A Novel Discriminative Framework for Sentence-Level Discourse Analysis

Shafiq Joty and Giuseppe Carenini and Raymond T. Ng
{rjoty, carenini, rng}@cs.ubc.ca
Department of Computer Science
University of British Columbia
Vancouver, BC, V6T 1Z4, Canada

Abstract

We propose a complete probabilistic discriminative framework for performing sentence-level discourse analysis. Our framework comprises a discourse segmenter, based on a binary classifier, and a discourse parser, which applies an optimal CKY-like parsing algorithm to probabilities inferred from a Dynamic Conditional Random Field. We show on two corpora that our approach outperforms the state-of-the-art, often by a wide margin.

1 Introduction

Automatic discourse analysis has been shown to be critical in several fundamental Natural Language Processing (NLP) tasks including text generation (Prasad et al., 2005), summarization (Marcu, 2000b), sentence compression (Sporleder and Lapata, 2005) and question answering (Verberne et al., 2007). Rhetorical Structure Theory (RST) (Mann and Thompson, 1988), one of the most influential theories of discourse, posits a tree representation of a discourse, known as a Discourse Tree (DT), as exemplified by the sample DT shown in Figure 1. The leaves of a DT correspond to contiguous atomic text spans, also called Elementary Discourse Units (EDUs) (three in the example). The adjacent EDUs are connected by a *rhetorical* relation (e.g., ELABORATION), and the resulting larger text spans are recursively also subject to this relation linking. A span linked by a rhetorical relation can be either a NUCLEUS or a SATELLITE depending on how central the message is to the author. Discourse analysis in RST involves two subtasks: (i) breaking the text into EDUs (known as *discourse segmentation*) and (ii) linking the EDUs into a labeled hierarchical tree structure (known as *discourse parsing*).

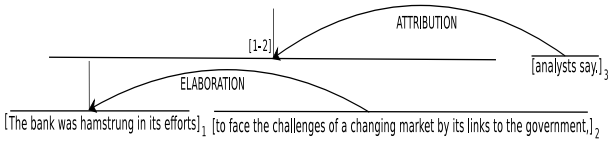

Figure 1: Discourse structure of a sentence in RST-DT.

Previous studies on discourse analysis have been quite successful in identifying what machine learning approaches and what features are more useful for automatic discourse segmentation and parsing (Soricut and Marcu, 2003; Subba and Eugenio, 2009; duVerle and Prendinger, 2009). However, all the proposed solutions suffer from at least one of the following two key limitations: first, they make strong independence assumptions on the structure and the labels of the resulting DT, and typically model the construction of the DT and the labeling of the relations separately; second, they apply a greedy, suboptimal algorithm to build the structure of the DT.

In this paper, we propose a new *sentence-level* discourse parser that addresses both limitations. The crucial component is a probabilistic discriminative parsing model, expressed as a Dynamic Conditional Random Field (DCRF) (Sutton et al., 2007). By representing the *structure* and the *relation* of each discourse tree constituent jointly and by explicitly capturing the *sequential* and *hierarchical* dependencies between constituents of a discourse tree, our DCRF model does not make any independence assumption among these properties. Furthermore, our

parsing model supports a bottom-up parsing algorithm which is non-greedy and provably optimal.

The discourse parser assumes that the input text has been already segmented into EDUs. As an additional contribution of this paper, we propose a novel discriminative approach to discourse segmentation that not only achieves state-of-the-art performance, but also reduces the time and space complexities by using fewer features. Notice that the combination of our segmenter with our parser forms a complete probabilistic discriminative framework for performing sentence-level discourse analysis.

Our framework was tested in a series of experiments. The empirical evaluation indicates that our approach to discourse parsing outperforms the state-of-the-art by a wide margin. Moreover, we show this to be the case on two very different genres: news articles and instructional how-to-do manuals.

In the rest of the paper, after discussing related work, we present our discourse parser. Then, we describe our segmenter. The experiments and the corpora we used are described next, followed by a discussion of the key results and some error analysis.

2 Related work

Automatic discourse analysis has a long history; see (Stede, 2011) for a detailed overview. Soricut and Marcu (2003) present the publicly available SPADE[1] system that comes with probabilistic models for *sentence-level* discourse segmentation and parsing based on lexical and syntactic features derived from the lexicalized syntactic tree of a sentence. Their parsing algorithm finds the most probable DT for a sentence, where the probabilities of the constituents are estimated by their parsing model. A constituent (e.g., ATTRIBUTION-NS[(1,2),3] in Figure 1) in a DT has two components, first, the *label* denoting the relation and second, the *structure* indicating which spans are being linked by the relation. The nuclearity statuses of the spans are built into the relation labels (e.g., NS[(1,2),3] means that span (1,2) is the NUCLEUS and it comes before span 3 which is the SATELLITE). SPADE is limited in several ways. It makes an independence assumption between the label and the structure while modeling a constituent, and it ignores the sequential and hierarchical dependencies between the constituents in the parsing model. Furthermore, SPADE relies only on lexico-syntactic features, and it follows a *generative* approach to estimate the model parameters for the segmentation and the parsing models. SPADE was trained and tested on the RST-DT corpus (Carlson et al., 2002), which contains human-annotated discourse trees for news articles.

Subsequent research addresses the question of how much syntax one really needs in discourse analysis. Sporleder and Lapata (2005) focus on discourse chunking, comprising the two subtasks of segmentation and non-hierarchical nuclearity assignment. More specifically, they examine whether features derived via part of speech (POS) and chunk taggers would be sufficient for these purposes. Their results on RST-DT turn out to be comparable to SPADE without using any features from the syntactic tree. Later, Fisher and Roark (2007) demonstrate over 4% absolute "performance gain" in segmentation, by combining the features extracted from the syntactic tree with the ones derived via taggers. Using quite a large number of features in a binary log-linear model they achieve the state-of-the-art segmentation performance on the RST-DT test set.

On the different genre of *instructional manuals*, Subba and Eugenio (2009) propose a shift-reduce parser that relies on a classifier to find the appropriate relation between two text segments. Their classifier is based on Inductive Logic Programming (ILP), which learns first-order logic rules from a large set of features including the linguistically rich *compositional semantics* coming from a semantic parser. They show that the compositional semantics improves the classification performance. However, their discourse parser implements a greedy approach (hence not optimal) and their classifier disregards the sequence and hierarchical dependencies.

Using RST-DT, Hernault et al. (2010) present the HILDA system that comes with a segmenter and a parser based on Support Vector Machines (SVMs). The segmenter is a binary SVM classifier which relies on the same lexico-syntactic features used in SPADE, but with more context. The discourse parser builds a DT iteratively utilizing two SVM classifiers in each iteration: (i) a binary classifier decides which of the two adjacent spans to link, and (ii) a multi-class classifier then connects the se-

[1] http://www.isi.edu/licensed-sw/spade/

lected spans with the appropriate relation. They use a very large set of features in their parser. However, taking a radically-greedy approach, they model structure and relations separately, and ignore the sequence dependencies in their models.

Recently, there has been an explosion of interest in Conditional Random Fields (CRFs) (Lafferty et al., 2001) for solving structured output classification problems, with many successful applications in NLP including syntactic parsing (Finkel et al., 2008), syntactic chunking (Sha and Pereira, 2003) and discourse chunking (Ghosh et al., 2011) in Penn Discourse Treebank (Prasad et al., 2008). CRFs being a discriminative approach to sequence modeling (i.e., directly models the conditional $p(\mathbf{y}|\mathbf{x}, \Theta)$), have several advantages over its generative counterparts such as Hidden Markov Models (HMMs) and Markov Random Fields (MRFs), which first model the joint $p(\mathbf{y}, \mathbf{x}|\Theta)$, then infer the conditional $p(\mathbf{y}|\mathbf{x}, \Theta)$). Key advantages include the ability to incorporate arbitrary overlapping local and global features, and the ability to relax strong independence assumptions. It has been advocated that CRFs are generally more accurate since they do not "waste effort" modeling complex distributions (i.e., $p(\mathbf{x})$) that are not relevant for the target task (Murphy, 2012).

3 The Discourse Parser

Assuming that a sentence is already segmented into a sequence of EDUs $e_1, e_2, \ldots e_n$ manually or by an automatic segmenter (see Section 4), the discourse parsing problem is to decide which spans to connect (i.e., *structure* of the DT) and which relations (i.e., *labels* of the internal nodes) to use in the process of building the hierarchical DT. To build the DTs effectively, a common assumption is that they are *binary* trees (Soricut and Marcu, 2003; duVerle and Prendinger, 2009). That is, multi-nuclear relations (e.g., LIST, JOINT, SEQUENCE) involving more than two EDUs are mapped to a hierarchical right-branching binary tree. For example, a flat $LIST(e_1, e_2, e_3, e_4)$ is mapped to a right-branching binary tree $LIST(e_1, LIST(e_2, LIST(e_3, e_4)))$.

Our discourse parser has two components. The first component, the *parsing model*, assigns a probability to every possible DT. The second component, the *parsing algorithm*, finds the most probable DT among the candidate discourse trees.

3.1 Parsing Model

A DT can be represented as a set of constituents of the form $R[i, m, j]$, which denotes a rhetorical relation R that holds between the span containing EDUs i through m, and the span containing EDUs $m{+}1$ through j. For example, the DT in Figure 1 can be written as {ELABORATION-NS[1,1,2], ATTRIBUTION-NS[1,2,3]}. Notice that a relation R also indicates the nuclearity assignments of the spans being connected, which can be one of NUCLEUS-SATELLITE (NS), SATELLITE-NUCLEUS (SN) and NUCLEUS-NUCLEUS (NN).

Given the model parameters Θ and a candidate DT T, for all the constituents c in T, our parsing model estimates the *conditional probability* $P(c|C, \Theta)$, which specifies the joint probability of the relation R and the structure $[i, m, j]$ associated with the constituent c, given that c has a set of sub-constituents C. For instance, for the DT shown in Figure 1, our model would estimate $P(R'[1, 1, 2]|\Theta)$, $P(R'[2, 2, 3]|\Theta)$, $P(R'[1, 2, 3]|R''[1, 1, 2], \Theta)$ etc. for all R' and R'' ranging on the set of relations. In what follows we describe our probabilistic parsing model to compute all these conditional probabilities $P(c|C, \Theta)$. We will demonstrate how our approach not only models the structure and the relation jointly, but it also captures *linear sequence dependencies* and *hierarchical dependencies* between constituents of a DT.

Our novel parsing model is the Dynamic Conditional Random Field (DCRF) (Sutton et al., 2007) shown in Figure 2. A DCRF is a generalization of linear-chain CRFs to represent complex interaction between labels, such as when performing multiple labeling tasks on the same sequence. The *observed* nodes W_j in the figure are the text spans. A text span can be either an EDU or a concatenation of a sequence of EDUs. The *structure* nodes $S_j{\in}\{0, 1\}$ in the figure represent whether text spans W_{j-1} and W_j should be connected or not. The *relation* nodes $R_j{\in}\{1 \ldots M\}$ denote the discourse relation between spans W_{j-1} and W_j, given that M is the total number of relations in our relation set. Notice that we now model the structure and the relation jointly and also take the sequential dependencies between adjacent constituents into consideration.

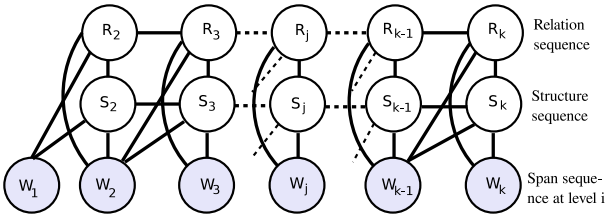

Figure 2: A Dynamic CRF as a discourse parsing model.

We can obtain the conditional probabilities of the constituents (i.e., $P(c|C,\Theta)$) of all candidate DTs for a sentence by applying the DCRF parsing model recursively at different levels, and by computing the posterior marginals of the relation-structure pairs. To illustrate, consider the example sentence in Figure 1 where we have three EDUs e_1, e_2 and e_3. The DCRF model for the first level is shown in Figure 3(a), where the (observed) EDUs are the spans in the span sequence. Given this model, we obtain the probabilities of the constituents $R[1,1,2]$ and $R[2,2,3]$ by computing the posterior marginals $P(R_2, S_2{=}1|e_1, e_2, e_3, \Theta)$ and $P(R_3, S_3{=}1|e_1, e_2, e_3, \Theta)$, respectively. At the second level (see Figure 3(b)), there are two possible span sequences $(e_{1:2}, e_3)$ and $(e_1, e_{2:3})$. In the first sequence, EDUs e_1 and e_2 are linked into a larger span, and in the second one, EDUs e_2 and e_3 are connected into a larger span. We apply our DCRF model to the two possible span sequences and obtain the probabilities of the constituents $R[1,2,3]$ and $R[1,1,3]$ by computing the posterior marginals $P(R_3, S_3{=}1|e_{1:2}, e_3, \Theta)$ and $P(R_{2:3}, S_{2:3}{=}1|e_1, e_{2:3}, \Theta)$, respectively.

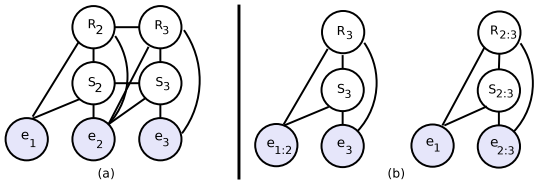

Figure 3: DCRF model applied to the sequences at different levels in the example in Fig. 1. (a) A sequence at the first level (b) Two possible sequences at the second level.

To further clarify the process, let us assume that the sentence contains four EDUs e_1, e_2, e_3 and e_4. At the first level (see Figure 4(a)), there is only one possible span sequence to which we apply our DCRF model. We obtain the probabilities of the constituents $R[1,1,2]$, $R[2,2,3]$ and $R[3,3,4]$ by computing the posterior marginals $P(R_2, S_2{=}1|e_1, e_2, e_3, e_4, \Theta)$, $P(R_3, S_3{=}1|e_1, e_2, e_3, e_4, \Theta)$ and $P(R_4, S_4{=}1|e_1, e_2, e_3, e_4, \Theta)$, respectively. At the second level (see Figure 4(b)), there are three possible span sequences $(e_{1:2}, e_3, e_4)$, $(e_1, e_{2:3}, e_4)$ and $(e_1, e_2, e_{3:4})$. When the DCRF model is applied to the first sequence $(e_{1:2}, e_3, e_4)$, we obtain the probabilities of the constituent $R[1,2,3]$ by computing the posterior marginal $P(R_3, S_3{=}1|e_{1:2}, e_3, e_4, \Theta)$. Likewise, the posterior marginals $P(R_{2:3}, S_{2:3}{=}1|e_1, e_{2:3}, e_4, \Theta)$ and $P(R_4, S_4{=}1|e_1, e_{2:3}, e_4, \Theta)$ in the DCRF model applied to the second sequence $(e_1, e_{2:3}, e_4)$ represents the probabilities of the constituents $R[1,1,3]$ and $R[2,3,4]$, respectively. Similarly, we attain the probabilities of the constituent $R[2,2,4]$ from the DCRF model applied to the sequence $(e_1, e_2, e_{3:4})$ by computing the posterior marginal $P(R_{3:4}, S_{3:4}{=}1|e_1, e_2, e_{3:4}, \Theta)$. At the third level (see Figure 4(c)), there are two possible span sequences $(e_{1:3}, e_4)$ and $(e_1, e_{2:4})$, to which we apply our DCRF model and acquire the probabilities of the constituents $R[1,3,4]$ and $R[1,1,4]$ by computing the posterior marginals $P(R_4, S_4{=}1|e_{1:3}, e_4, \Theta)$ and $P(R_{2:4}, S_{2:4}{=}1|e_1, e_{2:4}, \Theta)$, respectively.

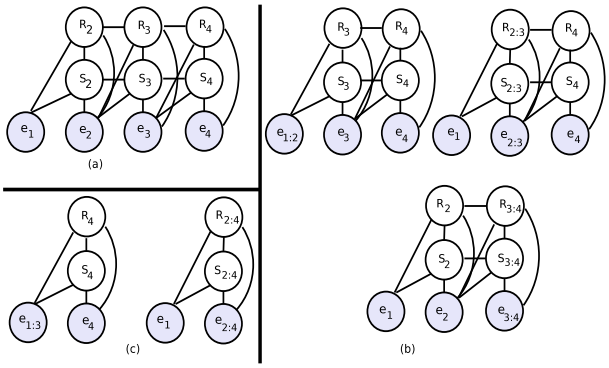

Figure 4: DCRF model applied to the sequences at different levels of a discourse tree. (a) A sequence at the first level, (b) Three possible sequences at the second level, (c) Two possible sequences at the third level.

Our DCRF model is designed using MALLET (McCallum, 2002). In order to avoid overfitting we regularize the DCRF model with l_2 regularization and learn the model parameters using the limited-memory BFGS (L-BFGS) fitting algorithm. Since

exact inference can be intractable in DCRF models, we perform approximate inference (to compute the posterior marginals) using tree-based reparameterization (Wainwright et al., 2002).

3.1.1 Features Used in the Parsing Model

Crucial to parsing performance is the set of features used, as summarized in Table 1. Note that these features are defined on two consecutive spans W_{j-1} and W_j of a span sequence. Most of the features have been explored in previous studies. However, we improve some of these as explained below.

Organizational features encode useful information about the surface structure of a sentence as shown by (duVerle and Prendinger, 2009). We measure the length of the spans in terms of the number of *EDUs* and *tokens* in it. However, in order to better adjust to the length variations, rather than computing their absolute numbers in a span, we choose to measure their *relative numbers* with respect to their total numbers in the sentence. For example, in a sentence containing three EDUs, a span containing two of these EDUs will have a relative EDU number of 0.67. We also measure the *distances* of the spans from the beginning and to the end of the sentence in terms of the number of EDUs.

8 organizational features
Relative number of EDUs in *span 1* and *span 2*.
Relative number of tokens in *span 1* and *span 2*.
Distances of span 1 in EDUs to the *beginning* and to the *end*.
Distances of span 2 in EDUs to the *beginning* and to the *end*.
8 N-gram features
Beginning and *end* lexical N-grams in span 1.
Beginning and *end* lexical N-grams in span 2.
Beginning and *end* POS N-grams in span 1.
Beginning and *end* POS N-grams in span 2.
5 dominance set features
Syntactic labels of the *head* node and the *attachment* node.
Lexical heads of the *head* node and the *attachment* node.
Dominance relationship between the two text spans.
2 contextual features
Previous and *next* feature vectors.
2 substructure features
Root nodes of the *left* and *right* rhetorical subtrees.

Table 1: Features used in the DCRF parsing model.

Discourse connectives (e.g., *because*, *but*), when present, signal rhetorical relations between two text segments (Knott and Dale, 1994; Marcu, 2000a). However, previous studies (e.g., Hernault et al. (2010), Biran and Rambow (2011)) suggest that an empirically acquired lexical N-gram dictionary is more effective than a fixed list of connectives, since this approach is domain independent and capable of capturing non-lexical cues such as punctuations. To build the *lexical* **N-gram** dictionary empirically from the training corpus we consider the first and last N tokens ($N \in \{1, 2\}$) of each span and rank them according to their mutual information[2] with the two labels, *Structure* and *Relation*. Intuitively, the most informative cues are not only the most frequent, but also the ones that are indicative of the labels in the training data (Blitzer, 2008). In addition to the lexical N-grams we also encode *POS* tags of the first and last N tokens ($N \in \{1, 2\}$) as features.

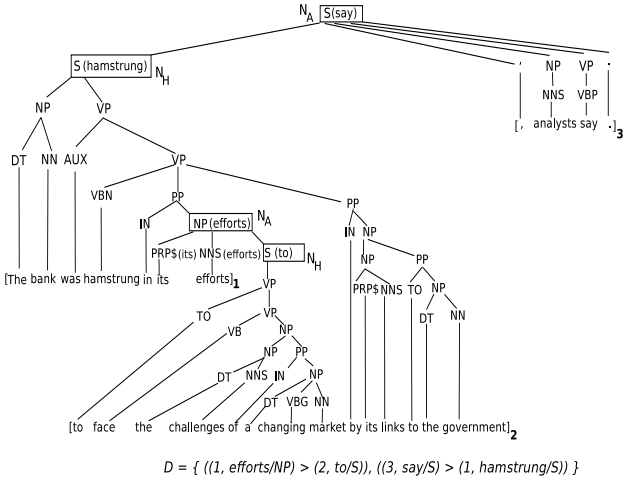

Figure 5: A discourse segmented lexicalized syntactic tree. Boxed nodes form the dominance set D.

Dominance set extracted from the Discourse Segmented Lexicalized Syntactic Tree (DS-LST) (Soricut and Marcu, 2003) has been shown to be a very effective feature in SPADE. Figure 5 shows the DS-LST for our running example (see Figure 1 and 3). In a DS-LST, each EDU except the one with the root node must have a *head node* N_H that is attached to an *attachment node* N_A residing in a separate EDU. A dominance set D (shown at the bottom of Figure 5 for our example) contains these *attachment* points of the EDUs in a DS-LST. In addition to the syntactic and lexical information of the head and attachment nodes, each element in D also represents a dominance relationship between the EDUs involved. The

[2] In contrast, HILDA ranks the N-grams by frequencies.

EDU with N_A dominates the EDU with N_H. In order to extract dominance set features for two consecutive spans $e_{i:j}$ and $e_{j+1:k}$, we first compute D from the DS-LST of the sentence. We then extract the element from D that holds across the EDUs j and $j+1$. In our running example, for the spans e_1 and e_2 (Figure 3(a)), the relevant dominance set element is *(1, efforts/NP)>(2, to/S)*. We encode the syntactic labels and lexical heads of N_H and N_A and the dominance relationship (i.e., which of the two spans is dominating) as features in our model.

We also incorporate more **contextual** information by including the above features computed for the neighboring span pairs in the current feature vector.

We incorporate *hierarchical dependencies* between constituents in a DT by means of the **substructure** features. For the two adjacent spans $e_{i:j}$ and $e_{j+1:k}$, we extract the roots of the rhetorical subtrees spanning over $e_{i:j}$ (left) and $e_{j+1:k}$ (right). In our example (see Figure 1 and Figure 3 (b)), the root of the rhetorical subtree spanning over $e_{1:2}$ is ELABORATION-NS. However, this assumes the presence of a labeled DT which is not the case when we apply the parser to a new sentence. This problem can be easily solved by looping twice through building the model and the parsing algorithm (described below). We first build the model without considering the substructure features. Then we find the optimal DT employing our parsing algorithm. This intermediate DT will now provide labels for the substructures. Next we can build a new, more accurate model by including the substructure features, and run again the parsing algorithm to find the final optimal DT.

3.2 Parsing Algorithm

Our parsing model above assigns a conditional probability to every possible DT constituent for a sentence, the job of the parsing algorithm is to find the most probable DT. Formally, this can be written as,

$DT^* = argmax_{DT} P(DT|\Theta)$

Our discourse parser implements a probabilistic CKY-like bottom-up algorithm for computing the most likely parse of a sentence using dynamic programming; see (Jurafsky and Martin, 2008) for a description. Specifically, with n number of EDUs in a sentence, we use the upper-triangular portion of the $n \times n$ Dynamic Programming Table (DPT). The cell $[i, j]$ in the DPT represents the span containing EDUs i through j and stores the probability of a constituent $R[i, m, j]$, where $m = argmax_{i \leq k \leq j} P(R[i, k, j])$.

In contrast to HILDA which implements a greedy algorithm, our approach finds a DT that is globally optimal. Our approach is also different from SPADE's implementation. SPADE first finds the *tree structure* that is globally optimal, then it assigns the most probable *relations* to the internal nodes. More specifically, the cell $[i, j]$ in SPADE's DPT stores the probability of a constituent $R[i, m, j]$, where $m = argmax_{i \leq k \leq j} P([i, k, j])$. Disregarding the relation label R while building the DPT, this approach may find a tree that is **not** globally optimal.

4 The Discourse Segmenter

Our discourse parser above assumes that the input sentences have been already segmented into EDUs. Since it has been shown that discourse segmentation is a primary source of inaccuracy for discourse parsing (Soricut and Marcu, 2003), we have developed our own segmenter, that not only achieves state-of-the-art performance as shown later, but also reduces the time complexity by using fewer features.

Our segmenter implements a binary classifier to decide for each word (except the last word) in a sentence, whether to put an EDU boundary *after* that word. We use a Logistic Regression (LR) (i.e., discriminative) model with l_2 regularization and learn the model parameters using the L-BFGS algorithm, which gives quadratic convergence rate. To avoid overfitting, we use 5-fold cross validation to learn the regularization strength parameter from the training data. We also use a simple *bagging* technique (Breiman, 1996) to deal with the sparsity of *boundary* tags. Note that, our first attempt at this task implemented a linear-chain CRF model to capture the sequence dependencies between the tags in a discriminative way. However, the binary LR classifier, using the same features, not only outperforms the CRF model, but also reduces the space complexity.

4.1 Features Used in the Segmentation Model

Our set of features for discourse segmentation are mostly inspired from previous studies but used in a novel way as we describe below.

Our first subset of features which we call *SPADE*

features, includes the lexico-syntactic patterns extracted from the lexicalized syntactic tree for the given sentence. These features replicates the features used in SPADE, but used in a discriminative way. To decide on an EDU boundary after a token w_k, we find the lowest constituent in the lexicalized syntactic tree that spans over tokens $w_i \dots w_j$ such that $i \leq k < j$. The production that expands this constituent in the tree and its different variations, form the feature set. For example in Figure 5, the production *NP(efforts)→PRP$(its)NNS(efforts)↑S(to)* and its different variations depending on whether they include the lexical heads and how many non-terminals (up to two) to consider before and after the potential EDU boundary (↑), are used to determine the existence of a boundary after the word *efforts* (see (Fisher and Roark, 2007) for details). SPADE uses these features in a generative way, meaning that, it inserts an EDU boundary if the relative frequency (i.e., Maximum Likelihood Estimate (MLE)) of a potential boundary given the production in the training corpus is greater than 0.5. If the production has not been observed frequently enough, it uses its other variations to perform further smoothing. In contrast, we compute the MLE estimates for a production and its other variations, and use those as features with/without binarizing the values.

Shallow syntactic parse (or *Chunk*) and *POS* tags have been shown to possess valuable cues for discourse segmentation (Fisher and Roark, 2007). For example, it is less likely that an EDU boundary occurs within a chunk. We, therefore, annotate the tokens of a sentence with chunk and POS tags by a state-of-the-art tagger[3] and encode these as features.

EDUs are normally multi-word strings. Thus, a token near the beginning or end of a sentence is unlikely to be the end of a segment. Therefore, for each token we include its *relative position* in the sentence and *distances* to the beginning and end as features.

It is unlikely that two consecutive tokens are tagged with EDU boundaries. We incorporate *contextual* information for a token by including the above features computed for its neighboring tokens.

We also experimented with different N-gram ($N \in \{1, 2, 3\}$) features extracted from the token sequence, POS sequence and chunk sequence. However, since such features did not improve the segmentation accuracy on the development set, they were excluded from our final set of features.

5 Experiments

5.1 Corpora

To demonstrate the generality of our model, we experiment with two different genres. First, we use the standard *RST-DT* corpus (Carlson et al., 2002) that contains discourse annotations for 385 Wall Street Journal news articles from the Penn Treebank (Marcus et al., 1994). Second, we use the *Instructional* corpus developed by Subba and Eugenio (2009) that contains discourse annotations for 176 instructional how-to-do manuals on home-repair.

The RST-DT corpus is partitioned into a training set of 347 documents (7673 sentences) and a test set of 38 documents (991 sentences), and 53 documents (1208 sentences) have been (doubly) annotated by two human annotators, based on which we compute the human agreement. We use the human-annotated syntactic trees from Penn Treebank to train SPADE in our experiments using RST-DT as done in (Soricut and Marcu, 2003). We extracted a sentence-level DT from a document-level DT by finding the subtree that exactly spans over the sentence. By our count, 7321 sentences in the training set, 951 sentences in the test set and 1114 sentences in the doubly-annotated set have a well-formed DT in RST-DT. The Instructional corpus contains 3430 sentences in total, out of which 3032 have a well-formed DT. This forms our sentence-level corpora for discourse parsing. However, the existence of a well-formed DT in not a necessity for discourse segmentation, therefore, we do not exclude any sentence in our discourse segmentation experiments.

5.2 Experimental Setup

We perform our experiments on discourse parsing in RST-DT with the 18 coarser relations (see Figure 6) defined in (Carlson and Marcu, 2001) and also used in SPADE and HILDA. By attaching the nuclearity statuses (i.e., NS, SN, NN) to these relations we get 39 distinct relations[4]. Our experiments on the Instructional corpus consider the same 26 primary relations (e.g., GOAL:ACT,

[3] http://cogcomp.cs.illinois.edu/page/software

[4] Not all relations take all the possible nuclearity statuses.

CAUSE:EFFECT, GENERAL-SPECIFIC) used in (Subba and Eugenio, 2009) and also treat the reversals of non-commutative relations as separate relations. That is, PREPARATION-ACT and ACT-PREPARATION are two different relations. Attaching the nuclearity statuses to these relations gives 70 distinct relations in the Instructional corpus.

We use SPADE as our baseline model and apply the same modifications to its default setting as described in (Fisher and Roark, 2007), which delivers improved performance. Specifically, in testing, we replace the Charniak parser (Charniak, 2000) with a more accurate reranking parser (Charniak and Johnson, 2005). We use the reranking parser in all our models to generate the syntactic trees. This parser was trained on the sections of the Penn Treebank not included in the test set. For a fair comparison, we apply the same canonical lexical head projection rules (Magerman, 1995; Collins, 2003) to lexicalize the syntactic trees as done in SPADE and HILDA. Note that, all the previous works described in Section 2, report their models' performance on a particular test set of a specific corpus. To compare our results with the previous studies, we test our models on those specific test sets. In addition, we show more general performance based on 10-fold cross validation.

5.3 Parsing based on Manual Segmentation

First, we present the results of our discourse parser based on *manual* segmentation. The parsing performance is assessed using the unlabeled (i.e., span) and labeled (i.e., nuclearity, relation) precision, recall and F-score as described in (Marcu, 2000b, page 143). For brevity, we report only the F-scores in Table 2. Notice that, our parser (DCRF) consistently outperforms SPADE (SP) on the RST-DT test set[5]. Especially, on relation labeling, which is the hardest among the three tasks, we get an absolute F-score improvement of 9.5%, which represents a relative error rate reduction of 29.3%. Our F-score of 77.1 in relation labeling is also close to the human agreement (i.e., F-score of 83.0) on the doubly-annotated data. Our results on the RST-DT test set are consistent with the mean scores over 10-folds, when we perform 10-fold cross validation on RST-DT.

The improvement is even larger on the Instructional corpus, where we compare our mean results over 10-folds with the results reported in Subba and Eugenio (S&E) (2009) on a test set[6], giving absolute F-score improvements of 4.8%, 15.5% and 10.6% in span, nuclearity and relations, respectively. Our parser reduces the errors by 67.6%, 54.6% and 28.6% in span, nuclearity and relations, respectively.

	RST-DT				Instructional	
	Test set		10-fold	Doubly	S&E	10-fold
Scores	SP	DCRF	DCRF	Human	ILP	DCRF
Span	93.5	**94.6**	93.7	95.7	92.9	**97.7**
Nuc.	85.8	**86.9**	85.2	90.4	71.8	**87.2**
Rel.	67.6	**77.1**	75.4	83.0	63.0	**73.6**

Table 2: Parsing results using *manual* segmentation.

If we compare the performance of our model on the two corpora, we see that our model is more accurate in finding the right tree structure (see Span) on the Instructional corpus. This may be due to the fact that sentences in the Instructional domain are relatively short and contain fewer EDUs than sentences in the News domain, thus making it easier to find the right tree structure. However, when we compare the performance on the relation labeling task, we observe a decrease on the Instructional corpus. This may be due to the small amount of data available for training and the imbalanced distribution of a large number of discourse relations in this corpus.

To analyze the features, Table 3 presents the parsing results on the RST-DT test set using different subsets of features. Every new subset of features appears to improve the accuracy. More specifically, when we add the *organizational* features with the *dominance set* features (see S_2), we get about 2% absolute improvement in nuclearity and relations. With *N-gram* features (S_3), the gain is even higher; 6% in relations and 3.5% in nuclearity, demonstrating the utility of the N-gram features. This is consistent with the findings of (duVerle and Prendinger, 2009; Schilder, 2002). Including the *Contextual* features (S_4), we get further 3% and 2.2% improvements in nuclearity and relations, respectively. Notice that, adding the *substructure* features (S_5) does

[5]The improvements are statistically significant ($p < 0.01$).

[6]Subba and Eugenio (2009) report their results based on an arbitrary split between a training set and a test set. We asked the authors for their particular split. However, since we could not obtain that information, we compare our model's performance based on 10-fold cross validation with their reported results.

not help much in sentence-level parsing, giving only an improvement of 0.8% in relations. Therefore, one may choose to avoid using this computationally expensive feature in time-constrained scenarios. However, in the future, it will be interesting to see its importance in document-level parsing with large trees.

Scores	S_1	S_2	S_3	S_4	S_5
Span	91.3	92.1	93.3	94.6	94.6
Nuclearity	78.2	80.3	83.8	86.8	86.9
Relation	66.2	68.1	74.1	76.3	77.1

Table 3: Parsing results based on manual segmentation using different subsets of features on RST-DT test set. Feature subsets S_1 = {Dominance set}, S_2 = {Dominance set, Organizational}, S_3 = {Dominance set, Organizational, N-gram}, S_4 = {Dominance set, Organizational, N-gram, Contextual}, S_5 (all) = {Dominance set, Organizational, N-gram, Contextual, Substructure}.

5.4 Evaluation of the Discourse Segmenter

We evaluate the segmentation accuracy with respect to the intra-sentential segment boundaries following (Fisher and Roark, 2007). Specifically, if a sentence contains n EDUs, which corresponds to $n-1$ intra-sentence segment boundaries, we measure the model's ability to correctly identify these $n-1$ boundaries. Human agreement for this task is quite high (F-score of 98.3) on RST-DT.

Table 4 shows the results of different models in (P)recision, (R)ecall, and (F)-score on the two corpora. We compare our model's (LR) results with HILDA (HIL), SPADE (SP) and the results reported in Fisher and Roark (F&R) (2007) on the RST-DT test set. HILDA gives the weakest performance[7]. Our results are also much better than SPADE[8], with an absolute F-score improvement of 4.9%, and comparable to the results of F&R, even though we use fewer features. Furthermore, we perform 10-fold cross validation on both corpora and compare with SPADE. However, SPADE does not come with a training module for its segmenter. We reimplemented this module and verified it on the RST-DT test set. Due to the lack of human-annotated syntactic trees in the *Instructional* corpus, we train SPADE in this corpus using the syntactic trees produced by the reranking parser. Our model delivers absolute F-score improvements of 3.8% and 8.1% on the RST-DT and the Instructional corpora, respectively, which is statistically significant in both cases ($p < 3.0e-06$). However, when we compare our results on the two corpora, we observe a substantial decrease in performance on the Instructional corpus. This could be due to a smaller amount of data in this corpus and the inaccuracies in the syntactic parser and taggers, which are trained on news articles.

	RST-DT						Instructional	
	Test Set				10-fold		10-fold	10-fold
	HIL	SP	F&R	LR	SP	LR	SP	LR
P	77.9	83.8	**91.3**	88.0	83.7	**87.5**	65.1	**73.9**
R	70.6	86.8	89.7	**92.3**	86.2	**89.9**	82.8	**89.7**
F	74.1	85.2	**90.5**	90.1	84.9	**88.7**	72.8	**80.9**

Table 4: Segmentation results of different models.

5.5 Parsing based on Automatic Segmentation

In order to evaluate our full system, we feed our discourse parser the output of our discourse segmenter. Table 5 shows the F-score results. We compare our results with SPADE on the RST-DT test set. We achieve absolute F-score improvements of 3.6%, 3.4% and 7.4% in span, nuclearity and relation, respectively. These improvements are statistically significant ($p<0.001$). Our system, therefore, reduces the errors by 15.5%, 11.4%, and 17.6% in span, nuclearity and relations, respectively. These results are also consistent with the mean results over 10-folds.

	RST-DT			Instructional
	Test set		10-fold	10-fold
Scores	SPADE	DCRF	DCRF	DCRF
Span	76.7	**80.3**	78.7	71.9
Nuclearity	70.2	**73.6**	72.2	64.3
Relation	58.0	**65.4**	64.2	54.8

Table 5: Parsing results using *automatic* segmentation.

For the Instructional corpus, the last column of Table 5 shows the mean 10-fold cross validation results. We cannot compare with S&E because no results were reported using an automatic segmenter. However, it is interesting to observe how much our full system is affected by an automatic segmenter on both RST-DT and the Instructional corpus (see Table 2 and Table 5). Nevertheless, taking into account the segmentation results in Table 4, this is

[7] Note that, the high segmentation accuracy reported in (Hernault et al., 2010) is due to a less stringent evaluation metric.

[8] The improvements are statistically significant ($p<2.4e-06$)

not surprising because previous studies (Soricut and Marcu, 2003) have already shown that automatic segmentation is the primary impediment to high accuracy discourse parsing. This demonstrates the need for a more accurate segmentation model in the Instructional genre. A promising future direction would be to apply effective domain adaptation methods (e.g., `easyadapt` (Daume, 2007)) to improve the segmentation performance in the Instructional domain by leveraging the rich data in RST-DT.

5.6 Error Analysis and Discussion

The results in Table 2 suggest that given a manually segmented discourse, our sentence-level discourse parser finds the unlabeled (i.e., span) discourse tree and assigns the nuclearity statuses to the spans at a performance level close to human annotators. We, therefore, look more closely into the performance of our parser on the hardest task of *relation labeling*.

Figure 6 shows the confusion matrix for the relation labeling task using manual segmentation on the RST-DT test set. The relation labels are ordered according to their frequency in the RST-DT training set and represented by their initial letters. For example, EL represents ELABORATION and CA represents CAUSE. In general, errors can be explained by two different phenomena acting together: (i) the frequency of the relations in the training data, and (ii) the semantic (or pragmatic) similarity between the relations. The most frequent relations (e.g., ELABORATION) tend to confuse the less frequent ones (e.g., SUMMARY), and the relations which are semantically similar (e.g., CAUSE, EXPLANATION) confuse each other, making it hard to distinguish for the computational models. Notice that, the confusions caused by JOINT appears to be high considering its frequency. The confusion between JOINT and TEMPORAL may be due to the fact that both of these coarser relations[9] contain finer relations (i.e., *list* in JOINT and *sequence* in TEMPORAL), which are semantically similar, as pointed out by Carlson and Marcu (2001). The confusion between JOINT and BACKGROUND may be explained by their different (semantic vs. pragmatic) interpretation in the RST theory (Stede, 2011, page 85).

[9]JOINT is actually not a relation, but is characterized by juxtaposition of two EDUs without a relation.

	TO	EV	SU	MA	COMP	EX	COND	TE	CA	EN	BA	CONT	JO	SA	AT	EL
TO	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	2
EV	0	0	0	0	0	0	0	0	0	0	0	0	1	1	3	2
SU	0	0	6	0	0	0	0	0	0	0	0	1	2	0	0	10
MA	0	0	0	10	0	1	0	1	0	0	0	0	2	0	1	7
COMP	0	0	0	1	1	1	0	0	2	0	3	2	1	0	0	6
EX	0	0	0	0	0	9	0	0	4	1	2	0	0	1	4	1
COND	0	0	0	0	0	0	20	3	0	1	1	1	1	2	6	7
TE	0	0	0	0	0	0	0	11	1	0	5	0	9	4	2	9
CA	0	0	0	1	0	4	0	1	5	4	1	1	6	1	6	3
EN	0	0	0	1	0	0	0	1	0	24	2	0	1	1	1	9
BA	0	0	0	0	1	1	2	7	1	0	15	2	7	4	6	15
CONT	0	0	0	0	1	1	2	1	0	0	4	26	4	6	5	6
JO	0	0	0	0	0	2	0	3	1	0	3	1	43	7	4	13
SA	0	0	2	0	0	0	3	2	0	3	0	0	0	80	3	31
AT	0	1	0	0	0	3	3	2	2	0	2	2	1	15	276	20
EL	1	0	1	3	2	3	2	5	5	11	5	6	14	9	19	295

Figure 6: Confusion matrix for the relation labels on the RST-DT test set. Y-axis represents *true* and X-axis represents *predicted* labels. The relation labels are **TO**PIC-COMMENT, **EV**ALUATION, **SU**MMARY, **MA**NNER-MEANS, **COMP**ARISON, **EX**PLANATION, **COND**ITION, **TE**MPORAL, **CA**USE, **EN**ABLEMENT, **BA**CKGROUND, **CONT**RAST, **JO**INT, **SA**ME-UNIT, **AT**TRIBUTION, **EL**ABORATION.

Based on these observations we will pursue two ways to improve our discourse parser. We need a more robust (e.g., *bagging*) method to deal with the imbalanced distribution of relations, along with a better representation of semantic knowledge. For example, *compositional semantics* (Subba and Eugenio, 2009) and *subjectivity* (Somasundaran, 2010) can be quite relevant for identifying relations.

6 Conclusion

In this paper, we have described a complete probabilistic discriminative framework for performing sentence-level discourse analysis. Experiments indicate that our approach outperforms the state-of-the-art on two corpora, often by a wide margin.

In ongoing work, we plan to generalize our DCRF-based parser to multi-sentential text and also verify to what extent parsing and segmentation can be jointly performed. A longer term goal is to extend our framework to also work with graph structures of discourse, as recommended by several recent discourse theories (Wolf and Gibson, 2005). Once we achieve similar performance on graph structures, we will perform extrinsic evaluation to determine their relative utility for various NLP tasks.

Acknowledgments

We are grateful to G. Murray, J. CK Cheung, the 3 reviewers and the NSERC CGS-D award.

References

Or Biran and Owen Rambow. 2011. Identifying Justifications in Written Dialogs by Classifying Text as Argumentative. *Int. J. Semantic Computing*, 5(4):363–381.

J. Blitzer, 2008. *Domain Adaptation of Natural Language Processing Systems*. PhD thesis, University of Pennsylvania.

L. Breiman. 1996. Bagging predictors. *Machine Learning*, 24(2):123–140, August.

L. Carlson and D. Marcu. 2001. Discourse Tagging Reference Manual. Technical Report ISI-TR-545, University of Southern California Information Sciences Institute.

L. Carlson, D. Marcu, and M. Okurowski. 2002. RST Discourse Treebank (RST-DT) LDC2002T07. *Linguistic Data Consortium, Philadelphia*.

E. Charniak and M. Johnson. 2005. Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 173–180, NJ, USA. ACL.

E. Charniak. 2000. A Maximum-Entropy-Inspired Parser. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, pages 132–139, Seattle, Washington. ACL.

M. Collins. 2003. Head-Driven Statistical Models for Natural Language Parsing. *Computational Linguistics*, 29(4):589–637, December.

H. Daume. 2007. Frustratingly Easy Domain Adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 256–263, Prague, Czech Republic. ACL.

D. duVerle and H. Prendinger. 2009. A Novel Discourse Parser based on Support Vector Machine Classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 665–673, Suntec, Singapore. ACL.

J. Finkel, A. Kleeman, and C. Manning. 2008. Efficient, Feature-based, Conditional Random Field Parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 959–967, Columbus, Ohio, USA. ACL.

S. Fisher and B. Roark. 2007. The Utility of Parse-derived Features for Automatic Discourse Segmentation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 488–495, Prague, Czech Republic. ACL.

S. Ghosh, R. Johansson, G. Riccardi, and S. Tonelli. 2011. Shallow Discourse Parsing with Conditional Random Fields. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 1071–1079, Chiang Mai, Thailand. AFNLP.

H. Hernault, H. Prendinger, D. duVerle, and M. Ishizuka. 2010. HILDA: A Discourse Parser Using Support Vector Machine Classification. *Dialogue and Discourse*, 1(3):1–33.

D. Jurafsky and J. Martin, 2008. *Speech and Language Processing*, chapter 14. Prentice Hall.

A. Knott and R. Dale. 1994. Using Linguistic Phenomena to Motivate a Set of Coherence Relations. *Discourse Processes*, 18(1):35–62.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

D. Magerman. 1995. Statistical Decision-tree Models for Parsing. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 276–283, Cambridge, Massachusetts. ACL.

W. Mann and S. Thompson. 1988. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text*, 8(3):243–281.

D. Marcu. 2000a. The Rhetorical Parsing of Unrestricted Texts: A Surface-based Approach. *Computational Linguistics*, 26:395–448.

D. Marcu. 2000b. *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, MA, USA.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1994. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

A. McCallum. 2002. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu.

K. Murphy. 2012. *Machine Learning A Probabilistic Perspective (Forthcoming, August 2012)*. MIT Press, Cambridge, MA, USA.

R. Prasad, A. Joshi, N. Dinesh, A. Lee, E. Miltsakaki, and B. Webber. 2005. The Penn Discourse TreeBank as a Resource for Natural Language Generation. In *Proceedings of the Corpus Linguistics Workshop on Using Corpora for Natural Language Generation*, pages 25–32, Birmingham, U.K.

R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*, pages 2961–2968, Marrakech, Morocco. ELRA.

F. Schilder. 2002. Robust Discourse Parsing via Discourse Markers, Topicality and Position. *Natural Language Engineering*, 8(3):235–255, June.

F. Sha and F. Pereira. 2003. Shallow Parsing with Conditional Random Fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 134–141, Edmonton, Canada. ACL.

S. Somasundaran, 2010. *Discourse-Level Relations for Opinion Analysis*. PhD thesis, University of Pittsburgh.

R. Soricut and D. Marcu. 2003. Sentence Level Discourse Parsing Using Syntactic and Lexical Information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 149–156, Edmonton, Canada. ACL.

C. Sporleder and M. Lapata. 2005. Discourse Chunking and its Application to Sentence Compression. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 257–264, Vancouver, British Columbia, Canada. ACL.

M. Stede. 2011. *Discourse Processing*. Synthesis Lectures on Human Language Technologies. Morgan And Claypool Publishers, November.

R. Subba and B. Di Eugenio. 2009. An Effective Discourse Parser that Uses Rich Linguistic Information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 566–574, Boulder, Colorado. ACL.

C. Sutton, A. McCallum, and K. Rohanimanesh. 2007. Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data. *Journal of Machine Learning Research (JMLR)*, 8:693–723.

S. Verberne, L. Boves, N. Oostdijk, and P. Coppen. 2007. Evaluating Discourse-based Answer Extraction for Why-question Answering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 735–736, Amsterdam, The Netherlands. ACM.

M. Wainwright, T. Jaakkola, and A. Willsky. 2002. Tree-based Reparameterization for Approximate Inference on Loopy Graphs. In *Advances in Neural Information Processing Systems 14*, pages 1001–1008. MIT Press.

F. Wolf and E. Gibson. 2005. Representing Discourse Coherence: A Corpus-Based Study. *Computational Linguistics*, 31:249–288, June.