37523 LF::NUMBER :VALUE 300)
      (THE V37539 (SET-OF (LF::FRUIT ORANGE)))
```

Figure 2: Parser logical form (together with a graphical approximation of the semantic content) for *At least three hundred pounds of the oranges were put in the truck.*

system is in active use in our spoken dialog understanding work in several different domains. It operates in close to real-time for short utterances, but degrades in performance as utterances become longer than 8 or 9 words. As one way to control ambiguity, the grammar makes use of selectional restrictions. Our semantic model utilizes two related mechanisms: first, an ontology of the predicates that are used to create the logical forms, and second, a vector of semantic features associated with these predicates that are used for selectional restrictions. The grammar computes a flattened and unscoped logical form using reified events (see also (Copestake et al., 1997) for a flat semantic representation), with many of its word senses derived from FrameNet frames (Johnson and Fillmore, 2000) and semantic roles (Fillmore, 1968). An example of the logical form representation produced by the parser is shown in Figure 2, in both a dependency graph (upper) and the actual parser output (lower).[4]

[4]Term constructors appearing at the leftmost edge of terms in the parser output are F (relation), A (indefinite entity), THE (definite entity) and QUANTITY-TERM (numeric expressions).

3 Collins Parser Evaluation

As a pilot experiment, we evaluated the performance of the Collins parser on a single dialog of 167 sentences from the Monroe corpus, dialog 3. We extracted context-free grammar backbones from our TRIPS gold standard parses to score the Collins' output against. The evaluation was complicated by difference in tree formats, illustrated in Figure 3. The two parsers use a different (though closely related) set of syntactic categories. The TRIPS structure generally has more levels of structure (roughly corresponding to levels in X-bar theory) than the Penn Treebank analyses (Marcus et al., 1993), in particular for base noun phrases.

We converted the TRIPS category labels to their nearest equivalent in Penn Treebank inventory before scoring the Collins parser in terms of labeled precision and recall of constituents, the standard measures in the statistical parsing community. Overall recall was 32%, while precision was 64%. While we expect the Collins parser to have low recall (it generates fewer constituents overall), the low precision indicates that simply relabeling constituents on a one-for-one basis is not sufficient to resolve the differences in the two formalisms. Precision and recall broken down by constituent type is shown in Table 1.

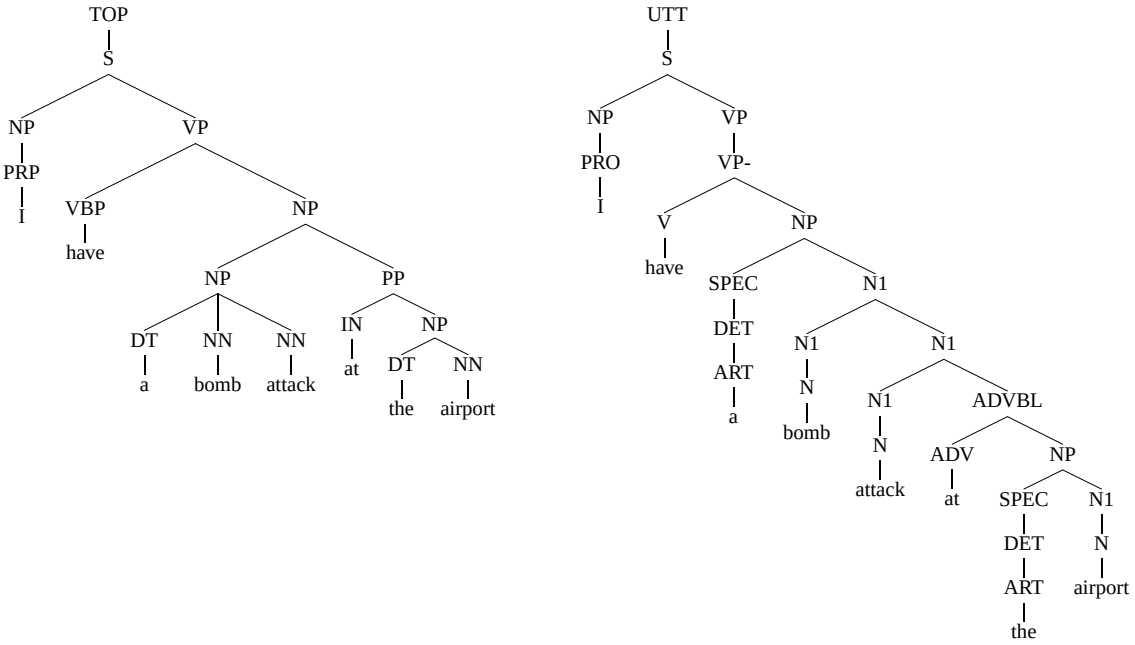


Figure 3: Skeleton tree output from the Collins parser (left) and the TRIPS parser (right) for *I have a bomb attack at the airport*.

However, 82% of the sentences have no crossing brackets in the Collins parse. That is, while the parser may not generate the same set of constituents, it generates very few constituents that straddle the boundaries of any constituent in the TRIPS parse. At this level, the parsers agree about the structure of the sentences to a degree that is perhaps surprising given the very different domain on which the Collins parser is trained. This indicates that the low performance on the other measures has more to do with differences in the annotation style than real mistakes by the Collins parser.

The high level of agreement on unlabeled bracketings led us to believe that the Collins structure could be used as a filter for constituents generated by the TRIPS parser. We tested this strategy in experiments reported in the following section.

4 Experiments

In all the experiments, we used a subset of five dialogs (consisting of 1326 utterances) from the Monroe corpus, described in 2.1. Pilot trials were conducted on dialog 3 (167 utterances), and the experiments were run with the remaining dialogs (1, 2, 4 and 5).

4.1 Method

The first experiment evaluates whether we can extract information from the Collins output that is reliable enough to provide significant improvements to the TRIPS parser. In order to compare our performance with (Frank et al., 2002), the test only uses utterances for which we have a gold-standard. In addition, we report our experiments only on utterances 6 words or longer (with an average of 10.3 words per utterance), as shorter utterances pose little problem for the TRIPS parser and thus running the Collins pre-processing step would not be productive.

We parsed dialogs 1, 2, 4 and 5 with the Collins parser, and extracted the phrase-level bracketing for the most reliable constituents (those which has a precision of at least 60%) in our pilot study: NP, VP and ADVP.[5] From this information we constructed a parse skeleton for each utterance, such as the one shown in Figure 4.

For our experiments we modified the TRIPS parser so that when a constituent is to be added to the chart, if the constituent type and its start and end positions are found in the skeleton then the ranking for that constituent is boosted by a small amount. In pilot trials we determined the optimal boost weight to be 3% (see Table 2).

With a broad coverage grammar, it is possible that the parser could run almost indefinitely on sentences that are difficult to parse. Thus we set an upper limit on the number of constituents that can be added to the chart before the parser quits. The parser runs until it finds a complete analysis or hits this upper

[5]The Collins parse time for the 309 utterances of 6 words or longer was 30 seconds.

label	gold	recall	produced	precision	crossing
ADJ	2	0.0%	0	0.0%	0.0%
ADJP	17	17.6%	7	42.9%	28.6%
ADVP	106	23.6%	35	71.4%	11.4%
CD	17	0.0%	0	0.0%	0.0%
DT	39	0.0%	0	0.0%	0.0%
FRAG	0	0.0%	2	0.0%	0.0%
INTJ	0	0.0%	19	0.0%	0.0%
N	5	0.0%	0	0.0%	0.0%
NNP	5	0.0%	0	0.0%	0.0%
NP	170	79.4%	225	60.0%	8.9%
NPSEQ	5	0.0%	0	0.0%	0.0%
NX	106	0.0%	0	0.0%	0.0%
PP	4	50.0%	37	5.4%	13.5%
PRED	6	0.0%	0	0.0%	0.0%
PRT	0	0.0%	2	0.0%	0.0%
QP	16	0.0%	1	0.0%	100.0%
RB	5	0.0%	0	0.0%	0.0%
S	75	42.7%	83	38.6%	6.0%
SBAR	18	50.0%	17	52.9%	23.5%
SBARQ	0	0.0%	1	0.0%	0.0%
SINV	0	0.0%	2	0.0%	0.0%
SPEC	61	0.0%	0	0.0%	0.0%
SQ	0	0.0%	2	0.0%	0.0%
UTT	185	0.0%	0	0.0%	0.0%
UTTWORD	15	0.0%	0	0.0%	0.0%
VB	6	0.0%	0	0.0%	0.0%
VP	235	43.8%	124	83.1%	7.3%
WHNP	0	0.0%	3	0.0%	0.0%

Table 1: Breakdown of Collins parser performance by constituent type. Recall refers to how many of the gold-standard TRIPS constituents were produced by Collins, precision to how many of the produced constituents matched TRIPS, and crossing brackets to the percentage of TRIPS constituents that were violated by any bracketing produced by Collins.

So [NP I] [VP guess that if [NP we] [VP send [NP one ambulance] to [NP the airport]] [NP we] [VP can [VP get [NP more people off] [ADVP quickly]]]

Figure 4: Skeleton filter for the utterance *So I guess that if we send one ambulance to the airport we can get more people off quickly*.

Boost weight	1%	2%	3%	4%	5%
Speedup factor	1.1	1.3	2.4	2.0	1.2

Table 2: Pilot trials on dialog 3 to determine boost factor.

limit.In the first experiment, this upper limit is set at 10000 constituents. In addition, we performed the same experiments with lower upper limits to explore the question of how much of the parser time is spent on the sentences that hit the maximum chart size limit. In the second experiment we used an upper limit of 5000, and in the third we used an upper limit of 1500 (the standard value for use in our real-time dialog system to avoid long delays in responding).

4.2 Results

Results show significant improvements in the speed of parsing. Table 3 shows the exact match sentence accuracy and timing results for parsing with and without skeletons with a maximum chart size of 10000. The first row shows how many utterances of 6 words or longer were parsed in each dialog. The next two rows show exact match sentence accuracy results for parses obtained with and without

Dialog	1	2	4	5	Total
Utts (6+ words)	83	78	78	70	309
Sentence accuracy w/ skeleton	57.8	50	37.2	52.9	49.5
Sentence accuracy no skeleton	56.6	48.7	35.9	52.9	48.5
Time w/ skeleton	46	85	127	45	303
Time no skeleton	90	190	321	60	661
Speedup Factor	1.9	2.2	2.5	1.3	2.0

Table 3: Sentence accuracy and timing results with maximum chart size 10000 for utterances of 6 or more words.

Dialog	1	2	4	5	Total
Utts (6+ words)	83	78	78	70	309
Sentence accuracy w/ skeleton	57.8	50	37.2	52.9	49.5
Sentence accuracy no skeleton	55.4	48.7	35.9	52.9	48.2
Time w/ skeleton	46	82	126	45	299
Time no skeleton	90	148	286	59	583
Speedup Factor	1.9	1.8	2.3	1.3	1.8

Table 4: Sentence accuracy and timing results with maximum chart size 5000 for utterances of 6 or more words.

skeletons. The next two rows show the total time (in seconds) to parse the dialogs with and without the skeletons. The last row shows the speed up factor (computed as time-without-skeletons/time-with-skeletons).[6]

We see substantial speed-ups in the parser using this technique. The parser using skeletons completed the parses in less than half of the time of the original parser. Looking at individual utterances, 70% were parsed more quickly with the skeletons, while 25% were slower. Overall, our simple approach appears to provide a substantial payoff in speed along with a small improvement in accuracy.

Note that we use a strict criterion for accuracy, so both the correct logical form as well as the correct syntactic structure must be computed by the parser for an analysis to be considered correct in our evaluation. A correct logical form requires correct word sense disambiguation, constituent dependencies, and semantic role assignment (see section 2.2). For example, in some cases the parser produces a structurally correct parse, but selects an inappropriate word sense, in which case the analysis is considered incorrect. One such case is the utterance *You know where the little loop is*, in which the *where* is assigned the sense TO-LOC (which should only be used for trajectories, as in *Where did he go*), when in this utterance the correct sense for *where* is SPATIAL-LOC.

To explore the question of how much of the speed increase is the result of time spent on difficult sentences that cause the parser to reach the maximum chart size limit, we performed the same experiment with a smaller maximum chart size of 5000, shown in Table 4. As expected the speed-up gain declined to 1.8, still quite a respectable gain, and again there is no loss of accuracy.

[6]These experiments were run with CMU Common LISP 18e and a Linux 2.4.20 kernel on a 2 GHz Xeon dual processor with 1.0 GB total memory.

Dialog	1	2	4	5	Total
Utts (6+ words)	83	78	78	70	309
Sentence accuracy w/ skeleton	57.8	48.7	37.2	52.9	49.2
Sentence accuracy no skeleton	55.4	47.4	35.9	52.9	47.9
Time w/ skeleton	47	76	109	45	277
Time no skeleton	74	92	150	59	375
Speedup Factor	1.6	1.2	1.4	1.3	1.4

Table 5: Sentence accuracy and timing results with maximum chart size 1500 for utterances of 6 more words.

As we drop the chart size to 1500, the speed-up drops to just 1.4, as shown in Table 5. However, we have improvements in accuracy using skeletons when we parse with low upper limits. In certain cases the skeleton guides the parser to the correct parse more quickly, so it can be found even when the maximum chart size is reduced. For example, for the utterance *And meanwhile we send two ambulances from the Strong Hospital to take the six wounded people from the airport* (from dialog 1), a correct full sentence analysis is found with the larger maximum chart sizes (5000 or more), but with a maximum chart size of 1500 the correct analysis for this utterance is found only with the help of the skeleton.

Our best results are similar to those reported in (Frank et al., 2002), who show a speed-up factor of 2.26, although they use a much larger maximum chart size (70,000). Because of the differences in grammars and parsers, it is not clear how to fairly compare the chart sizes.

5 Conclusion

With minimal modifications to our deep parser, we have been able to achieve a substantial increase in parsing speed with this technique along with a small increase in accuracy. The experiments reported here investigated this technique using off-line methods. Given our promising results, we are currently working to integrate an on-line shallow parsing filter into our collaborative dialog assistant.

Acknowledgments

We thank Micha Elsner, David Ganzhorn and Allison Rosenberg for verification of accuracy results, and Myroslava Dzikovska and Joel Tetreault for helpful comments and discussion. This research was partially supported by NSF grant IIS-0328810 and DARPA grant NBCH-D-03-0010.

References

James F. Allen, Donna K. Byron, Myroslava O. Dzikovska, George Ferguson, Lucien Galescu, and Amanda Stent. 2001. Towards conversational human-computer interaction. *AI Magazine*, 22(4):27–35.

James F. Allen. 1995. *Natural Language Understanding*. Benjamin Cummings, Redwood City, CA.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, pages 598–603, Menlo Park, August. AAAI Press.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Ann Copestake, Dan Flickinger, and Ivan A. Sag. 1997. Minimal Recursion Semantics: An introduction. Technical report, CSLI, Stanford University, CA.

Myroslava O. Dzikovska. 2004. *A Practical Semantic Representation for Natural Language Parsing*. Ph.D. thesis, University of Rochester.

Charles J. Fillmore. 1968. The case for case. In Emmon Bach and Robert Harms, editors, *Universals in Linguistic Theory*, pages 1–90. Holt, Rinehart and Winston.

Anette Frank, Markus Becker, Berthold Crysmann, Bernd Kiefer, and Ulrich Schaefer. 2002. Integrated shallow and deep parsing: TopP meets HPSG. In *COLING'02*, Taipei.

Daniel Gildea. 2001. Corpus variation and parser performance. In *2001 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 167–202, Pittsburgh, PA.

Rebecca Hwa. 1999. Supervised grammar induction using training data with limited constituent information. In *Proceedings of the 37th Annual Meeting of the ACL*, College Park, Maryland.

Christopher Johnson and Charles J. Fillmore. 2000. The FrameNet tagset for frame-semantic and syntactic coding of predicate-argument structure. In *Proceedings ANLP-NAACL 2000*.

Ronald M. Kaplan, Stefan Riezler, Tracy Holloway King, John T. Maxwell III, Alexander Vasserman, and Richard Crouch. 2004. Speed and accuracy in shallow and deep stochastic parsing. In *NAACL'04*, Boston.

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

Carl Pollard and Ivan Sag. 1994. *Head-driven Phrase Structure Grammar*. Chicago University Press.

Amanda J. Stent. 2001. *Dialogue Systems as Conversational Partners: Applying Conversation Acts Theory to Natural Language Generation for Task-Oriented Mixed-Initiative Spoken Dialogue*. Ph.D. thesis, University of Rochester.

Mary Swift, Myroslava Dzikovska, Joel Tetreault, and James Allen. 2004. Semi-automatic syntactic and semantic corpus annotation with a deep parser. In *LREC'04*, Lisbon.

Joel Tetreault, Mary Swift, Preethum Prithviraj, Myroslava Dzikovska, and James Allen. 2004. Discourse annotation in the Monroe corpus. In *ACL'04 Workshop on Discourse Annotation*, Barcelona.