Sentence Level Discourse Parsing using Syntactic and Lexical Information

Radu Soricut and Daniel Marcu
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
{radu, marcu}@isi.edu

Abstract

We introduce two probabilistic models that can be used to identify elementary discourse units and build sentence-level discourse parse trees. The models use syntactic and lexical features. A discourse parsing algorithm that implements these models derives discourse parse trees with an error reduction of 18.8% over a state-of-the-art decision-based discourse parser. A set of empirical evaluations shows that our discourse parsing model is sophisticated enough to yield discourse trees at an accuracy level that matches near-human levels of performance.

1 Introduction

By exploiting information encoded in human-produced syntactic trees (Marcus et al., 1993), research on probabilistic models of syntax has driven the performance of syntactic parsers to about 90% accuracy (Charniak, 2000; Collins, 2000). The absence of semantic and discourse annotated corpora prevented similar developments in semantic/discourse parsing. Fortunately, recent annotation projects have taken significant steps towards developing semantic (Fillmore et al., 2002; Kingsbury and Palmer, 2002) and discourse (Carlson et al., 2003) annotated corpora. Some of these annotation efforts have already had a computational impact. For example, Gildea and Jurafsky (2002) developed statistical models for automatically inducing semantic roles. In this paper, we describe probabilistic models and algorithms that exploit the discourse-annotated corpus produced by Carlson et al. (2003).

A discourse structure is a tree whose leaves correspond to *elementary discourse units (edu)*s, and whose internal nodes correspond to contiguous text spans (called discourse spans). An example of a discourse structure is the tree given in Figure 1. Each internal node in a discourse tree is characterized by a *rhetorical relation*, such as ATTRIBUTION and ENABLEMENT. Within a rhetorical relation a discourse span is also labeled as either NUCLEUS or SATELLITE. The distinction between nuclei and satellites comes from the empirical observation that a nucleus expresses what is more essential to the writer's purpose than a satellite. Discourse trees can be represented graphically in the style shown in Figure 1. The arrows link the satellite to the nucleus of a rhetorical relation. Arrows are labeled with the name of the rhetorical relation that holds between the linked units. Horizontal lines correspond to text spans, and vertical lines identify text spans which are nuclei.

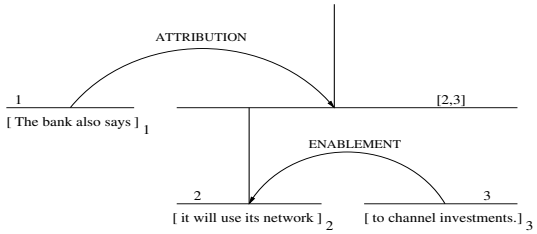

Figure 1: Discourse structure of a sentence.

In this paper, we introduce two probabilistic models that can be used to identify elementary discourse units and build sentence-level discourse parse trees. We show how syntactic and lexical information can be exploited in the process of identifying elementary units of discourse and building sentence-level discourse trees. Our evaluation indicates that the discourse parsing model we propose is sophisticated enough to achieve near-human levels of performance on the task of deriving sentence-level discourse trees, when working with human-produced syntactic trees and discourse segments.

2 The Corpus

For the experiments described in this paper, we use a publicly available corpus (RST-DT, 2002) that contains 385

Wall Street Journal articles from the Penn Treebank. The corpus comes conveniently partitioned into a Training set of 347 articles (6132 sentences) and a Test set of 38 articles (991 sentences). Each document in the corpus is paired with a discourse structure (tree) that was manually built in the style of Rhetorical Structure Theory (Mann and Thompson, 1988). (See (Carlson et al., 2003) for details concerning the corpus and the annotation process.) Out of the 385 articles in the corpus, 53 have been independently annotated by two human annotators. We used this doubly-annotated subset to compute human agreement on the task of discourse structure derivation. In our experiments we used as discourse structures only the discourse sub-trees spanning over individual sentences.

Because the discourse structures had been built on top of sentences already associated with syntactic trees from the Penn Treebank, we were able to create a composite corpus which allowed us to perform an empirically driven syntax-discourse relationship study. This composite corpus was created by associating each sentence s in the discourse corpus with its corresponding Penn Treebank syntactic parse tree $syntacticTree(s)$ and its corresponding sentence-level discourse tree $discourseTree(s)$. Although human annotators were free to build their discourse structures without enforcing the existence of well-formed discourse sub-trees for each sentence, in about 95% of the cases in the (RST-DT, 2002) corpus, there exists a discourse sub-tree $discourseTree(s)$ associated with each sentence s. The remaining 5% of the sentences cannot be used in our approach, as no well-formed discourse tree can be associated with these sentences.

Therefore, our Training section consists of a set of 5809 triples of the form

$$\langle s, syntacticTree(s), discourseTree(s)\rangle$$

which are used to train the parameters of the statistical models. Our Test section consists of a set of 946 triples of a similar form, which are used to evaluate the performance of our discourse parser.

The (RST-DT, 2002) corpus uses 110 different rhetorical relations. We found it useful to also compact these relations into classes, as described by Carlson et al. (2003), and operate with the resulting 18 labels as well (seen as coarser granularity rhetorical relations). Operating with different levels of granularity allows one to get deeper insight into the difficulties of assigning the appropriate rhetorical relation, if any, to two adjacent text spans.

3 The Discourse Segmenter

We break down the problem of building sentence-level discourse trees into two sub-problems: *discourse segmentation* and *discourse parsing*. Discourse segmentation is covered by this section, while discourse parsing is covered by Section 4.

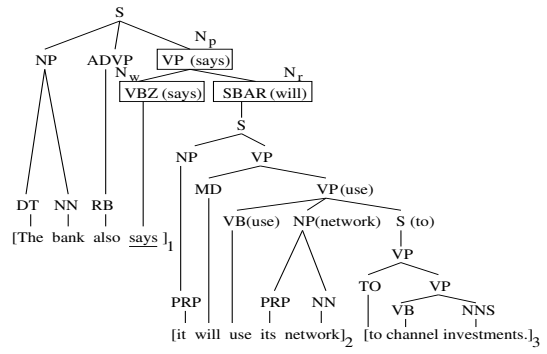

Figure 2: Discourse segmentation using lexicalized syntactic trees.

Discourse segmentation is the process in which a given text is broken into non-overlapping segments called elementary discourse units (*edu*s). In the present work, elementary discourse units are taken to be clauses or clause-like units that are unequivocally the NUCLEUS or SATELLITE of a rhetorical relation that holds between two adjacent spans of text (see (Carlson et al., 2003) for details). Our approach to discourse segmentation breaks the problem further into two sub-problems: *sentence segmentation* and *sentence-level discourse segmentation*. The problem of sentence segmentation has been studied extensively, and tools such as those described by Palmer and Hearst (1997) and Ratnaparkhi (1998) can handle it well. In this section, we present a discourse segmentation algorithm that deals with segmenting sentences into elementary discourse units.

3.1 The Discourse Segmentation Model

The discourse segmenter proposed here takes as input a sentence and outputs its elementary discourse unit boundaries. Our statistical approach to sentence segmentation uses two components: a *statistical model* which assigns a probability to the insertion of a discourse boundary after each word in a sentence, and a *segmenter*, which uses the probabilities computed by the model for inserting discourse boundaries. We first focus on the statistical model.

A good model of discourse segmentation needs to account both for local interactions at the word level and for global interactions at more abstract levels. Consider, for example, the syntactic tree in Figure 2. According to our hypothesis, the discourse boundary inserted between the words `says` and `it` is best explained not by the words alone, but by the lexicalized syntactic structure `[VP(says) [VBZ(says)`$_{\uparrow}$`SBAR(will)]]`, signaled by the boxed nodes in Figure 2. Hence, we hypothesize that the discourse boundary in our example is best explained by the global interaction between the verb (the act of saying) and its clausal complement (what is being said).

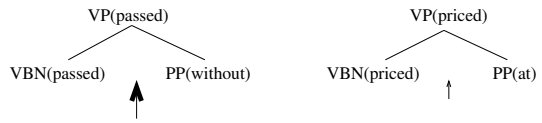

Figure 3: The same syntactic information indicates discourse boundaries depending on the lexical heads involved.

Given a sentence $s = w_1 w_2 \ldots w_i \ldots w_n$, we first find the syntactic parse tree t of s. We used in our experiments both syntactic parse trees obtained using Charniak's parser (2000) and syntactic parse trees from the PennTree bank. Our statistical model assigns a segmenting probability $P(b_i|w_i, t)$ for each word w_i, where $b_i \in \{$`boundary, no-boundary`$\}$. Because our model is concerned with discourse segmentation at sentence level, we define $P(\texttt{boundary}|w_n, t) = 1$, i.e., the sentence boundary is always a discourse boundary as well.

Our model uses both lexical and syntactic features for determining the probability of inserting discourse boundaries. We apply canonical lexical head projection rules (Magerman, 1995) in order to lexicalize syntactic trees. For each word w, the upper-most node with lexical head w which has a right sibling node determines the features on the basis of which we decide whether to insert a discourse boundary. We denote such node N_w, and the features we use are node N_w, its parent N_p, and the siblings of N_w. In the example in Figure 2, we determine whether to insert a discourse boundary after the word `says` using as features node $N_p = \texttt{VP(says)}$ and its children $N_w = \texttt{VBZ(says)}$ and $N_r = \texttt{SBAR(will)}$. We use our corpus to estimate the likelihood of inserting a discourse boundary between word w and the next word using formula (1),

$$P(b|w,t) \simeq \frac{Cnt(N_p \rightarrow \ldots N_w \uparrow N_r \ldots)}{Cnt(N_p \rightarrow \ldots N_w N_r \ldots)} \qquad (1)$$

where the numerator represents all the counts of the rule $N_p \rightarrow \ldots N_w N_r \ldots$ for which a discourse boundary has been inserted after word w, and the denominator represents all the counts of the rule.

Because we want to account for boundaries that are motivated lexically as well, the counts used in formula (1) are defined over lexicalized rules. Without lexicalization, the syntactic context alone is too general and fails to distinguish genuine cases of discourse boundaries from incorrect ones. As can be seen in Figure 3, the same syntactic context may indicate a discourse boundary when the lexical heads `passed` and `without` are present, but it may not indicate a boundary when the lexical heads `priced` and `at` are present.

The discourse segmentation model uses the corpus presented in Section 2 in order to estimate probabilities for inserting discourse boundaries using equation (1). We also use a simple interpolation method for smoothing lexicalized rules to accommodate data sparseness.

Once we have the segmenting probabilities given by the statistical model, a straightforward algorithm is used to implement the *segmenter*. Given a syntactic tree t, the algorithm inserts a boundary after each word w for which $P(\texttt{boundary}|w, t) > 0.5$.

4 The Discourse Parser

In the setting presented here, the input to the discourse parser is a Discourse Segmented Lexicalized Syntactic Tree (i.e., a lexicalized syntactic parse tree in which the discourse boundaries have been identified), henceforth called a DS-LST. An example of a DS-LST in the tree in Figure 2. The output of the discourse parser is a discourse parse tree, such as the one presented in Figure 1.

As in other statistical approaches, we identify two components that perform the discourse parsing task. The first component is the *parsing model*, which assigns a probability to every potential candidate parse tree. Formally, given a discourse tree DT and a set of parameters Θ, the parsing model estimates the conditional probability $P(DT|\Theta)$. The most likely parse is then given by formula (2).

$$DT_{best} = argmax_{DT}\ P(DT|\Theta) \qquad (2)$$

The second component is called the *discourse parser*, and it is an algorithm for finding DT_{best}. We first focus on the parsing model.

A discourse parse tree can be formally represented as a set of *tuples*. The discourse tree in Figure 1, for example, can be formally written as the set of tuples {ATTRIBUTION-SN[1,1,3], ENABLEMENT-NS[2,2,3]}. A tuple is of the form $R[i, m, j]$, and denotes a discourse relation R that holds between the discourse span that contains *edu*s i through m, and the discourse span that contains *edu*s $m+1$ through j. Each relation R also signals explicitly the nuclearity assignment, which can be NUCLEUS-SATELLITE (NS), SATELLITE-NUCLEUS (SN), or NUCLEUS-NUCLEUS (NN). This notation assumes that all relations R are binary relations. The assumption is justified empirically: 99% of the nodes of the discourse trees in our corpus are binary nodes. Using only binary relations makes our discourse model easier to build and reason with.

In what follows we make use of two functions: function rel applied to a tuple $R[i, m, j]$ yields the discourse relation R; function ds applied to a tuple $R[i, m, j]$ yields the structure $[i, m, j]$. Given a set of adequate parameters Θ, our discourse model estimates the goodness of a discourse parse tree DT using formula (3).

$$P(DT|\Theta) = \prod_{c \in DT} P_s(ds(c)|\Theta) \times P_r(rel(c)|\Theta) \qquad (3)$$

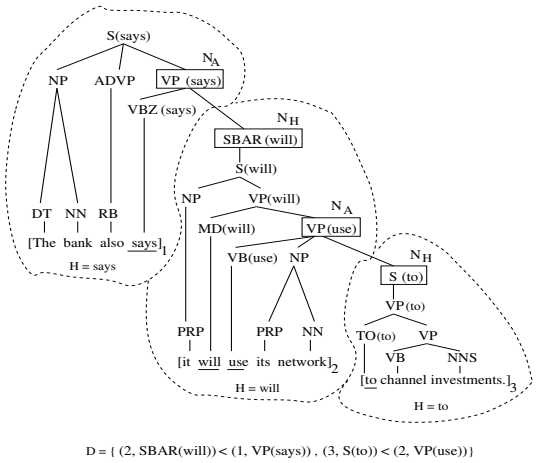

Figure 4: Dominance set extracted from a DS-LST.

For each tuple $c \in DT$, the probability P_s estimates the goodness of the structure of c. We expect these probabilities to prefer the hierarchical structure (1, (2, 3)) over ((1,2), 3) for the discourse tree in Figure 1. For each tuple $c \in DT$, the probability P_r estimates the goodness of the discourse relation of c. We expect these probabilities to prefer the rhetorical relation ATTRIBUTION-NS over CONTRAST-NN for the relation between spans 1 and $[2, 3]$ in the discourse tree in Figure 1. The overall probability of a discourse tree is obtained multiplying the structural probabilities P_s and the relational probabilities P_r for all the tuples in the discourse tree.

Our discourse model uses as Θ the information present in the input DS-LST. However, given such a tree ST as input, one cannot estimate probabilities such as $P(DT \mid ST)$ without running into a severe sparseness problem. To overcome this, we map the input DS-LST into a more abstract representation that contains only the salient features of the DS-LST. This mapping leads to the notion of a dominance set over a discourse segmented lexicalized syntactic tree. In what follows, we define this notion and show that it provides adequate parameterization for the discourse parsing problem.

4.1 The Dominance Set of a DS-LST

The dominance set of a DS-LST contains feature representations of a discourse segmented lexicalized syntactic tree. Each feature is a representation of the syntactic and lexical information that is found at the point where two *edu*s are joined together in a DS-LST. Our hypothesis is that such "attachment" points in the structure of a DS-LST (the boxed nodes in the tree in Figure 4) carry the most indicative information with respect to the potential discourse tree we want to build. A set representation of the "attachment" points of a DS-LST is called the dominance set of a DS-LST.

For each *edu* E we identify a word w in E as the *head word* of *edu* E and denote it H. H is defined as the word with the highest occurrence as a lexical head in the lexicalized tree among all the words in E. The node in which H occurs highest is called the *head node* of *edu* E and is denoted N_H. The *edu* which has as head node the root of the DS-LST is called the *exception edu*. In our example, the head word for *edu* 2 is $H = \texttt{will}$, and its head node is $N_H = \texttt{SBAR(will)}$; the head word for *edu* 3 is $H = \texttt{to}$, and its head node is $N_H = \texttt{S(to)}$. The exception *edu* is *edu* 1.

For each *edu* E which is not the exception *edu*, there exists a node which is the parent of the head node of E, and the lexical head of this node is guaranteed to belong to a different *edu* than E, call it F. We call this node the *attachment node* of E and denote it N_A. In our example, the attachment node of *edu* 2 is $N_A = \texttt{VP(says)}$, and its lexical head `says` belongs to *edu* 1; the attachment node of *edu* 3 is $N_A = \texttt{VP(use)}$, and its lexical head `use` belongs to *edu* 2. We write formally that two *edu*s E and F are linked through a head node N_H and an attachment node N_A as $(E, N_H) < (F, N_A)$.

The dominance set of a DS-LST is given by all the *edu* pairs linked through a head node and an attachment node in the DS-LST. Each element in the dominance set represents a *dominance relationship* between the *edu*s involved. Figure 4 shows the dominance set D for our example DS-LST. We say that *edu* 2 is dominated by *edu* 1 (shortly written $2 < 1$), and *edu* 3 is dominated by *edu* 2 ($3 < 2$).

4.2 The Discourse Model

Our discourse parsing model uses the dominance set D of a DS-LST as the conditioning parameter Θ in equation (3). The discourse parsing model we propose uses the dominance set D to compute the probability of a discourse parse tree DT according to formula (4).

$$P(DT|D) = \prod_{c \in DT} P_s(ds(c)|filter_s(c, D)) \times P_r(rel(c)|filter_r(c, D))) \quad (4)$$

Different projections of D are used to accurately estimate the structure probabilities P_s and the relation probabilities P_r associated with a tuple in a discourse tree. The projection functions $filter_s$ and $filter_r$ ensure that, for each tuple $c \in DT$, only the information in D relevant to c is to be conditioned upon. In the case of P_s (the probability of the structure $[i, m, j]$), we filter out the lexical heads and keep only the syntactic labels; also, we filter out all the elements of D which do not have at least one *edu* inside the span of c. In our running example, for instance, for $c =$ ENABLEMENT-NS$[2, 2, 3]$, $filter_s(c, D) = \{(2, SBAR) < (1, VP), (3, S) < (2, VP)\}$. The span of c is $[2, 3]$, and set D has two elements involving *edu*s

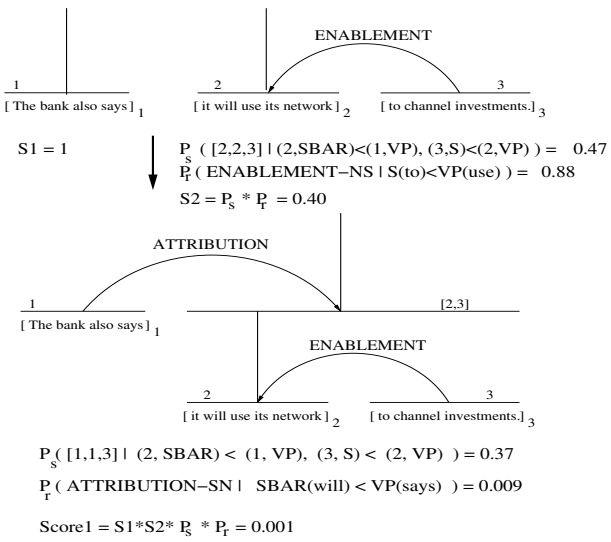

Figure 5: Bottom-up discourse parsing.

from it, namely the dominance relationships $2 < 1$ and $3 < 2$. To decide the appropriate structure, $filter_s$ keeps them both; this is because a different dominance relationship between *edu*s 1 and 2, namely $1 < 2$, would most likely influence the structure probability of c.

In the case of P_r (the probability of the relation R), we keep both the lexical heads and the syntactic labels, but filter out the *edu* identifiers (clearly, the relation between two spans does not depend on the positions of the spans involved); also, we filter out all the elements of D whose dominance relationship does not hold across the two sub-spans of c. In our running example, for $c =$ ENABLEMENT-NS$[2, 2, 3]$, $filter_r(c, D) = \{S(to) < VP(use)\}$. The two sub-spans of c are $[2, 2]$ and $[3, 3]$, and only the dominance relationship $3 < 2$ holds across these spans; the other dominance relationship in D, $2 < 1$, does not influence the choice for the relation label of c.

The conditional probabilities involved in equation (4) are estimated from the training corpus using maximum likelihood estimation. A simple interpolation method is used for smoothing to accommodate data sparseness. The counts for the dependency sets are also smoothed using symbolic names for the *edu* identifiers and accounting only for the distance between them.

4.3 The Discourse Parser

Our *discourse parser* implements a classical bottom-up algorithm. The parser searches through the space of all legal discourse parse trees and uses a dynamic programming algorithm. If two constituents are derived for the same discourse span, then the constituent for which the model assigns a lower probability can be safely discarded.

Figure 5 shows a discourse structure created in a bottom-up manner for the DS-LST in Figure 2. Tuple ENABLEMENT-NS[2,2,3] has a score of 0.40, obtained as the product between the structure probability P_s of 0.47 and the relation probability P_r of 0.88. Tuple ATTRIBUTION-SN[1,1,3] has a score of 0.37 for the structure, and a score of 0.009 for the relation. The final score for the entire discourse structure is 0.001. All probabilities used were estimated from our training corpus. According to our discourse model, the discourse structure in Figure 5 is the most likely among all the legal discourse structures for our example sentence.

5 Evaluation

In this section we present the evaluations carried out for both the discourse segmentation task and the discourse parsing task. For this evaluation, we re-trained Charniak's parser (2000) such that the test sentences from the discourse corpus were not seen by the syntactic parser during training.

5.1 Evaluation of the Discourse Segmenter

We train our discourse segmenter on the Training section of the corpus described in Section 2, and test it on the Test section. The training regime uses syntactic trees from the Penn Treebank. The metric we use to evaluate the discourse segmenter records the accuracy of the discourse segmenter with respect to its ability to insert inside-sentence discourse boundaries. That is, if a sentence has 3 *edu*s, which correspond to 2 inside-sentence discourse boundaries, we measure the ability of our algorithm to correctly identify these 2 boundaries. We report our evaluation results using recall, precision, and F-score figures. This metric is harsher than the metric previously used by Marcu (2000), who assesses the performance of a discourse segmentation algorithm by counting how often the algorithm makes `boundary` and `no-boundary` decisions for *every* word in a sentence.

We compare the performance of our probabilistic discourse segmenter with the performance of the decision-based segmenter proposed by (Marcu, 2000) and the performance of two baseline algorithms. The first baseline ($B1DS$) uses punctuation to determine when to insert a boundary; because commas are often used to indicate breaks inside long sentences, $B1DS$ inserts discourse boundaries after each comma. The second baseline ($B2DS$) uses syntactic information; because long sentences often have embedded sentences, $B2DS$ inserts discourse boundaries after each text span whose corresponding syntactic subtree is labeled `S`, `SBAR`, or `SINV`. We also compute the agreement between human annotators on the discourse segmentation task (HDS), using the doubly-annotated discourse corpus mentioned in Section 2.

	Recall	Precision	F-score
$B1DS$	28.2	37.1	32.0
$B2DS$	25.4	64.9	36.5
$DecDS$	77.1	83.3	80.1
$SynDS(T^-)$	**82.7**	**83.5**	**83.1**
$SynDS(T^+)$	**85.4**	**84.1**	**84.7**
HDS	98.2	98.5	98.3

Table 1: Discourse segmenter evaluation

Table 1 shows the results obtained by the algorithm described in this paper ($SynDS(T^-)$) using syntactic trees produced by Charniak's parser (2000), in comparison with the results obtained by the algorithm described in (Marcu, 2000) ($DecDS$), and baseline algorithms $B1DS$ and $B2DS$, on the same test set. Crucial to the performance of the discourse segmenter is the recall figure, because we want to find as many discourse boundaries as possible. The baseline algorithms are too simplistic to yield good results (recall figures of 28.2% and 25.4%). The algorithm presented in this paper gives an error reduction in missed discourse boundaries of 24.5% (recall accuracy improvement from 77.1% to 82.7%) over (Marcu, 2000). The overall error reduction is of 15.1% (improvement in F-score from 80.1% to 83.1%).

In order to asses the impact on the performance of the discourse segmenter due to incorrect syntactic parse trees, we also carry an evaluation using syntactic trees from the Penn Treebank. The results are shown in row $SynDS(T^+)$. Perfect syntactic trees lead to a further error reduction of 9.5% (F-score improvement from 83.1% to 84.7%). The performance ceiling for discourse segmentation is given by the human annotation agreement F-score of 98.3%.

5.2 Evaluation of the Discourse Parser

We train our discourse parsing model on the Training section of the corpus described in Section 2, and test it on the Test section. The training regime uses syntactic trees from the Penn Treebank. The performance is assessed using labeled recall and labeled precision as defined by the standard Parseval metric (Black et al., 1991). As mentioned in Section 2, we use both 18 labels and 110 labels for the discourse relations. The recall and precision figures are combined into an F-score figure in the usual manner.

The discourse parsing model uses syntactic trees produced by Charniak's parser (2000) and discourse segments produced by the algorithm described in Section 3. We compare the performance of our model ($SynDP$) with the performance of the decision-based discourse parsing model ($DecDP$) proposed by (Marcu, 2000), and with the performance of a baseline algorithm (BDP). The baseline algorithm builds right-branching discourse trees labeled with the most frequent relation encountered in the training set (i.e., ELABORATION-NS). We also compute the agreement between human annotators on the discourse parsing task (HDP), using the doubly-annotated discourse corpus mentioned in Section 2. The results are shown in Table 2. The baseline algorithm has a performance of 23.4% and 20.7% F-score, when using 18 labels and 110 labels, respectively. Our algorithm has a performance of 49.0% and 45.6% F-score, when using 18 labels and 110 labels, respectively. These results represent an error reduction of 18.8% (F-score improvement from 37.2% to 49.0%) over a state-of-the-art discourse parser (Marcu, 2000) when using 18 labels, and an error reduction of 15.7% (F-score improvement from 35.5% to 45.6%) when using 110 labels. The performance ceiling for sentence-level discourse structure derivation is given by the human annotation agreement F-score of 77.0% and 71.9%, when using 18 labels and 110 labels, respectively. The performance gap between the results of $SynDP$ and human agreement is still large, and it can be attributed to three possible causes: errors made by the syntactic parser, errors made by the discourse segmenter, and the weakness of our discourse model.

	BDP	$DecDP$	$SynDP$	HDP
Unlabeled	64.0	67.0	**70.5**	92.8
18 Labels	23.4	37.2	**49.0**	77.0
110 Labels	20.7	35.5	**45.6**	71.9

Table 2: $SynDP$ performance compared to baseline, state-of-the-art, and human performance

	T^-S^-	T^+S^-	T^-S^+	T^+S^+
Unlabeled	70.5	73.0	92.8	**96.2**
18 Labels	49.0	56.4	63.8	**75.5**
110 Labels	45.6	52.6	59.5	**70.3**

Table 3: $SynDP$ performance with human-level accuracy for syntactic trees and discourse boundaries.

In order to quantitatively asses the impact in performance of each possible cause of error, we perform further experiments. We replace the syntactic parse trees produced by Charniak's parser at 90% accuracy (T^-) with the corresponding Penn Treebank syntactic parse trees produced by human annotators (T^+). We also replace the discourse boundaries produced by our discourse segmenter at 83% accuracy (S^-) with the discourse boundaries taken from (RST-DT, 2002), which are produced by the human annotators (S^+).

The results are shown in Table 3. The results in column T^+S^- show that using perfect syntactic trees leads

to an error reduction of 14.5% (F-score improvement from 49.0% to 56.4%) when using 18 labels, and an error reduction of 12.9% (F-score improvement from 45.6% to 52.6%) when using 110 labels. The results in column T^-S^+ show that the impact of perfect discourse segmentation is double the impact of perfect syntactic trees. Human-level performance on discourse segmentation leads to an error reduction of 29.0% (F-score improvement from 49.0% to 63.8%) when using 18 labels, and an error reduction of 25.6% (F-score improvement from 45.6% to 59.5%) when using 110 labels. Together, perfect syntactic trees and perfect discourse segmentation lead to an error reduction of 52.0% (F-score improvement from 49.0% to 75.5%) when using 18 labels, and an error reduction of 45.5% (F-score improvement from 45.6% to 70.3%) when using 110 labels. The results in column T^+S^+ in Table 3 compare extremely favorable with the results in column HDP in Table 2. The discourse parsing model produces unlabeled discourse structure at a performance level similar to human annotators (F-score of 96.2%). When using 18 labels, the distance between our discourse parsing model performance level and human annotators performance level is of absolute 1.5% (75.5% versus 77%). When using 110 labels, the distance is of absolute 1.6% (70.3% versus 71.9%). Our evaluation shows that our discourse model is sophisticated enough to match near-human levels of performance.

6 Conclusion

In this paper, we have introduced a discourse parsing model that uses syntactic and lexical features to estimate the adequacy of sentence-level discourse structures. Our model defines and exploits a set of syntactically motivated lexico-grammatical dominance relations that fall naturally from a syntactic representation of sentences.

The most interesting finding is that these dominance relations encode sufficient information to enable the derivation of discourse structures that are almost indistinguishable from those built by human annotators. Our experiments empirically show that, at the sentence level, there is an extremely strong correlation between syntax and discourse. This is even more remarkable given that the discourse corpus (RST-DT, 2002) was built with no syntactic theory in mind. The annotators used by Carlson et al. (2003) were not instructed to build discourse trees that were consistent with the syntax of the sentences. Yet, they built discourse structures at sentence level that are not only consistent with the syntactic structures of sentences, but also derivable from them.

Recent work on Tree Adjoining Grammar-based lexicalized models of discourse (Forbes et al., 2001) has already shown how to exploit within a single framework lexical, syntactic, and discourse cues. Various linguistics studies have also shown how intertwined syntax and discourse are (Maynard, 1998). However, to our knowledge, this is the first paper that empirically shows that the connection between syntax and discourse can be computationally exploited at high levels of accuracy on open domain, newspaper text.

Another interesting finding is that the performance of current state-of-the-art syntactic parsers (Charniak, 2000) is not a bottleneck for coming up with a good solution to the sentence-level discourse parsing problem. Little improvement comes from using manually built syntactic parse trees instead of automatically derived trees. However, experiments show that there is much to be gained if better discourse segmentation algorithms are found; 83% accuracy on this task is not sufficient for building highly accurate discourse trees.

We believe that semantic/discourse segmentation is a notoriously under-researched problem. For example, Gildea and Jurafsky (2002) present a semantic parser that optimistically assumes that has access to perfect semantic segments. Our results suggest that more effort needs to be put on semantic/discourse-based segmentation. Improvements in this area will have a significant impact on both semantic and discourse parsing.

References

E. Black, S. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings of Speech and Natural Language Workshop*, pages 306–311, Pacific Groove, CA. DARPA.

L. Carlson, D. Marcu, and M. E. Okurowski. 2003. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In Jan van Kuppevelt and Ronnie Smith, editors, *Current Directions in Discourse and Dialogue*. Kluwer Academic Publishers. To appear.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the NAACL 2000*, pages 132–139, Seattle, Washington, April 29 – May 3.

Michael Collins. 2000. Discriminative reranking for natural language parsing. In *Proceedings of ICML 2000*, Stanford University, Palo Alto, CA, June 29–July 2.

C. J. Fillmore, C. F. Baker, and S. Hiroaki. 2002. The framenet database and software tools. In *Proceedings of the LREC 2002*, pages 1157–1160, LREC.

K. Forbes, E. Miltsakaki, R. Prasad, A. Sarkar, A. Joshi, and B. Webber. 2001. D-LTAG System: Discourse parsing with a lexicalized tree-adjoining grammar. In

ESSLLI'2001 Workshop on Information Structure, Discourse Structure and Discourse Semantics.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic role. *Computational Linguistics*, 28(3):245–288.

Paul Kingsbury and Martha Palmer. 2002. From Treebank to Propbank. In *Proceedings of the LREC 2002*, Las Palmas, Canary Islands, Spain, May 28-June 3.

David M. Magerman. 1995. Statistical decision-tree models for parsing. In *Proceedings of the ACL 1995*, pages 276–283, Cambridge, Massachusetts, June 26-30.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Daniel Marcu. 2000. *The Theory and Practice of Discourse Parsing and Summarization*. The MIT Press, Cambridge, Massachusetts.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Senko K. Maynard. 1998. *Principles of Japanese Discourse: A Handbook*. Cambridge University Press.

David D. Palmer and Marti A. Hearst. 1997. Adaptive multilingual sentence boundary disambiguation. *Computational Linguistics*, 23(2):241–269, June.

Adwait Ratnaparkhi. 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania.

RST-DT. 2002. RST Discourse Treebank. Linguistic Data Consortium. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002T07.