Annotation for and Robust Parsing of Discourse Structure on Unrestricted Texts

Jason Baldridge, Nicholas Asher, and Julie Hunter

The University of Texas at Austin
1 University Station B5100
Austin, TX 78712-0198 USA
{jbaldrid,nasher,jhunter}@mail.utexas.edu

Abstract Predicting discourse structure on naturally occurring texts and dialogs is challenging and computationally intensive. Attempts to construct hand-built systems have run into problems both in how to specify the required knowledge and how to perform the necessary computations in an efficient manner. Data-driven approaches have recently shown to be successful for handling challenging aspects of discourse without using lots of fine-grained semantic detail, but they require annotated material for training. We describe our effort to annotate Segmented Discourse Representation Structures on Wall Street Journal texts, arguing that graph-based representations are necessary for adequately capturing the dependencies found in the data. We then explore two data-driven parsing strategies for recovering discourse structures. We show that the generative PCFG model of B&L is inherently limited by its inability to incorporate new features when learning from small data sets, and we show how recent developments in dependency parsing and discriminative learning can be utilized to get around this problem and thereby improve parsing accuracy. Results from exploratory experiments on Verbmobil dialogs and our annotated news wire texts are given; these results suggest that these methods do indeed enhance performance and have the potential for significant further improvements by developing richer feature sets.

Keywords: *discourse structure, SDRT, probabilistic parsing, Verbmobil, rhetorical relations, dependency grammar*

1 Introduction

Specifying discourse structure is a challenging and interesting problem for computational treatments. While that fact alone makes it a problem worth pursuing for some, discourse parsing has many theoretical motivations and practical benefits. Discourse structure has been shown to play a role in determining the content conveyed by a text in many ways; it affects the interpretation of anaphoric expressions, the temporal and spatio-temporal structure of the text, presupposition, the resolution of lexical and other ambiguities (e.g., verb phrase ellipsis, quantification scope), and the calculation of implicatures and conversational goals of agents in dialog (Asher and Lascarides, 2003). Discourse parsing thus is essential to recovering the conveyed content of a text. It also has uses in many practical applications, e.g. generation (Hovy, 1993), text summarization (Marcu, 1997), and automated essay evaluation (Higgins et al., 2004).

Another, perhaps less obvious, reason to focus computational attention on discourse structure is to provide a means of empirical verification of explicit claims made by theories of discourse. Segmented Discourse Representation Theory (SDRT, Asher and Lascarides (2003)) posits a constraint, called the *right frontier constraint*, on the set of potential antecedents available for an anaphoric expression. While there is a great deal of theoretical evidence that this constraint holds for constructed examples, it has yet to be adequately tested on corpus data.

An automated method for building discourse structures would make it possible to quickly test hypotheses about discourse structure such as the right frontier constraint.

We could also hand-annotate a corpus of discourse structures and then examine whether the discourse constraints for coreference postulated by a theory of discourse such as SDRT are verified by the corpus. Asher (2006) adopted a manual approach to testing the right frontier constraint and found that violations were extremely rare: around 2% of a small sample of the corpus. However, this approach was very time consuming and prone to error; automating the process would be preferable. A parser that can construct discourse structures, paired with a coreference resolver would be one strategy.

Our goal is to create a discourse parser and anaphora resolution system to test discourse theories empirically. Our strategy requires the construction of a corpus annotated with discourse structure and coreference information. So far, we have annotated the MUC6[1] corpus for discourse structure and are in the process of annotating the ACE2[2] corpus; both are already annotated for coreference with respect to individual entities evoked in the text. One of our aims is to investigate whether using the right frontier constraint helps identify coreference links more accurately. Coreference systems generally use simple features such as string similarity and distance (Soon et al., 2001; Ng and Cardie, 2002; Morton, 2000; Kehler et al., 2004; Yang et al., 2003; Ng, 2005; Denis and Baldridge, 2007a) (see Mitkov (2002) for an overview). Richer features are difficult to incorporate, but are likely to be necessary to produce more accurate systems than the current state-of-the art. SDRT's right-frontier constraint has the potential to reduce the number of antecedents available for coreference. On the other hand, coreference can disambiguate the choice of rhetorical relation, as Asher and Lascarides (2003) show on constructed examples Thus, another goal is to see whether the coreference resolution can aid discourse parsing.

In this paper, we discuss some of the challenges presented by discourse parsing and how developments in machine learning and its application to NLP may enable progress for this task. We restrict our discussion primarily to the problem of computing discourse structure. For details of our anaphora resolution system, see Denis and Baldridge (2007a,b). We will first present some background relevant to the problem of computing discourse structure and its effects, and then review previous work on computing discourse structure in a limited domain. We present the generative parsing model of Baldridge and Lascarides (2005b) for parsing appointment scheduling dialogs and discusses its strengths and weaknesses for the task. We then move on to a discussion of discourse parsing on open domain texts, detailing our experiences with discourse structure annotation as well as the results of our computational experiments with a discriminative, dependency grammar approach for building discourse structures.

[1]The Message Understanding Conference, `http://www-nlpir.nist.gov/related_projects/muc/`.
[2]The Automated Content Extraction program, `http://www.nist.gov/speech/tests/ace/`.

2 Background

People have analyzed discourse in a computational setting on many different levels, and in many different ways. Some work in discourse makes little or no appeal to explicit discourse structure. Instead, these theories of discourse ground their approach in something else: speech acts, which are properties of individual utterances (Stolcke et al., 2000); plans and intentions (Grosz and Sidner, 1986; Traum, 1997); discourse modes (e.g. Narrative, Argumentative) (Smith, 2003); or the algorithmic characterization of salience, as in centering theory (Grosz et al., 1995).

Other work considers the connection *between* utterances and segments of utterances. For example, the Penn Discourse Treebank (Miltsakaki et al., 2004) singles out discourse cue phrases in a text and specifies their arguments. This annotation process establishes a set of relations between utterances, while leaving their precise interpretation open. SDRT (Asher, 1993; Asher and Lascarides, 2003), and Rhetorical Structure Theory (RST, Mann and Thompson (1987)), also consider connections between segments, but postulate abstract, interpreted relations which connect (possibly) complex segments of utterances. SDRT furthermore defines the logical consequences of such relations as part of conveyed content and focuses on the effects of these relations and the discourse structure built from them with respect to a variety of areas in semantics and pragmatics. SDRT also provides a logic combining various information sources for building a rich logical form of a discourse. Many of these theories agree that discourse structure of the sort that specifies the relations between elementary discourse units in a text is based on rather rich information sources, including semantics, of the sort that SDRT explores in detail.

These approaches often have complementary goals and should not necessarily be seen as being in opposition to each other. They can all be viewed as legitimate facets of a very rich and complex domain of study and they may all prove useful in computational implementations that attempt to exploit non-trivial discourse-level information. However, the various approaches to discourse structure suggest representations with different computational paradigms. As an example, systems which label utterances with discourse tags are straightforward to implement; it is a labeling task that can be mostly determined by the properties of the utterance itself. Richer representations, however, bring a consequent increase in complexity for machine learning models, which we can roughly characterize with the following three levels: label classification, sequential classification, and structured classification. Examples of each include text classification, part-of-speech tagging, and syntactic parsing, respectively.

Discourse structure computation in both RST and SDRT is a structured classification task. Both theories posit representations with a hierarchical structure to discourse. Implementation of discourse structure is more complicated for SDRT than for RST because the former uses full, acyclic graphs, whereas the latter uses only binary tree structures. In SDRT's representations,

nodes, which represent discourse constituents, may have multiple parents, and there may be multiple arcs between nodes. SDRT also allows for crossing dependencies. There is evidence both within the SDRT framework and without that tree structures alone are not sufficient to adequately represent discourse structure (Wolf and Gibson, 2005). Below we will present data from our corpus concerning verbs of saying that we think provide further evidence against the use of binary tree structures. SDRT actually provides a compromise between RST's binary trees and Wolf and Gibson's very general graphs. While it allows crossing dependencies and multiple parents, SDRT's graphs are generally much sparser than those proposed by Wolf and Gibson.

We focus here on building discourse structures according to SDRT. There are a number of reasons why SDRT is an attractive choice for computational implementation. It supports the notion of a hierarchical structure of rhetorical connections in which the relations have model-theoretic interpretations that add content to the logical forms of the individual utterances. It thus augments compositional semantics with information that can only be gained by considering the wider discourse. SDRT is compatible with semantics produced by many computational grammars. SDRT's right frontier constraint, which goes back to Polanyi (1985), places strong constraints on coreference. It is modular in how information is computed and utilized, and it is compatible with many other ways of looking at discourse-level phenomena.

While building discourse structures according to SDRT requires us to deal with structured classification, we can make progress without modeling rich information content, despite SDRT's reliance on it from the formal perspective. We approximate such information with surface level cues that are gleaned from corpus data. We expect that the resulting discourse structures will help us infer rich information content during later processing.

We now turn to some details about SDRT. A Segmented Discourse Representation Structure (SDRS) is a triple $\langle A, \mathcal{F}, Last \rangle$, where:

- A is a set of labels.
- $Last$ is a label in A (intuitively, this is the label of the content of the last clause that was added to the logical form); and
- $\mathcal{F}$ is a function which assigns each member of A a member of a formula of the SDRS language, which includes formulas of some version of dynamic semantics (Discourse Representation Theory, Dynamic Predicate Logic, Update Semantics, Martin Löf Type Theory, among others.)

This notion of discourse structure is very abstract and thus very general. One important distinction for SDRT (and for many other theories of discourse structure) that needs to be added to

understand the notion of a right frontier is the distinction between two types of discourse relation, *subordinating* and *coordinating*. Theory internal tests can be used to determine whether a given discourse relation is subordinating or coordinating (Asher and Vieu, 2005). These tests confirm that the discourse relation of *Narration* is a prime example of a coordinating relation, while the relation of *Elaboration* is a prime example of a subordinating relation.

The difference between coordinating and subordinating relations for defining the right frontier constraints is best understood by moving from the abstract definition of an SDRS to a graphical representation. The following is an algorithm for constructing a graph from an SDRS understood as above:

- Each constituent (or label) is a node.
- Each subordinating relation creates a downward edge.
- Each coordinating relation creates a horizontal edge.

These graphs are useful in that they represent discourse coherence via their connectedness; the degree of connectedness of the graph is one measure of coherence. The graphical representation also imposes some constraints on what sort of SDRSs are possible. For example, no two nodes can be connected by both a subordinating and coordinating relation. On the other hand, several edges (of the same type) are possible between two constituents. Anaphora resolution and SDRS update are dependent on the graph structure. New discourse constituents must be attached to the right frontier of the graph, that is either to LAST or to some node in the path from LAST to the top node of the graph. Similarly, anaphors must find their antecedents in constituents somewhere along the right frontier. Nevertheless, SDRT allows for the construction of a complex constituent from new information prior to incorporating this new information into the already extant discourse structure. That is, new information does not have to be added one basic constituent at a time. This provides for crossing dependencies with respect to surface structure. We note that such an approach makes perfect sense from our semantic perspective: SDRSs are *semantic* representations and do not necessarily reflect surface word order or even surface syntactic order.

To get a feel for the structures posited by SDRT and for its semantic implications about conveyed content, consider the temporal consequences of a text. The temporal structure of a discourse is more elaborate than what is suggested by the formal semantic analysis of tenses. There are clearly temporal shifts that show that the treatment of tenses cannot simply rely on the superficial order of the sentences in the text.

(1) a. (π_1) John had a great evening last night.

b. (π_2) He had a great meal.

c. (π_3) He ate salmon.

d. (π_4) He devoured lots of cheese.

e. (π_5) He then won a dancing competition.

(1c-d) provides 'more detail' about the event in (1b), which itself elaborates on (1a). (1e) continues the elaboration of John's evening that (1b) started, forming a *narrative* with it (temporal progression). Clearly, the ordering of events does not follow the order of sentences, but rather obeys the constraints imposed by discourse structure, as shown graphically below. Thus the eventualities that are understood as elaborating on others are temporally subordinate to them, and those events that represent narrative continuity are understood as following each other.

SDRT provides the following discourse structure for (1), which allows us to get a proper treatment of the tenses therein.

(1′) $\langle A, \mathcal{F}, \text{last}\rangle$, where:

- $A = \{\pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6, \pi_7\}$
- $\mathcal{F}(\pi_1) = K_{\pi_1}, \mathcal{F}(\pi_2) = K_{\pi_2}, \mathcal{F}(\pi_3) = K_{\pi_3}, \mathcal{F}(\pi_4) = K_{\pi_4}, \mathcal{F}(\pi_5) = K_{\pi_5}$,
  $\mathcal{F}(\pi_0) = \textit{Elaboration}(\pi_1, \pi_6)$
  $\mathcal{F}(\pi_6) = \textit{Narration}(\pi_2, \pi_5) \wedge \textit{Elaboration}(\pi_2, \pi_7)$
  $\mathcal{F}(\pi_7) = \textit{Narration}(\pi_3, \pi_4)$
- $\text{last} = \pi_5$

Here π_6 and π_7 are discourse constituents created by the process of inferring the discourse structure. See Asher and Lascarides (2003) for details. The corresponding graph of (1′) is given in Figure 1.

RST and SDRT treat (1) in a similar fashion. But SDRT has a much more flexible representational system when it comes to handling complex constituents as arguments to discourse relations. RST adopts the Nuclearity Principle for interpreting trees involving nested subordinating relations. Consider the following example in which (π_2) elaborates on (π_1) and (π_3) elaborates on (π_2).

(2) a. (π_1) John quit school.

b. (π_2) He was having academic troubles.

c. (π_3) He was failing all his classes.

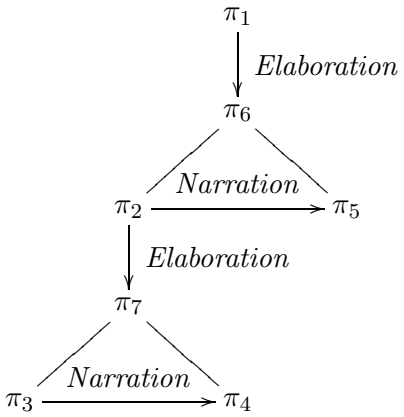


Figure 1: SDRT graph for (1)

RST's tree structure produces the tree in Figure 2. The Nuclearity Principle says that (π_2) is the second argument of the topmost *Elaboration*, which is what is intuitively desired in this case.

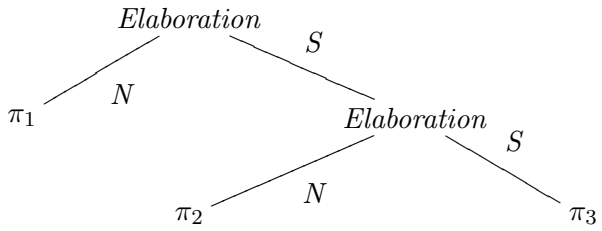


Figure 2: RST tree for (2). N indicates a *nucleus* and S a *satellite*.

However, the Nuclearity Principle runs into a problem with many texts in the MUC6 and ACE2 corpora, in particular with clauses that attribute a content to an agent. Such clauses typically contain reportative verbs like *say*, *affirm*, *assert*, *acknowledge*., *maintain*, and the like. Hunter et al. (2006) distinguish two sorts of discourse relations involving these attributive verbs—*Source* and *Attribution*. *Source* follows the approach of Carlson et al. (2001) to reportative verbs (they use *Attribution* for this relation) in which the material attributed to the agent is the nucleus while the fact that the agent reported this content is a satellite. With this structure in mind, let's consider the following example from our MUC corpus:

(3) a. (π_1) Wall Street traders said

b. (π_2) Piedmont shares fell

c. (π_3) partly because of market uncertainty about federal regulatory approval for a merger with USAir.

(π_2, π_3) should together form the constituent that is the argument to the *Source* (or *Attribution*) relation whose first argument is π_1. But in the RST version of (Carlson et al., 2001) this is not possible, because the Nuclearity Principle dictates that only π_2 is the argument to the relation introduced by reportative verb, as can be seen for the RST tree for (3):

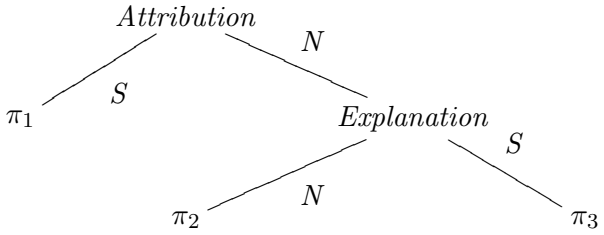


Figure 3: RST tree for (3)

In SDRT, (3) poses no problem. Because SDRT allows the construction of a complex constituent prior to attachment, the abstract SDRS is easily constructed.

- $A = \{\pi_0, \pi_1, \pi_2, \pi_3, \pi\}$
- $\mathcal{F}(\pi_1) =$ Wall street traders said p
  $\mathcal{F}(\pi_2) =$ Piedmont shares fell
  $\mathcal{F}(\pi_3) =$ market uncertainty about fed..
  $\mathcal{F}(\pi_0) =$ *Source*(π, π_1)
  $\mathcal{F}(\pi) =$ *Explanation*(π_2, π_3)
- $Last = \pi_3$

The graph for (3) is (graphically) more difficult to construct. We have a complex constituent π containing two constituents π_2 and π_3 linked by *Elaboration*, while the relation *Source*, which is also subordinating, connects π with π_1. The graph is given in Figure 4.

In our annotation work on the corpus, we have thus found SDRT to have certain advantages over alternative theories like RST. SDRT's graphs are more flexible and expressive than RST trees. Nevertheless, converting the abstract, declarative logical formalism of SDRT into an implementable algorithm is far from trivial. SDRT allows for numerous discourse structures for a

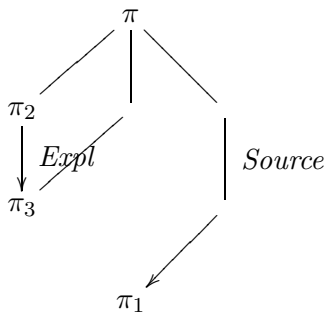


Figure 4: SDRT graph for (3)

given text. So a discourse parser must manage these ambiguities, just as a syntactic parser must manage syntactic ambiguities generated by its grammar. In addition the analytical theory for discourse structure computation is adapted only for toy examples. It cannot be at the present time extended to deal with open domain texts without an army of programmers and linguists. The rules for computing discourse relations require a fine linguistic nose, and the logic used to compute discourse structure in SDRT requires some work to scale up well (although see Schlangen and Lascarides (2002a,b) for some more optimistic appraisals on this score). We have thus decided to pursue an approximation of the analytic theory using machine learning methods. We review below previous work in this area before moving on to the approach we have adopted for open domain text.

3 Discourse Parsing for Dialog

Work on syntactic parsing has progressed at a tremendous rate over the last two decades. We have gone from having grammars built for small domains to having wide-coverage probabilistic grammars that are induced from syntactically annotated treebanks (Collins, 1997; Charniak, 2000; Collins, 2003). The modeling techniques which underlie the grammars have become increasingly sophisticated (Clark and Curran, 2004; Charniak and Johnson, 2005), and the representations produced by these parser-grammars have become richer (Bos et al., 2004). Within computational linguistics, there is also a fair amount of consensus on what are acceptable metrics for evaluating parsing systems. Equally important, there is a general sense that such parsing systems are in fact useful components in the overall project of building natural language understanding systems (e.g, in the Shalmaneser system (Erk and Pado, 2006)).

149 PAM: *maybe we can get together, and, discuss, the planning, say, two hours, in the next, couple weeks,*
150 PAM: *let me know what your schedule is like.*
151 CAE: *okay, let me see.*
152 CAE: *twenty,*
153 CAE: *actually, July twenty sixth and twenty seventh looks good,*
154 CAE: *the twenty sixth afternoon,*
155 CAE: *or the twenty seventh, before three p.m., geez.*
156 CAE: *I am out of town the thirtieth through the,*
157 CAE: *the third, I am in San Francisco.*

Figure 5: A dialog extract from Redwoods.

It actually is not very surprising that less progress has been made at the discourse level. First, theoretical work on syntax is more mature than the theoretical work on discourse structure. Furthermore, the problem is much more complex, involving a much richer range of informational sources than syntax. In part because of the youth of the formal work in this area, there is also perhaps less agreement than in syntax as to just what the task should encompass. Early work by Marcu led to wide-coverage, but hand-built, systems for creating structures according to RST (Marcu, 1997). Later systems used decision trees to determine shift-reduce parsing actions (Marcu, 1999) and a conditional probability model (Soricut and Marcu, 2003), though the latter was only applied to the task of creating discourse structures for individual sentences. More recently, Bangalore et al. (2006) used an incremental, probabilistic parser to structure dialogs, but found that it did not produce better results than using simpler segmentation information.

Baldridge and Lascarides (2005b) use head-driven generative parsing strategies from sentential parsing (e.g., Collins (2003)) to build SDRSs for the Verbmobil appointment scheduling and travel planning dialogs that make up a large part of the Redwoods Treebank (Oepen et al., 2002). An example dialog is that given in Figure 5.

The Redwoods Treebank provides labeled training material for parse selection mechanisms for HPSG grammars (Toutanova et al., 2004; Baldridge and Osborne, 2004). Whereas the Penn Treebank has an implicit grammar underlying its trees, the English Redwoods Treebank uses an explicit manually written grammar, the *English Resource Grammar* (ERG, Copestake and Flickinger (2000)). For each in-coverage sentence, Redwoods enumerates the set of ERG analyses, together with a hand-picked member from this set which is identified as the contextually correct one. Using Redwoods, the ERG and a capable parser, different views of analyses can be recovered: phrase structures, (underspecified) logical forms and elementary dependencies.

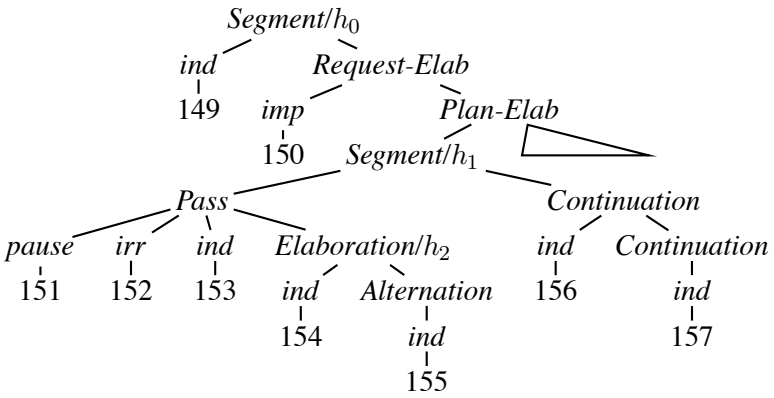


Figure 6: An SDRS tree analysis for the sub-dialog given in Figure 5.

Discourse-level information is crucial for interpreting these dialogs, e.g. for resolving the anaphoric temporal description in utterance 154 of Figure 5 to the twenty sixth of *July* in the afternoon, identifying that time and before 3pm on the twenty-seventh as potential times to meet, while ruling out July thirtieth to August third. Baldridge and Lascarides' objective was to build discourse structures that could eventually aid in the computation of speech act related goals and resolution of bridging relations to enable more accurate and robust determination of such resolutions. Their strategy involved initial annotation of a set of seed dialogs using a tree-based representation, alongside parallel development of a discourse parser. The intention was that this parser could eventually be used to speed up the annotation task. A key aspect of the discourse parser is that it should not require a lot of deep processing: it should not have to rely on rich knowledge about the domain, and should be robust enough able to handle incomplete or ungrammatical utterances like 152 and recognize that utterances 151 and 152 have no overall effect on the time and place to meet.

An example SDRS as a tree is given in Figure 6. Discourse relations appear as the constituent labels rather than arc labels (as in traditional SDRS graph representations). The SDRS formulas can be recovered by the tree traversal method given by Baldridge and Lascarides (2005a). While these trees are not capable of representing all discourse structures, they are sufficiently expressive for the majority of those needed for the Verbmobil domain. They also simultaneously represent discourse segmentation and rhetorical relations.

Using such trees, 100 dialogs were annotated and reviewed to create a gold standard. On average, each dialog contains 9.3 turns, 30.4 utterances, and 234.0 words. There are 34 non-terminal symbols, 31 of which are rhetorical relations (e.g. *Elaboration*, *Continue*) and the

Syntax	Discourse
phrase structure trees	SDRSs as trees
part-of-speech tags	sentence moods
words	utterances
heads	the unique *ind, int, imp, Pass*, or *Segment* child in each sub-tree
dependency relations	rhetorical relations

Figure 7: Mapping syntax to discourse for features in the discourse parser.

other three of which are structural relations (*Segment*, *Pass*, *Top*). Additionally, six terminal symbols consisting of three sentence moods (*Indicative*, *Interrogative*, and *Imperative*) and three ignorable types (*Irrelevant*, *Pause*, and *Pleasantry*) are utilized.

The major advantage of utilizing a tree-based representation is that standard statistical parsing strategies could be applied straightforwardly. There are many well-tested, effective probabilistic parsing methodologies for phrase structure grammars (Collins, 2003; Charniak, 1997). Creating a discourse parser based on these methods essentially reduces to defining several mappings, outlined in Figure 7.

The notion of "head" in discourse trees may seem odd. The main utility of such a concept is that it projects information from lower in a tree to interact probabilistically with elements higher up; for example, in syntax, the decision regarding whether to attach a subject is conditioned by the head verb, even though it is not in the immediate structural context of the S node dominating the attachment. A finding from syntactic parsing is that, even though the notion of a syntactic head is linguistically very well-motivated, the choice of what is the head (e.g. determiners versus nouns heading noun phrases) is actually not that important in terms of improving parser accuracy (Bikel, 2004). The key is percolation of lexical information; for discourse, it means percolation of discourse cue phrases and sentence moods associated with the heads.

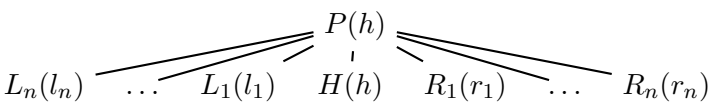
There is a significant body of work on probabilistic parsing, especially that dealing with the English sentences found in the annotated Penn Treebank. One of the most important developments in this work is that of Collins (2003). Collins created several lexicalized head-driven generative parsing models that incorporate varying levels of structural information, such as distance features, the complement/adjunct distinction, subcategorization and gaps. These models are attractive for constructing discourse trees, which contain heads that establish non-local dependencies in a manner similar to that in syntactic parsing. Also, the co-dependent tasks of determining segmentation and choosing the rhetorical connections are both heavily influenced by the content of the utterances/segments which are being considered, and lexicalization allows

the model to probabilistically relate such utterances/segments very directly.

Probabilistic Context Free Grammars (PCFGs) determine the conditional probability of a right-hand side of a rule given the left-hand side, $\mathcal{P}(RHS|LHS)$ (Manning and Schütze, 1999). The probability assigned to an entire tree is simply the multiplication of the probabilities of the rules used in each individual expansion. Collins instead decomposes the calculation of such probabilities by first generating a head and then generating its left and right modifiers independently. In a supervised setting, doing this gathers a much larger set of rules from a set of labeled data than a standard PCFG, which learns only rules that are directly observed. Given that many sequences of coherent discourse moves will be unobserved in a training corpus, however large, Collins' model appears more suitable for adaptation to discourse parsing than a simple PCFG.

The decomposition of a rule begins by noting that rules in a lexicalized PCFG have the form:

(4)



where h is the head word, $H(h)$ is the label of the head constituent, $P(h)$ is its parent, and $L_i(l_i)$ and $R_i(r_i)$ are the n left and m right modifiers, respectively. It is also necessary to include $STOP$ symbols L_{n+1} and R_{m+1} on either side to allow the Markov process to properly model the sequences of modifiers. By assuming these modifiers are generated independently of each other but are dependent on the head and its parent, the probability of such expansions can be calculated as follows (where $\mathcal{P}_h$, $\mathcal{P}_l$ and $\mathcal{P}_r$ are the probabilities for the head, left-modifiers and right-modifiers respectively):

(5) $$\mathcal{P}(L_n(l_n)\ldots L_1(l_1)H(h)R_1(r_1)\ldots R_m(r_m)|P(h)) = \mathcal{P}_h(H|P(h)) \times \prod_{i=1\ldots n+1} \mathcal{P}_l(L_i(l_i)|P(h),H) \times \prod_{i=1\ldots m+1} \mathcal{P}_r(R_i(r_i)|P(h),H)$$

This provides the simplest of models. More conditioning information can of course be added from any structure which has already been generated. For example, Collins' model 1 adds a distance feature that indicates whether the head and modifier it is generating are adjacent and whether a verb is in the string between the head and the modifier.

In such models, probabilities are determined by counting events observed in a training corpus. The reliability of these probabilities are highly dependent on obtaining a sufficient number of such events and the performance of parsers based on them consequently suffers in the face of data sparsity. This is an especially dire problem with small data sets such as the Verbmobil annotations of B&L, which contain just 100 annotated dialogs.

One way to deal with this problem is linear interpolation. If the probability of generating a head is conditioned on the parent P, the head word w, and the head tag t, an interpolated probability estimate $\tilde{\mathcal{P}}_h(H|P,w,t)$ can be calculated as:

Feature	Description
P	Parent label
H	Head label
t	Head utterance sentence mood
w	First discourse cue phrase in utterance
Δ	Modifier adjacency
HCR	Label of head's child relation node
ST	Whether head utterance starts a turn
TC	Number of turn changes in head constituent (0,1, or ≥ 2)
TM	Whether head utterance indicates good or bad times, both, or neither

Figure 8: Features used by B&L in their generative parsing model.

$$\tilde{\mathcal{P}}_h(H|P,w,t) = \lambda_1 \mathcal{P}_h(H|P,w,t) + (1-\lambda_1)(\lambda_2 \mathcal{P}_h(H|P,t) + (1-\lambda_2)\mathcal{P}_h(H|P)) \quad (6)$$

where the λ_i's ensure that the combination of the three models (each of a different level of specificity) is a well-formed probability distribution. This allows the parser to utilize a term such as $\mathcal{P}_h(H|P)$ when it has not observed any actual events for the terms that are conditioned on more features (and which thus have zero probability).

B&L adapted this parsing paradigm for discourse, using the features summarized in Figure 3. These features are much like those used in syntactic parsing models for capturing local tree configurations and node labels, e.g. parent node labels, head node labels, and distance features. However there are significant differences since the trees encode discourse structures rather than syntactic ones. For example, rather than part-of-speech tags, the sentence moods *imperative*, *indicative* and *interrogative* are used. Lexicalization in syntax involves the words found on the leaves. Since leaves in discourse trees are entire sentences, they use the first discourse cue word in the sentence as an approximation for lexicalization. They also use dialog based features and domain specific features which are irrelevant for our domains.

B&L assumed gold standard segmentation and sentence moods as input to the parser, and also used gold standard (human annotated) values for the domain specific "good-time/bad-time" feature. The goal was to see how the addition of these informative features would improve over a baseline of attaching every utterance to the preceding utterance via the most common discourse relation (*Continuation*). Success was measured in terms of the recovery of the correct spans for discourse segments and the correct labels on those spans. Scores were given as f-scores that combine precision and recall for this task. The best model utilized all of the features, and obtained labeled and unlabeled f-scores of 39.5% and 67.4% respectively. This significantly

beat the baseline's values of 7.4% and 53.3%.

B&L's model and results thus provide evidence that a syntactic parsing paradigm can generalize to discourse structure, and it showed that dialog and domain specific features have large positive impact on performance. That is to say that some of the information that would otherwise play a crucial role in actual theorem-proving methods can be used as soft constraints in a probabilistic model to nudge a discourse parser toward the right analyses. Furthermore, in an informal experiment, using the parser output sped up annotation by a factor of one-and-a-half.

Nonetheless, performance would ideally be higher. Using B&L's best model but computing stop probabilities separately from the generation of other dependents significantly improved performance to 41%/71% labeled/unlabeled f-score. However, that is the current performance cap for this model, despite efforts to add further features, e.g. for temporal indices and speech act related goals[3]. Adding further features actually hurt performance: there simply is not enough data to derive reliable estimates for them in the annotated Verbmobil material.

Data sparsity is a problem for any data-driven approach. However, it is particularly grave for generative models of the kind used by Baldridge and Lascarides. The probability estimate that we would ideally be able to use directly is the following (for left dependents of the head):

(7) $$\tilde{\mathcal{P}}(L|P, H, w, t, \Delta, HCR, ST, TC, TM)$$

Given the values these random variables can take, there are over one million possible instantiations. It is thus necessary to interpolate such terms with less specific ones in order to ensure we do not end up with zero-counts:

(8) $$\begin{aligned}\tilde{\mathcal{P}}(L|P, H, w, t, \Delta, HCR, ST, TC, TM) = \\ \lambda_1 \mathcal{P}(L|P, H, w, t, \Delta, HCR, ST, TC, TM) \\ + \ (1 - \lambda_1)(\lambda_2 \mathcal{P}(L|P, H, t, \Delta, ST) \ + \ (1 - \lambda_2)\mathcal{P}(L|P, H, t))\end{aligned}$$

If we lack values for any of the variables in the most specific term, then the whole term is zero. Thus, with a small corpus, many features will rarely make a difference, since the interpolated model will simple revert to the less specific estimates. How one chooses to break up the full term into less specific ones also leads to rather large effects on the performance of the parser.

Another way to deal with sparse counts is to make independence assumptions so that the probability in (3) is broken up into the product of multiple terms such as $\mathcal{P}(L|P, H, w)$ and $\mathcal{P}(L|HCR, ST)$ for which we can obtain higher frequencies. We tried this, but it performed very poorly – the features are far from independent and cannot be separated in this manner.

[3]This is heretofore unreported work conducted by one of the authors, Jason Baldridge, with Alex Lascarides and Ben Hutchinson.

Such generative parsing models can be highly effective when there is plentiful data, but they suffer heavily in the face of little data. We found that progress simply leveled off, and that adding features actually hurt performance. Because of this, we have chosen to explore a discriminative parsing model that allows many rich non-independent features to be added straightforwardly.

4 Annotation of Newswire Texts

In this section we detail our experiences concerning annotation and discourse parsing over open domain texts in the MUC6 and ACE2 corpora. As we mentioned earlier, an implementation of the extant SDRT glue logic for building discourse structures is insufficient to deal with open domain text, and we cannot envision an extended version at the present time able to deal with the problem. Thus, we have opted for a machine learning based approach to discourse structure computation based on superficial features, like B&L. To build an implementation to test these ideas, we have had to devise a corpus of texts annotated for discourse structure in SDRT.

Each of the 60 texts in the MUC6 corpus, and now 18 of the news stories in ACE2, were annotated with a discourse structure by two people familiar with SDRT. The annotators then conferred and agreed upon a gold standard. Our annotation effort took the hierarchical structure of SDRT seriously and built graphs in which the nodes are discourse units and the arcs represent discourse relations between the units. The units could either be elementary discourse units (EDUs) or complex ones. As is standard for annotation work with discourse structure, we first must identify EDUs, which are conveyed by clauses or by appositive elements or non-restrictive relative clauses in a sentence. They are the words of discourse structure or the domain of comparison in centering theory. Even EDU's can be difficult to identify accurately (Soricut and Marcu, 2003).

We chose a small set of SDRT's relations based on an examination of the texts in our corpus. Several of these are subordinating discourse relations: *Elaboration*, *Background*, *Explanation*, *Source*, *Attribution*, *Result* and *Commentary*. The coordinating discourse relations are *Continuation*, *Contrast*, *Result*, *Narration*, *Alternation*, *Parallel*, *Consequence* and *Precondition*. We have provided intuitive definitions of each one of these relations and have provided surface indicators of when they occur in our annotation manual (Reese et al., 2007).

We assumed that in principle the units were recursively generated and could have an arbitrary though finite degree of complexity. We made explicit, however, only those complex segments in which constituents were linked by coordinating discourse relations. Consider the following fragment of a text in the ACE2 corpus:

(9) 84. While the notion of the government putting itself on the hook for private debtors has been controversial,

85. officials at the Finance Ministry have said

86. it may in fact do so.

87. "If that becomes the only choice,

88. I will consider it,"

89. Kim said.

90. "I think it is necessary to make the rollover,

91. and it is important to make the creditors feel safe."

In this discourse, for instance, an *Explanation* link between the complex constituents [87,88] and [90,91] occurs intuitively within the scope of the *Source* relation whose second argument is 89, yet we did not represent that explicitly (incidentally, this text gives yet another example that poses difficulties for the Nuclearity Principle of interpretation of RST trees). The annotators agreed on the following discourse structure:

(10) a. *Source*(*86*,*85*)

b. *Contrast*(*84*,*86*)

c. *Elaboration*(*[84,86]*,*[87,88]*)

d. *Consequence*(*87*,*88*)

e. *Source*(*[87,88]*,*89*)

f. *Explanation*(*[87,88]*,*[90,91]*)

g. *Continuation*(*90*,*91*)

Notice how complex constituents can serve as arguments to discourse relations.

At first we sought to annotate all the links SDRT would postulate between EDUs, but found that annotators often missed discourse links. We then changed our annotation system to introduce complex nodes. For one thing, this reduced the number of discourse links annotators needed to make explicit. For instance, by writing *Elaboration*(*[84,86]*,*[87,88]*), which in SDRT entails *Elaboration*(*[84,86]*,*87*) and *Elaboration*(*[84,86]*,*88*), annotators were able to express with one link what must be expressed with two links in an annotation scheme in which only EDUs are elements of discourse relations. More importantly, the use of complex nodes allowed annotators to express relations between constituents that could not be expressed by relations

between EDUs. In writing *Elaboration([84,86],[87,88])*, annotators want the contrasting constituents 84 *and* 86 to be elaborated on by the conditional in EDUs 87 and 88, but they did not want [87,88] to elaborate on just one of the constituents. In effect the grouping of EDUs 84 and 86 indicates the presence of a constructed topic, something postulated by SDRT but well beyond the present capabilities of automated construction. The use of complex nodes like [84,86] as the first argument of a relation like *Elaboration* is a way of encoding the structural effect of such topics. The use of complex nodes in the annotation scheme thus allows much more expressivity and comes much closer to the full SDRS representations postulated by the theory.

At the same time this has complicated efforts to score inter annotator agreement. We must not only count correct links but also agreements as to what the complex nodes are. Annotators disagreed relatively often concerning the structure inside complex nodes, so we are working currently to determine how to rework our SDRS specifications in the manual to address this.

5 Discourse parsing as dependency parsing

Both the Verbmobil and MUC/ACE newswire domains exhibit complex segments, as outlined in the previous sections. The PCFG-based model of B&L is able to capture such segmentation, but as we discussed in 3, attempts to extend the model with additional features were unsuccessful. One alternative would be to use a discriminative parser, such as that of Taskar et al. (2004). Their model uses a discriminative max-margin method which can condition on arbitrary features without making independence assumptions. It is thus ideal in that it can produce both segmentation and allow us to use the features we would like to use. Unfortunately, it is very computationally intensive: even for parsing sentences of the Penn Treebank, Taskar et al. (2004) had to limit training and parsing to sentences with 15 words or less. Most of our texts have more than 20 EDUs (our words), with an average of over 33 per document. Furthermore, discourse requires far more features than syntactic parsing; this greatly increases the number of parameters which must be estimated, so their model is unlikely to be feasible in our setting.

An alternative which can take advantage of discriminative methods while remaining reasonably efficient is dependency parsing. Dependency grammars generally eschew complex constituents, such as *VP*'s; instead, words are directly linked to one another. Data-driven dependency parsers do not assume an explicit grammar, and instead build structure based on the empirically determined strength of linkage between parsing units.

In many ways, SDRSs are well-suited for dependency representations. It is not clear that an actual grammar should dictate whether and how EDUs combine (though see Forbes et al. (2003) for an approach which utilizes Tree-Adjoining Grammars anchored by discourse cue phrases). Discourse relations are much like dependency relations, rather than phrase structure labels (as

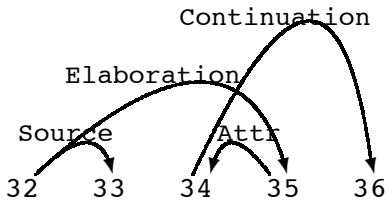


Figure 9: SDRS for example excerpt showing non-projective links. *Attr* stands for *Attribution*.

they are used by B&L). SDRSs also contain crossing (non-projective) dependencies, which can be captured more straightforwardly with dependency relations than with phrase-structure approaches. Finally, despite the presence of complex segments as discussed previously, the structures in the MUC6 annotations are largely formed of relations between two EDUs. This suggests that a dependency parsing approach may be appropriate for generating SDRSs.

Consider the following excerpt from one of the MUC texts:

(11) 32. That meeting made it clear to Mr. Coleman that the Teamsters and Pan Am are still deadlocked,"'

33. one source close to the negotiations said.

34. "As far as we're concerned, the company triggered this,"

35. "Mr. Genoese said."

36. "We're not negotiating any more with Pan Am."

The SDRS for this excerpt, give in Figure 9 in dependency graph format, contains non-projective dependencies which cannot be captured directly by a tree. The discourse structure links 32 to 35 via the *Elaboration* relation, but it also links 34 to 36, leading to crossing dependency arcs. Such configurations are quite common in the corpus, and require parsers that are able to learn from and reproduce non-projective dependencies.

McDonald et al. (2005b) describe a discriminative dependency parsing algorithm that has many desirable properties for dealing with such structures. While many dependency parsers use projective chart-based algorithms, e.g. Eisner's cubic generative algorithm (Eisner, 1996), McDonald et al. (2005b) instead formalize dependency parsing as the problem of finding a maximum spanning tree in a directed graph. They use the Chu-Liu-Edmonds algorithm (Chu

and Liu., 1965; Edmonds, 1967), which handles non-projective dependencies directly and runs in $O(n^2)$ (as opposed to $O(n^3)$ for chart-based approaches). Parameters in the model are set with the Margin Infused Relaxed Algorithm (MIRA) (Crammer and Singer, 2003), a discriminative learning algorithm. The weight of each link is determined as a linear combination of the weights of the features that are active between the two parsing units. Unlike history-based models like PCFGs, it does not condition on previous decisions. The Chu-Liu-Edmonds algorithm is then used to find the highest weighted tree given all the possible links. McDonald et al. show that this algorithm significantly improves performance on dependency parsing for Czech, especially on sentences which contain at least one crossed dependency. We thus expect it to work well for our discourse structures, which contain crossing dependency links like that shown in Figure 9.

The parsing model uses features that incorporate almost all the different ways in which the words and POS tags of a head and dependent (and words/tags in between them) can be related (McDonald et al., 2005a). The basic features are unigram features for the words and part-of-speech for both the parent and the child on their own, and bigram features mixing the words and parts-of-speech of both the parent and the child. Furthermore, there are extended features which encode trigrams of the parts-of-speech of the parent, child and words between them, and there are similar extended features for the words surrounding the parent and child. The model and features are implemented in an open source Java package called MSTParser.[4]

As an initial trial of the suitability of the approach of McDonald et al. (2005a), we tested MSTParser on the Verbmobil annotations. Since it cannot recover the full structures, we investigated how well it could recover the relations between heads of segments rather than between segments. For example, in the tree in Figure 6, B&L evaluated their model on recovery of relations like *Request-Elab*(*150*,*151-157*); restricted to scoring on heads only, the relation which needs to be recovered is *Request-Elab*(*150*,*153*) since 153 is the head of the complex segment containing EDUs 151 through 157. This kind of evaluation is less stringent, but is nonetheless standard for dependency parsing. The baseline of attaching to the previous EDU with *Continuation* is 12.9%/57.2%, versus 7.4%/53.3% with the more strict metric. Rescoring Baldridge and Lascarides' best model on head-head dependencies, the labeled/unlabeled performance is 48.8%/81.2%, versus 41.0%/71.0% on the more strict metric.

We used a very simple feature set: (1) the model had two "words" *begin_turn* and *in_turn* (inspired by Begin-Inside-Outside representations used in chunking and named entity recognition); and (2) it used the same "POS" tags as Baldridge and Lascarides – *Indicative*, *Interrogative*, *Imperative*, *Irrelevant*, *Pause*, and *Pleasantry*. On a ten-fold cross-validation using the Verbmobil dialogs using these features, MSTParser had labeled/unlabeled performance of 53.2%/86.5%, a significant boost over the PCFG-based model. This is particularly impressive given that the

[4] Available from `http://sourceforge.net/projects/mstparser`.

latter model had access to much more information, in the form of the features given in Figure 7.

This trial experiment has encouraged us to pursue the dependency parsing approach further and apply it to the MUC annotations. Because MIRA is discriminative, many more features can be included without running into problems due to independence assumptions or requiring intricate smoothing, unlike B&L. We have thus extended the parser to optionally use a wider range of features. These features are added to the parser as a list of additional observations, which are also used to generate many features that are combinations of these basic observations.

To apply the parser to the MUC annotations, it was necessary to ignore complex segments; for this, we take the first EDU in a complex segment. This is an approximation that is more crude than the head-head evaluation for Verbmobil, where the notion of head was clearly defined. Nonetheless, it provides a sense of the shape of modeling to come. With this approximation, the baseline for the MUC annotations is 10.4%/40.3% labeled/unlabeled accuracy.

We utilized several utterance-level features (observations) in the model:

- as the "words": discourse cue phrases (e.g., *because*, *although*), from the list of Oates (2001); when no cue phrase is present, the first word of the EDU is used instead
- as the POS tags: the EDU final punctuation
- the length of the EDU
- whether the first word is capitalized
- whether the first word is a discourse cue phrase
- whether the utterance starts or ends a paragraph

Using just the first two features, the parser gets 22.2%/42.8% accuracy, barely beating the baseline for unlabeled performance. By adding in the rest of these features, performance boosts to 23.0%/51.9%, significantly beating the baseline.

An interesting point is that these results were obtained with the non-projective Chu-Liu-Edmond algorithm. When using the same features with the projective Eisner algorithm performance was 22.8%/50.8%. The difference of one-percent gained by using the non-projective algorithm is inconclusive, but suggestive that the non-projective algorithm is more adequate. The projective algorithm simply cannot recover some of the dependencies encoded in the data.

These are encouraging initial results, especially given the relatively minimal effort put into obtaining them. The features described above are just a small subset of all which might be available. In future work, we will experiment with richer feature sets on the MUC texts, and also score on actual segmentation rather than the approximation used for these trials.

The major weakness of the approach outlined above is that it does not create the complex segments which occur in the SDRSs. We will consider various ways that we can reap the benefits of the non-projective, discriminative dependency parsing approach while working toward recovering full segmentation. One obvious alternative is to use SDRT principles to identify coherent segmentation patterns that can be recovered from the dependencies produced by the parser. Another would be to perform EDU level chunking to identify the major segments (which are typically fairly flat) and provide those as the input to the parser. The most elegant approach would be to extend the parser to deal with segmentation; however, this is likely to run into similar efficiency issues as faced by the max-margin parsing approach of Taskar et al. (2004).

6 Conclusion

We have given an overview of our ongoing efforts to annotate discourse structures and compute them using machine learning methods. We introduced and argued for SDRT's representations, which are the theoretical basis of our annotations, because of its greater expressive power than, for example, RST. We discussed previous work in data-driven discourse parsing, with a particular focus on the generative model of Baldridge and Lascarides (2005b). Even though this model is keyed into mostly surface-level features and thus avoids complex theorem-proving methods, we showed that data sparsity affects that model in a particularly severe manner. This leads to actual performance dips as additional features are added. Finally, we presented a view of discourse parsing as dependency parsing, which avails us of several recent advancements. In particular, the parser of McDonald et al. (2005a) uses a discriminative method, MIRA, that allows arbitrary features to be used without violating independence assumptions and it natively recovers non-projective dependencies. We gave results for several trial experiments that indicate that this approach will yield better results than the generative PCFG-based approach.

Our future direction for the discourse parser is to extend the features it uses and to recover segmentation as well as EDU-EDU dependencies. In parallel, we have investigated the combination of models for coreference (Denis and Baldridge, 2007b) using Integer Linear Programming (ILP, see Roth and Yih (2004) for an explication of its use for natural language processing tasks). ILP provides a principled way to integrate several different models in a joint, global fashion. This improved results for coreference using fairly standard features. More importantly, ILP uses declarative constraints that dictate how the decisions made by the base models must be coherent with respect to one another. For example, a constraint might state that a text mention that is anaphoric *must* have an antecedent or that an antecedent and anaphor must have the same named entity type. A key advantage of ILP is that each base model is trained and run independently. This should enable us to cleanly combine coreference resolution and discourse parsing.

Acknowledgements

The authors would like to thank Pascal Denis, Alex Lascarides, Brian Reese, and the anonymous reviewer for their suggestions and to Ryan McDonald for making MSTParser and its source code available. This work was supported by NSF grant IIS-0535154.

References

N. Asher. *Reference to Abstract Objects in Discourse*. Kluwer Academic Publishers, 1993.

N. Asher and A. Lascarides. *Logics of Conversation*. Cambridge University Press, Cambridge, UK, 2003.

N. Asher and L. Vieu. Subordinating and coordinating discourse relations. *Lingua*, 115(4): 591–610, 2005. Numero Special sur la semantique et le langage naturel.

Nicholas Asher. Troubles on the right frontier. In *Proceedings of Constraints in Discourse 2005*, 2006.

J. Baldridge and A. Lascarides. Annotating discourse structures for robust semantic interpretation. In *Proceedings of the 6th International Workshop on Computational Semantics*, Tilburg, The Netherlands, 2005a.

J. Baldridge and A. Lascarides. Probabilistic head-driven parsing for discourse structure. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 96–103, Ann Arbor, MI, 2005b.

J. Baldridge and M. Osborne. Active learning and the total cost of annotation. In *Proceedings of Empirical Approaches to Natural Language Processing (EMNLP)*, 2004.

S. Bangalore, G. Di Fabbrizio, and A. Stent. Learning the structure of task-driven human-human dialogs. In *Proceedings of ACL 2006*, 2006.

D. Bikel. Intricacies of Collins' parsing model. *Computational Linguistics*, 30(4):479–511, 2004.

J. Bos, S. Clark, M. Steedman, J. Curran, and J. Hockenmaier. Wide coverage semantic representations from a CCG parser. In *Proceedings of the International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland, 2004.

L. Carlson, D. Marcu, and M. Okurowski. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the 2nd SIGDIAL Workshop on Discourse and Dialogue, Eurospeech*, 2001.

E. Charniak. Statistical parsing with a context-free grammar and word statistics. In *Proc. of the AAAI*, pages 598–603, Providence, RI, 1997. AAAI Press/MIT Press.

E. Charniak. A maximum-entropy-inspired parser. In *Proceedings of the 1st Conference of the North American Chapter of the Assocation for Computational Linguistics*, pages 132–139, Seattle, WA, 2000.

E. Charniak and M. Johnson. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the ACL*, Ann Arbor, Michigan, 2005.

Y.J. Chu and T.H. Liu. On the shortest arborescence of a directed graph. *Science Sinica*, 14: 1396–1400, 1965.

S. Clark and J.R. Curran. Parsing the wsj using ccg and log-linear models. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 104–111, Barcelona, Spain, 2004.

M. Collins. Three generative, lexicalised models for statistical parsing. In *Proc. of the 35th Annual Meeting of the ACL*, pages 16–23, Madrid, Spain, 1997.

M. Collins. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–638, 2003.

A. Copestake and D. Flickinger. An open-source grammar development environment and english grammar using HPSG. In *Proceedings of the Second Conference on Language Resources and Evaluation (*LREC 2000*)*, Athens, 2000.

K. Crammer and Y. Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 2003.

P. Denis and J. Baldridge. A ranking approach to pronoun resolution. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1588–1593, Hyderabad, India, 2007a.

P. Denis and J. Baldridge. Joint determination of anaphoricity and coreference resolution using integer programming. In *Proceedings of the North American Association of Computational Linguistics*, Rochester, NY, 2007b.

J. Edmonds. Optimum branchings. *Journal of Research of the National Bureau of Standards*, 71(B):233–240, 1967.

J. Eisner. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, pages 340–345, Copenhagen, August 1996.

K. Erk and S. Pado. Shalmaneser – a toolchain for shallow semantic parsing. In *Proceedings of LREC-06*, Genoa, 2006.

K. Forbes, E. Miltsakaki, R. Prasad, A. Sarkar, A. Joshi, and B. Webber. D-LTAG system: Discourse parsing with a lexicalized tree-adjoining grammar. *Journal of Logic, Language, and Information*, 12(3), 2003. Special Issue: Discourse and Information Structure.

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

B. Grosz, A. Joshi, and S. Weinstein. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, 21(2):203–226, 1995.

D. Higgins, J. Burstein, D. Marcu, and C. Gentile. Evaluating multiple aspects of coherence in student essays. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*, Boston, MA, 2004.

E. Hovy. Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63(1-2):341–386, 1993.

J. Hunter, N. Asher, B. Reese, and P. Denis. Evidentiality and intensionality: Two uses of reportative constructions in discourse. In Candy Sidner, John Harpur, Anton Benz, and Peter Kühnlein, editors, *Proceedings of the Workshop on Constraints in Discourse*, pages 99–106, National University of Ireland, Maynooth, 2006.

A. Kehler, D. Appelt, L. Taylor, and A. Simma. The (non)utility of predicate-argument frequencies for pronoun interpretation. In *Proceedings of HLT/NAACL*, pages 289–296, 2004.

W. C. Mann and S. A. Thompson. Rhetorical structure theory: A framework for the analysis of texts. *International Pragmatics Association Papers in Pragmatics*, 1:79–105, 1987.

C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

D. Marcu. The rhetorical parsing of unrestricted natural language texts. In *Proceedings of ACL/EACL*, pages 96–103, Somerset, New Jersey, 1997.

D. Marcu. A decision-based approach to rhetorical parsing. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL99)*, pages 365–372, Maryland, 1999.

R. McDonald, K. Crammer, and F. Pereira. Online large-margin training of dependency parsers. In *Proceedings of ACL 2005*, Ann Arbor, MI, USA, 2005a.

R. McDonald, F. Pereira, K. Ribarov, and J. Hajic. Non-projective dependency parsing using spanning tree algorithms. In *HLT-EMNLP 2005*, Vancouver, B.C., 2005b.

E. Miltsakaki, R. Prasad, A. Joshi, and B. Webber. The Penn Discourse TreeBank. In *Proceedings of the Language Resources and Evaluation Conference*, Lisbon, Portugal, 2004.

R. Mitkov. *Anaphora Resolution*. Longman, Harlow, UK, 2002.

T. Morton. Coreference for NLP applications. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL 2000)*, Hong Kong, 2000.

V. Ng. Supervised ranking for pronoun resolution: Some recent improvements. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05)*, 2005.

V. Ng and C. Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 104–111, 2002.

S. Oates. Generating multiple discourse markers in text. Master's thesis, ITRI, University of Brighton, 2001.

S. Oepen, E. Callahan, C. Manning, and K. Toutanova. LinGO Redwoods—a rich and dynamic treebank for HPSG. In *Proceedings of the* LREC *parsing workshop: Beyond* PARSEVAL*, towards improved evaluation measures for parsing systems*, pages 17–22, Las Palmas, 2002.

L. Polanyi. A theory of discourse structure and discourse coherence. In P. D. Kroeber W. H. Eilfort and K. L. Peterson, editors, *Papers from the General Session at the 21st Regional Meeting of the Chicago Linguistics Society*. 1985.

B. Reese, J. Hunter, P. Denis, N. Asher, and J. Baldridge. Reference manual for the analysis and annotation of rhetorical structure. http://comp.ling.utexas.edu/discor/, 2007.

D. Roth and W. Yih. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the 8th Conference on Natural Language Learning*, 2004.

D. Schlangen and A. Lascarides. Resolving fragments using discourse information. In *Proceedings of the 6th International Workshop on the Semantics and Pragmatics of Dialogue (Edilog)*, Edinburgh, 2002a.

D. Schlangen and A. Lascarides. A compositional and constraint-based approach to non-sentential utterances. In *Proceedings of the 10th International Conference on Head-Driven Phrase Structure Grammar*, pages 380–390. CSLI, 2002b.

C. S. Smith. *Modes of Discourse*. Cambridge University Press, 2003.

W. Soon, H. Ng, and D. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001.

R. Soricut and D. Marcu. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of Human Language Technology and North American Association for Computational Linguistics*, Edmonton, Canada, 2003.

A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, D. Jurafsky R. Bates, P. Taylor, R. Martin, C. van Ess-Dykema, and M. Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374, 2000.

B. Taskar, D. Klein, M. Collins, D. Koller, and C. Manning. Max-margin parsing. In *Proceedings of Empirical Methods in Natural Language Processing*, Barcelona, Spain, 2004.

K. Toutanova, P. Markova, and C. Manning. The leaf projection path view of parse trees: Exploring string kernels for HPSG parse selection. In *Proceedings of Empirical Approaches to Natural Language Processing*, 2004.

D. Traum. A reactive-deliberative model of dialogue agency. In J.P. Muller, M. J. Wooldridge, and N. R. Jennings, editors, *Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages*, pages 157–171, Heidelberg, 1997.

F. Wolf and E. Gibson. Representing discourse coherence: A corpus-based study. *Computational Linguistics*, 31(2):249–287, 2005.

X. Yang, G. Zhou, J. Su, and C.L. Tan. Coreference resolution using competitive learning approach. In *Proceedings of the ACL*, pages 176–183, 2003.