Discourse Annotation in the Monroe Corpus

Joel Tetreault*, **Mary Swift***, **Preethum Prithviraj***, **Myroslava Dzikovska**†, **James Allen***
* Department of Computer Science, University of Rochester, Rochester, NY, 14620, USA
tetreaul,swift,prithvir,james@cs.rochester.edu
† Human Communications Research Centre, University of Edinburgh
2 Buccleuch Place, Edinburgh EH8 9LW
mdzikovs@inf.ed.ac.uk

Abstract

We describe a method for annotating spoken dialog corpora using both automatic and manual annotation. Our semi-automated method for corpus development results in a corpus combining rich semantics, discourse information and reference annotation, and allows us to explore issues relating these.

1 Introduction

Discourse information plays an important part in natural language systems performing tasks such as text summarization, question-answering systems and collaborative planning. But the type of discourse information that is relevant varies widely depending on domain, genre, number of participants, whether it is written or spoken, etc. Therefore empirical analysis is necessary to determine commonalities in the variations of discourse and develop general purpose algorithms for discourse analysis.

The heightened interest in human language technologies in the last decade has sparked several discourse annotation projects. Though there has been a lot of research, many of the projects focus on a few specific areas of discourse relevant to their respective system. For example, a text summarization system working on texts from the web would not need to know about dialogue modeling or grounding or prosody. In contrast, for a spoken dialogue system that collaborates with a user, such information is crucial but the organization of web pages is not.

In this paper we describe our work in the Monroe Project, an effort targeting the production and use of a linguistically rich annotated corpus of a series of task-oriented spoken dialogs in an emergency rescue domain. Our project differs from past projects involving reference annotation and discourse segmentation in that the semantics and discourse information is generated automatically. Most other work in this area has had minimal semantics or speech act tagging, if anything at all, which can be quite labor intensive to annotate. In addition, our domain is spoken language, which is rarely annotated for the information we are providing. We describe our research on reference resolution and discourse segmentation using the annotated corpus and the software tools we have developed to help us with different aspects of the annotation tasks.

2 Aims of Monroe Project

2.1 Parser Development

One of the aims of the Monroe Project was to develop a wide coverage grammar for spoken dialogue. Since parsing is just an initial stage of natural language understanding, the project was focused not just on obtaining syntactic trees alone (as is done in many other parsed corpora, for example, Penn TreeBank (Marcus et al., 1993) or Tiger (Brants and Plaehn, 2000)). Instead, we aimed to develop a parser and grammar for the production of syntactic parses and semantic representations useful in discourse processing.

The parser produces a domain-independent semantic representation with information necessary for referential and discourse processing, in particular, domain-independent representations of determiners and quantifiers (to be resolved by our reference module), domain-independent representations for discourse adverbials, and tense, aspect and modality information. This necessitated the development of a domain-independent logical form syntax and a domain-independent ontology as a source of semantic types for our representations (Dzikovska et al., 2004). In subsequent sections we discuss how the parser-generated representations are used as a basis for discourse annotation.

2.2 Reference Resolution Development

In spoken dialogue, choice of referring expression is influential and influenced by the main entities being discussed and the intentions of the speaker. If an entity is mentioned frequently, and thus is very important to the current topic, it is usually pronominalized. Psycholinguistic studies show that salient terms are usually evoked as pronouns because of the lighter inference load they place on the listener. Because pronouns occur frequently in discourse, it is very important to know what they resolve to, so the entire sentence can be processed correctly. A corpus annotated for reference relations allows one to compare the performance of different reference algorithms.

2.3 Discourse Segmentation

Another research area that can benefit from a discourse-annotated corpus is discourse structure. There has been plenty of theoretical work such as (Grosz and Sidner, 1986), (Moser and Moore, 1996) which shows that just as sentences can be decomposed into smaller constituents, a discourse can be decomposed into smaller units called discourse segments. Though there are many different ways to segment discourse, the common themes are that some sequences are more closely related than others (discourse segments) and that a discourse can be organized as a tree, with the leaves being the individual utterances and the interior nodes being discourse segments. The embeddedness of a segment effects which previous segments, and thus their entities, are accessible. As a discourse progresses, segments close and unless they are close to the root of the tree (have a low embedding) may not be accessible.

Discourse segmentation has implications for spoken dialogue systems. Properly detecting discourse structure can lead to improved reference resolution accuracy since competing antecedents in inaccessible clauses may be removed from consideration. Discourse segmentation is often closely related to plan and intention recognition, so recognizing one can lead to better detection of the other. Finally, segmentation reduces the size of the history or context maintained by a spoken dialogue system, thus decreasing the search space for referents.

3 Monroe Corpus Construction

The Monroe domain is a series of task-oriented dialogs between human participants (Stent, 2001) designed to encourage collaborative problem-solving and mixed-initiative interaction. It is a simulated rescue operation domain in which a controller receives emergency calls and is assisted by a system or another person in formulating a plan to handle emergencies ranging from requests for medical assistance to civil disorder to snow storms. Available resources include maps, repair crews, plows, ambulances, helicopters and police.

Each dialog consisted of the execution of one task which lasted about ten minutes. The two participants were told to construct a plan as if they were in an emergency control center. Each session was recorded to audio and video, then broken up into utterances under the guidelines of (Heeman and Allen, 1994). Finally, the segmented audio files were transcribed by hand. The entire Monroe corpus consists of 20 dialogs. The annotation work we report here is based on 5 dialogs totaling 1756 utterances [1].

Discourse annotation of the Monroe Corpus consisted of three phases: first, a semi-automated annotation loop that resulted in parser-generated syntactic and semantic analyses for each sentence. Second, the corpus was manually annotated for reference information for pronouns and coreferential information for definite noun phrases. Finally, discourse segmentation was conducted manually. In the following sections we discuss each of the three phases in more detail.

3.1 Building the Parsed Corpus

To build the annotated corpus, we needed to first have a parsed corpus as a source of discourse entities. We built a suite of tools to rapidly develop parsed corpora (Swift et al., 2004). These are Java GUI for annotating speech repairs, a LISP tool to parse annotated corpora and merge in changes, and a Java tool interface to manually check the automatically generated parser analyses (the CorpusTool).

Our goal in building the parsed corpus is to obtain the output suitable for further annotation for reference and discourse information. In particular, the parser achieves the following:

- Identifies the referring expressions. These are definite noun phrases, but also verb phrases and propositions which can be referred to by deictic pronouns such as *that*. All entities are assigned a unique variable name which can be used to identify the referent later.
- Identifies implicit entities. These are implicit subjects of imperatives, and also some implicit arguments of relational nouns (*e.g.*, the implied object in the phrase *the weight*) and of adverbials (*e.g.*, the implied reference time in *That happened before*).
- Identifies speech acts. These are based on the syntactic form of the utterance only, but they provide an initial analysis which can later be extended in annotation.

Examples of the logical form representation for the sentence *So the heart attack person can't go*

[1] The 5 Monroe dialogs are: s2, s4, s12, s16, s17

```
(TERM :VAR V3283471
      :LF (LF::THE V3283471 (:* LF::PERSON PERSON) :ASSOC-WITH (V3283440))
      :SEM ($ F::PHYS-OBJ (F::SPATIAL-ABSTRACTION F::SPATIAL-POINT)
           (F::GROUP -) (F::MOBILITY F::NON-SELF-MOVING)
           (F::FORM F::SOLID-OBJECT) (F::ORIGIN F::HUMAN)
           (F::OBJECT-FUNCTION F::OCCUPATION) (F::INTENTIONAL +)
           (F::INFORMATION -) (F::CONTAINER -) (F::KR-TYPE KR::PERSON)
           (F::TRAJECTORY -))
      :INPUT (THE HEART ATTACK PERSON))
```

Figure 1: Excerpt from full logical form for dialog s2 utterance 173

```
(UTT :TYPE UTT :SPEAKER :USER :ROOT V3286907
     :TERMS
     ((LF::SPEECHACT V3286907 SA_TELL :CONTENT V3283686 :MODS (V3283247))
     (LF::F V3283247 (:* LF::CONJUNCT SO) :OF V3286907)
     (LF::F V3283686 (:* LF::MOVE GO) :THEME V3283471 :MODS (V3284278)
     :TMA ((TENSE PRES) (MODALITY (:* LF::ABILITY CAN)) (NEGATION +)))
     (LF::THE V3283471 (:* LF::PERSON PERSON) :ASSOC-WITH (V3283440))
     (LF::KIND V3283440 (:* LF::MEDICAL-CONDITION HEART-ATTACK))
     (LF::F V3284278 (:* LF::TO-LOC THERE) :OF V3283686 :VAL V3286383)
     (LF::IMPRO V3286383 (OR LF::PHYS-OBJECT LF::REFERENTIAL-SEM)
     :CONTEXT-REL THERE))
```

Figure 2: Abbreviated LF representation for *So the heart attack person can't go there*

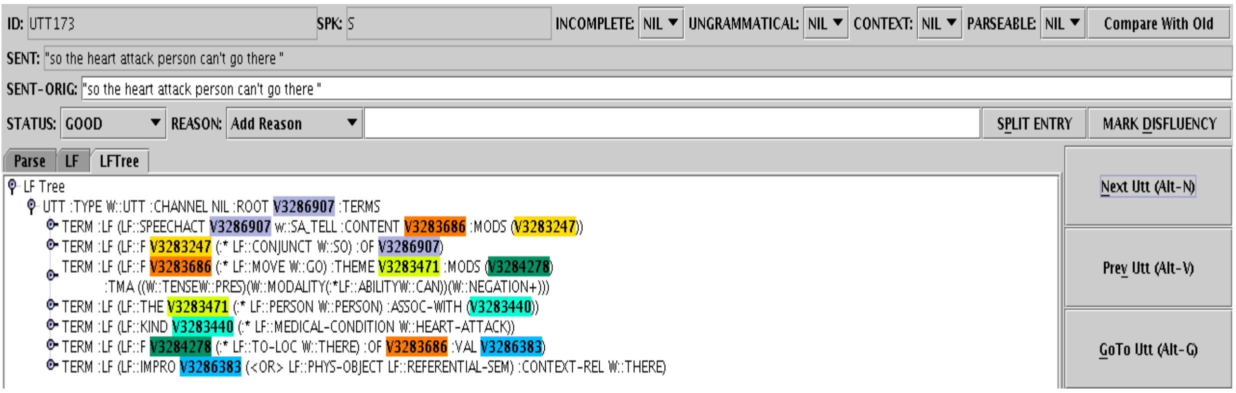

Figure 3: CorpusTool Abbreviated LF View

there (dialog s2, utterance 173) is shown in Figures 1 and 2. Figure 1 shows the full term for the noun phrase *the heart attack person*. It contains the term identifier `:VAR V3283471`, the logical form (`:LF`), the set of semantic features associated with the term (`:SEM`), and the list of words associated with the term (`:INPUT`). The semantic features are the domain-independent semantic properties of words encoded in our lexicon. We use them to express selectional restrictions (Dzikovska, 2004) and we are currently investigating their use in reference resolution. For discourse annotation, we primarily rely on the logical forms.

The abbreviated logical form for the sentence is shown in Figure 2. It contains the speech act for the utterance, `SA_TELL`, in the first term. There is a domain-independent term for the discourse adverbial *So*[2], and the term for the main event, `(LF::Move GO)`, which contains the tense and modal information in the `:TMA` field. The phrase *the heart attack person* is represented by two terms linked together with the `:ASSOC-WITH` relationship, to be resolved during discourse processing. Finally, there is a term for the adverbial modifier *there*, which also results in the implicit pronoun (the

[2] *So* is identified as a conjunct because it is a connective, and its meaning cannot be identified more specifically by the parser without pragmatic reasoning

last term in the representation) denoting a place to which the movement is directed. The terms provide the basic building blocks to be used in the discourse annotation, and their unique identifiers are used as reference indices, as discussed in the next section.

The corpus-building process consists of three stages: initial annotation, parsing and hand-checking. The initial annotation prepares the sentences as suitable inputs to the TRIPS parser. It is necessary because handling speech repairs and utterance segmentation is a difficult task, which our parser cannot do automatically at this point. Therefore, we start with segmenting the discourse turns into utterances and marking the speech repairs using our tool. We also mark incomplete and ungrammatical utterances which cannot be successfully interpreted.

Once the corpus is annotated for repairs, we use our automated LISP testing tool to parse the entire corpus. Our parser skips over the repairs we marked, and ignores incomplete and ungrammatical utterances. Then, it marks utterances "AUTO-GOOD" and "AUTO-BAD" as a guideline for annotators. As a first approximation, the utterances where there is a parse covering the entire utterance are marked as "AUTO-GOOD" and those where there is not are marked as "AUTO-BAD". Then these results are hand-checked by human annotators using our CorpusTool to inspect the analyses and either mark them as "GOOD", or mark the incorrect parses as "BAD", and add a reason code explaining the problem with the parse. Note that we use a strict criterion for accuracy so only utterances that have both a correct syntactic structure and a correct logical form can be marked as "GOOD". The CorpusTool allows annotators to view the syntactic and semantic representations at different levels of granularity. The top-level LF tree shown in Figure 3 allows a number of crucial aspects of the representation to be checked quickly. Note that the entity identifiers are color-coded, which is a great help for checking variable mappings. If everything shown in the top-level representation is correct, the full LF with all terms expanded can be viewed. Similarly, levels of the parse tree can be hidden or expanded as needed.

After the initial checking stage, we analyze the utterances marked "BAD" and make changes in the grammar and lexicon to address the BAD utterances whenever possible. Occasionally, when the problems are due to ambiguity, the parser is able to parse the utterance, but the interpretation it selects is not the correct one among possible alternatives. In this case, we manually select the correct parse and add it to the gold-standard corpus.

Once the changes have been made, we re-parse the corpus. Our parsing tool determines automatically which parses have been changed and marks them to be re-checked by the human annotators. The CorpusTool has the functionality to quickly locate the utterances marked as changed for re-checking. This allows us to quickly conduct several iterations of re-checking and re-parsing, bringing the coverage in the completed corpus high enough so that it may now be annotated for reference information. The hand-checking scheme was found to be quite reliable, with a kappa of 0.79. Currently, 85% of the grammatical sentences are marked as GOOD in the gold-standard coverage of the 5 dialogs in the Monroe corpus.

Several iterations of the check and re-parse cycle were needed to achieve parsing accuracy suitable for discourse annotation. Once the suitable accuracy level has been reached, the reference annotation process starts.

3.2 Adding Reference Information

As in the parser development phase, we built a Java tool for annotating the parsed corpora for reference. First, the relevant terms were extracted from the LF representation of the semantic parse. These included all verbs, noun phrases, implicit pronouns, etc. Next, the sentences were manually marked for reference using the tool (PronounTool).

There are many different ways to mark how entities refer. Our annotation scheme is based on the GNOME project scheme (Poesio, 2000) which annotates referential links between entities as well as their respective discourse and salience information. The main difference in our approach is that we do not annotate discourse units and certain semantic features, and most of the basic syntactic and semantic features are produced automatically for us in the parsing phase.

We use standoff annotation to separate our coreference annotation from the syntactic and semantic parse annotations. The standoff file for pronouns consists of two fields for each pronoun to handle the reference information: relation, which specifies how the entities are related; and refers-to, which specifies the id of the term the referential entity in question points to.

The focus for our work has been on coreferential pronouns and noun phrases, although we also annotated the classes of all other pronouns. Typically, the non-coreferential pronouns are difficult to annotate reliably since there are a myriad of different categories for bridging relations and for specifying

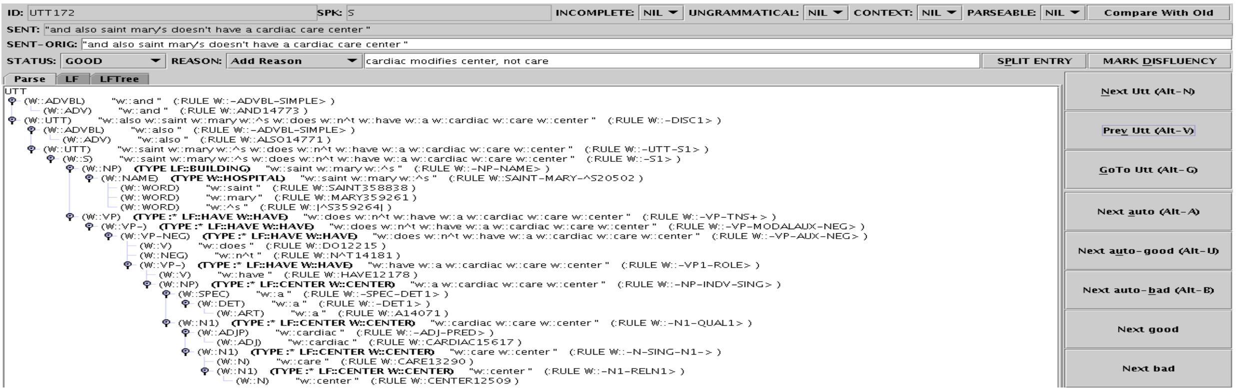

Figure 4: CorpusTool Parse View

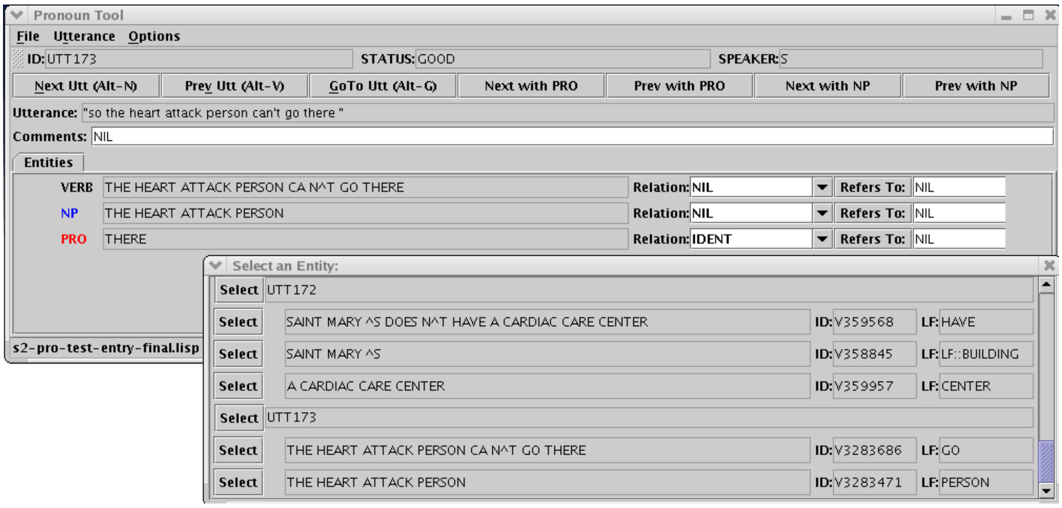

Figure 5: Pronoun Tool

demonstrative relations (Poesio and Viera, 1998). Because our focus was on coreferential entities, we had our annotators annotate only the main relation type for the non-coreferential pronouns since these could be done more reliably. The relations we used are listed below:

Identity both entities refer to the same object (coreference)

Dummy non-referential pronouns (expletive or pleonastic)

Indexicals expressions that refer to the discourse speakers or temporal relations (ie. I, you, us, now)

Action pronouns which refer to an action or event

Demonstrative pronouns that refer to an utterance or series of utterances

Functional pronouns that are indirectly related to another entity, most commonly bridging and one anaphora

Set plural pronouns that refer to a collection of mentioned entities

Hard pronouns that are too difficult to annotate

Entities in identity, action and functional relations had refers-to fields that pointed to the id of a specific term (or terms if the entity was a plural composed of other entities). Dummy had no refers-to set since they were not included in the evaluation. Demonstrative pronouns had refers-to fields pointing to either utterance numbers or a list of utterance numbers in the case of a discourse segment. Finally, there were some pronouns for which it was too difficult to decide what they referred to, if anything. These typically were found in incomplete sentences without a verb to provide semantic information.

After the annotation phase, a post-processing phase identifies all the noun phrases that refer to the same entity, and generates a unique chain-id for this entity. This is similar to the *ante* field in the GNOME scheme. The advantage of doing this processing is that it is possible for a referring expression to refer to a past instantiation that was not the last mentioned instantiation, which is usually what is annotated. As a result, it is necessary to mark all coreferential instantiations with the same identification tag.

Figure 5 shows a snapshot of the PronounTool in use for the pronoun *there* in the second utterance of our example. The top pane has buttons to skip to the next or previous utterance with a pronoun or noun phrase. The lower pane has the list of extracted entities for easy viewing. The "Relation" box is a drop down menu consisting of the relations listed above. In this case, the identity relation has been selected for *there*. The next step is to select an entity from the context that the pronoun refers to. By clicking on the "Refers To" box, a context window pops up with all the entities organized in order of appearance in the discourse. The user selects the entity and clicks "Select" and the antecedent id is added to the refers-to field.

Our aim with this part of the project (still in a preliminary stage) is to investigate whether a shallow discourse segmentation (which is generated automatically) is enough to aid in pronominal reference resolution. Previous work has focused on using complex nested tree structures to model discourse and dialogue. While this method may be the best way to go ultimately, empirical work has shown that it has been difficult to put into practice. There are many different schemes to choose from, for example Rhetorical Structure Theory (Mann and Thompson, 1986) or the stack model (Grosz and Sidner, 1986) and manually annotating with these schemes has variable reliability. Finally, annotating these schemes requires real-world knowledge, reasoning, and knowledge of salience and semantics, all of which make automatic segmentation difficult. However, past studies such as Tetreault and Allen (2003) show that for reference resolution, a highly-structured tree may be too constraining, so a shallower approach may be acceptable for studying the effect of discourse segmentation on resolution.

3.3 Discourse Segmentation

Our preliminary segmentation scheme is as follows. In a collaborative domain, participants work on a task until completion. During the conversation, the participants raise questions, supply answers, give orders or suggestions and acknowledge each other's information and beliefs. In our corpus, these speech acts and discourse cues such as *so* and *then* are tagged automatically for reliable annotation. We use this information to decide when to begin and end a discourse segment.

Roberts (1996) suggests that questions are good indicators of the start of a discourse segment because they open up a topic under discussion. An answer followed by a series of acknowledgments usually signal a segment close. Currently we annotate these segments manually by maintaining a "hold-out" file for each dialog which contains a list of all the segments and their start, end and type information.

UTT1	S	so gabriela
UTT2	U	yes
UTT3	S	at the rochester airport there has been a bomb attack
UTT4	U	oh my goodness
UTT5	S	but it's okay
UTT6	U	where is i
UTT7	U	just a second
UTT8	U	i can't find the rochester airport
UTT9	S	[i] it's
UTT10	U	i think i have a disability with maps
UTT11	U	have i ever told you that before
UTT12	S	it's located on brooks avenue
UTT13	U	oh thank you
UTT14	S	[i] do you see it
UTT15	U	yes

Figure 6: Excerpt from dialog s2

For example, given the discourse as shown in Figure 6, the discourse segments would be Figure 7. The starts of both segments are adjacent to sentences that are questions.

```
(SEGMENT :START utt6
         :END utt13
         :TYPE clarification
         :COMMENTS "has aside in middle")

(SEGMENT :START utt10
         :END utt11
         :TYPE aside
         :COMMENTS "same person aside.")
```

Figure 7: Discourse annotation for s2 excerpt

4 Results

Spoken dialogue is a very difficult domain to work with because utterances are often marred with disfluencies, speech repairs, and are incomplete or ungrammatical. Speakers will interrupt each other. As a result, many empirical methods that work well in very formal, structured domains such as newspaper texts or manuals tend to suffer. For example, many leading pronoun resolution methods perform around 80% accuracy over a corpus of syntactically-parsed Wall Street Journal articles (e.g., (Tetreault, 2001) and (Ge et al., 1998)), but in spoken dialogue the performance of these algorithms drops significantly (Byron, 2002).

However, by including semantic and discourse information, one is able to improve performance. Our preliminary results show that using the semantic feature lists associated with each entity as a filter for reference increases performance to 59% from 44%. Adding discourse segmentation boosts that figure to 66% over some parts of the corpus.

5 Conclusion

We have presented a description of our corpus annotation in the Monroe domain. It is novel in that it incorporates rich semantic information with reference and discourse information, a rarity for spoken dialogue domains which are typically very difficult to annotate. We expedite the annotation process and make it more reliable by semi-automating the parsing with checking and also by using two tools tailored for our domain to speed up annotation. The resulting corpus has several applications ranging from overall system development to the testing of theories and algorithms of reference and discourse. Our preliminary results demonstrate the usefulness of the corpus.

6 Acknowledgments

Partial support for this project was provided by ONR grant no. N00014-01-1-1015, "Portable Dialog Interfaces" and NSF grant 0328810 "Continuous Understanding".

References

T. Brants and O. Plaehn. 2000. Interactive corpus annotation. In *LREC '00*.

D. Byron. 2002. Resolving pronominal reference to abstract entities. In *ACL '02*, pages 80–87, Philadelphia, USA.

M. O. Dzikovska, M. D. Swift, and J. F. Allen. 2004. Building a computational lexicon and ontology with framenet. In *LREC workshop on Building Lexical Resources from Semantically Annotated Corpora*. Lisbon, Portugal, May.

M. Dzikovska. 2004. *A Practical Semantic Representation for Natural Language Parsing*. Ph.D. thesis, U. Rochester.

N. Ge, J. Hale, and E. Charniak. 1998. A statistical approach to anaphora resolution. *Proceedings of the Sixth Workshop on Very Large Corpora*.

B. Grosz and C. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

P. Heeman and J. Allen. 1994. The TRAINS93 dialogues. Technical Report TRAINS TN 94-2, U. Rochester.

W. Mann and S. Thompson. 1986. Rhetorical structure theory: Descripton and construction of text. Technical Report ISI/RS-86-174, USC/Information Sciences Institute, October.

M. P. Marcus, B Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.

M. Moser and J.D. Moore. 1996. Toward a synthesis of two accounts of discourse structure. *Computational Linguistics*, 22(3):409–419.

M. Poesio and R. Viera. 1998. A corpus-based investigation of definite description use. *Computational Linguistics*, 24(2):183–216.

M. Poesio. 2000. Annotating a corpus to develop and evaluate discourse entity realization algorithms: issues and preliminary results. In *LREC '00*, Athens.

C. Roberts. 1996. Information structure in discourse. *Papers in Semantics*, 49:43–70. Ohio State Working Papers in Linguistics.

A. Stent. 2001. *Dialogue Systems as Conversational Partners*. Ph.D. thesis, U. Rochester.

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

J. Tetreault and J. F. Allen. 2003. An empirical evaluation of pronoun resolution and clausal structure. In *2003 International Symposium on Reference Resolution and its Applications to Question Answering and Summarization*, pages 1–8, Venice, Italy.

J. Tetreault. 2001. A corpus-based evaluation of centering and pronoun resolution. *Computational Linguistics*, 27(4):507–520.