Automatically Identifying the Arguments of Discourse Connectives

Ben Wellner[†*]
[*]The MITRE Corporation
202 Burlington Road
Bedford, MA USA

James Pustejovsky[†]
[†]Department of Computer Science
Brandeis University
Waltham, MA USA

Abstract

In this paper we consider the problem of automatically identifying the arguments of discourse connectives (e.g., *and*, *because*, *nevertheless*) in the Penn Discourse TreeBank(PDTB). Rather than identifying the full *extents* of these arguments as annotated in the PDTB, however, we re-cast the problem to that of identifying the argument *heads*, effectively side-stepping the problem of *discourse segmentation*. We demonstrate significant gains using features derived from a dependency parse representation over those derived from a constituent-based tree parse. By also capturing inter-argument dependencies using a log-linear re-ranking model we identify both arguments correctly for over 74% of the connectives on held-out test data using gold-standard parses.

1 Introduction

The study of discourse is concerned with analyzing how phrase, clause or sentence-level units of text are *related* to each other within a larger unit of text (e.g., a document). Long recognized as important in dialog and text generation, this level of analysis is important generally for applications needing to place events and propositions in their proper context such as scenario-level information extraction, question answering, summarization, sentiment analysis and others.

In line with much of the NLP research agenda, recently a number of annotated corpora have emerged which encode discourse-level phenomena, making it possible to apply supervised, empirically-driven techniques to identifying discourse relations. Such corpora include the RST Discourse Treebank (Carlson et al., 2003) (based on Rhetorical Structure Theory), the Discourse GraphBank (Wolf and Gibson, 2005) (based on the relations of Hobbs (1985)) and the Penn Discourse Treebank (Miltsakaki et al., 2004b). While these corpora differ in many ways, they all more or less encode problems involving: 1) identifying/segmenting the basic units of discourse (e.g., clauses, phrases), 2) determining for which pairs of segments (or segment groups) a discourse relation exists, and 3) characterizing the *type* of relation (cause, elaboration, etc.) between segment pairs.

For our experiments in this paper, we use the Penn Discourse TreeBank (PDTB). The PDTB differs from most other discourse-level annotation efforts in its bottom-up, lexically-driven approach. Rather than identifying all possible discourse relations, the PDTB focuses on annotating relations lexicalized by discourse connectives that explicitly occur in the text along with their two arguments. [1] These discourse connectives include coordinating conjunctions (e.g., *and*, *or*), subordinating conjunctions (e.g., *because*, *when*, *since*) and discourse adverbials (e.g., *however*, *previously*, *nevertheless*).

In this paper we focus on problems (1) and (2)

[1] The final release of the PDTB, scheduled for release in August 2007, will annotate the *type* of the rhetorical relation holding between arguments of explicit connectives in addition to annotating relations between adjacent sentences where no lexical connective is present.

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 92–101, Prague, June 2007. ©2007 Association for Computational Linguistics

above. However, rather than explicitly identifying the discourse segments and then deciding for which pairs a relation exists, we focus on identifying relations between the pairs of *head words* that *represent* the discourse segments. In this sense, the problem resembles that of predicate-argument identification where the predicates are discourse connectives and the arguments are single words which serve as anchors for the discourse segments.

To address the problem of identifying the arguments of discourse connectives we incorporate a variety of lexical and syntactic features in a discriminative log-linear ranking model. To capture dependencies between the two arguments of a connective we use a log-linear *re*-ranking model to select the best argument pair from a set of N-best argument pairs provided by the independent argument models. Further, we provide an analysis of the contribution of the various features demonstrating that features based on a dependency parse representation outperform features derived from a constituent tree parse.

2 Overview of the Penn Discourse Treebank

Discourse arguments in the PDTB represent abstract objects (Asher, 1993) which include facts, propositions and events. Each argument must include at least one predicate and can be realized as: a clause, a VP within VP coordination, a nominalization (in certain, restricted cases), an anaphoric expression or a response to a question. Each connective has two arguments: Arg2 is the argument syntactically connected to the connective in the same sentence and Arg1 is the other argument which may lie in the same sentence as the connective or, generally, anywhere prior in the discourse.

The PDTB contains a total of 18505 explicit connectives annotated with discourse arguments. The annotations are layered on top of the Penn TreeBank-II (PTB) parse trees and cover all 25 Wall Street Journal (WSJ) sections.

2.1 Examples

Below are a few examples from the PDTB. Each Arg1 is denoted in *italics* and each Arg2 is denoted in **bold**. The head-words for each argument are underlined. We discuss and motivate the identification of head-words in Section 2.2.

(1) *Choose 203 business executives, including, perhaps, someone from your own staff,* and **put them out on the streets**, to be deprived for one month of their homes, families and income.

(1) shows an example of a coordinating connective *and* and its two arguments. In this case, the Arg1 lies in the same sentence as the connective. It is also possible for the Arg1 to lie outside the sentence (usually in the immediately preceding sentence) when the coordinating connective begins a sentence.

An example of the subordinating connective, *because* is shown below in (2). This example brings up some interesting ambiguities that arise quite regularly in the data. An alternative reading for this example might only include the extent *to duck liability* for the Arg1. That is, the predicate *be able* could be read to include the discourse relation and its two arguments as an argument.

(2) *Drug makers shouldn't be able to duck liability* because **people couldn't identify precisely which identical drug was used.**

Both coordinating and subordinating connectives are *structural* (Webber et al., 2003). Discourse adverbials however, take one argument, Arg2, structurally but the other can be anaphoric: its Arg1 may be present anywhere in the current running discourse with little or no restriction. Example (3) shows the case in which the Arg1 lies in the previous sentence. In many cases, however, it resides in the same sentence as the connective or many sentences prior in the discourse.

(3) *France's second-largest government-owned insurance company, Assurances Generales de France, has been building its own Nagivation Mixte stake*, currently thought to be between 8% and 10%. Analysts said

they don't think it is contemplating a takeover, however, and its officials couldn't be reached.

2.2 Head-Based Representation of the PDTB

In contrast to other annotations layered on the PTB such as PropBank and NomBank, the arguments of a discourse connective generally do not correspond to a single parse tree constituent. Arguments consist instead of a *set* of non-overlapping constituents from the parse tree (i.e. a forest). This target representation makes the process of identifying the arguments to discourse connectives difficult since the space of candidate arguments extents is considerably larger than for PropBank parsing, for example. Even without this added difficulty, discourse segmentation is one of the most difficult stages in discourse parsing (Soricut and Marcu, 2003). While the segments themselves may be useful in certain contexts, for many applications, if not most, it will still be necessary to *interpret* these segments (e.g. at the predicate-argument level). As such, we argue that, in general, identifying the lexical *heads* of these discourse segments is sufficient and perhaps even preferable for this stage of processing. A problem arises, however, with arguments that consist of sequences of abstract objects represented as coordinated or subordinated sequences of VPs, clauses or sentences. What should the head be in such cases? By convention we designate the extent head as the head of the first element in the sequence. In (4), the head of the Arg2 would be *went*, but it's implicit scope includes the second VP coordinate headed by *caught*.

(4) Mr. Dozen even related *the indignity suffered* when **he and two colleagues went on an overnight fishing expedition of the New Jersey shore and caught nothing.**

The problem then becomes how to determine the end of the sequence of abstract objects. In many cases, there is a "natural end" to such sequences based on the syntax. In (4), the natural end is simply the end of the VP coordination. Difficult cases remain, however, particularly with multi-sentential Arg1s of anaphoric connectives. Determining the end of the these arguments seems non-trivial.[2] Nevertheless, identifying the begininning of the argument (via its head) is an important step in modeling these difficult cases.

2.3 Head Identification

Identifying the head of a discourse argument given its extent (as described by a set of constituent sub-trees in the PTB) consists of two steps. First, we construct a single syntactic tree formed by taking all of the sub-trees in the extent, finding their least common ancestor (LCA) node and including all intermediate nodes from the subtrees to the LCA node. Then, a slight variation of the head finding algorithm in (Collins, 1999) is applied to the derived tree to find the head. Figure 1 provides an example indicating the arguments to the connective "After" and the derived argument heads.

3 Discourse Argument Identification

Identifying the arguments of discourse connectives can be naturally formulated as a binary classification task where separate classifiers are trained for each argument — i.e., Arg1 and Arg2. First, a set of candidate arguments, α_i is gathered for each connective, π. Training instances, $\langle \alpha_i, \pi \rangle$, are then created for each candidate with respect to the connective. A training instance is positive if α_i is the true argument for π and negative otherwise. At decoding time, the candidate classified positively with the highest probability (or score) compared to the other candidates is selected as the argument.

An alternative to using a standard classification approach is to use a *ranking* model. The advantage of the ranking model is that candidate instances are compared against each other *during training* as well as during decoding. In

[2] There are indications, however, that the end of the argument sometimes falls out of the (possibly non-lexicalized) discourse relations local to the argument.

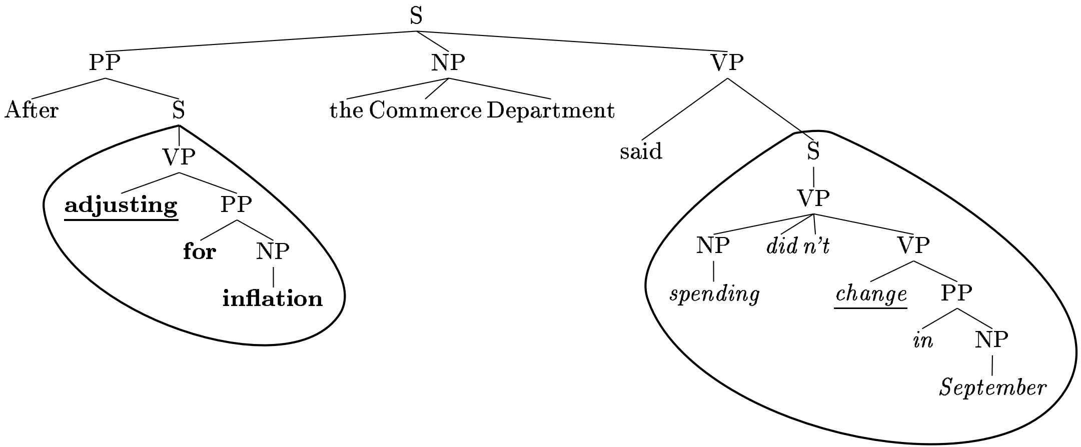

Figure 1: Syntactic structure and discourse arguments for the connective "After".

contrast, with a standard classifier, separate instances (i.e. candidates) are trained and classified as if they were completely independent. We use a log-linear ranking model. Such models have been used for a variety of other tasks including co-reference (Denis and Baldridge, 2007), question answering (Ravichandran et al., 2003) and parse re-ranking (Charniak and Johnson, 2005). For a given Arg1 candidate, α_i, the probability of that candidate being the argument given the connective, π, and the document, x, is defined according to the model as:

$$P_1(\alpha_i|\pi, x) = \frac{\exp\left(\sum_k \lambda_k f_k(\alpha_i, \pi, x)\right)}{\sum\limits_{\alpha_j \in C_1(\pi, x)} \exp\left(\sum_k \lambda_k f_k(\alpha_j, \pi, x)\right)} \tag{1}$$

where the f_k are feature functions, the λ_k are their weights and $C_1(\pi, x)$ is the set of candidate Arg1 arguments for the connective π in the document x. The model for Arg2 is defined analogously, but may in fact use a different set of features or a different candidate generation function. At training time, all potential candidates of a particular type for a given connective are provided to the ranking model as a distribution: the correct gold-standard candidate receiving a probability mass of 1.0 and the other candidates receiving masses of 0.0. During decoding, we select candidates in the same way as for training and produce a distribution over these candidates according to equation 1, selecting the candidate assigned the highest probability by the model as the argument.

We compared both the above ranking model and a standard binary Maximum Entropy model (i.e., logistic regression) and found the ranking model to have a small but consistent edge over the classifier. Accordingly, we only report results here using the ranking model.

3.1 Candidate Selection

Selecting the candidate arguments, α_i, is an important aspect of the problem. There are conceivably very many possible Arg1 candidates for a given connective stretching back from the sentence containing the connective to the beginning of the document. We employ two simple criteria to reduce the space of candidate argument head words. First, we only consider argument candidates that have an appropriate part-of-speech (all verbs, common nouns, adjectives). Second, we only consider candidates that are within 10 "steps" of the connective where a single step includes a sentence boundary or a syntactic dependency link within a sentence (see Figure 2). Only candidates lying within the same sentence as the connective are considered for Arg2.

3.2 Features

We used a variety of features for identifying the discourse arguments of a connective.

Baseline Features. Our baseline features included simply the connective and argument words, where the connective appears in the sentence, whether the argument precedes or follows the connective and whether the argument is in the same sentence as the connective or not.

Constituent Path Features. As noted in work on semantic role labeling, features derived from the constituent parse of the sentence can be very helpful for deriving the argument structure of predicating verbs (Toutanova et al., 2005) and nouns (Jiang and Ng, 2006). Syntax plays a strong role in identifying discourse arguments, too, though even for structural connectives it by no means "aligns" with the discourse structure (Dinesh et al., 2005). We introduced a feature capturing the constituent tree path from the connective to the candidate argument as well as variants in which repeated nodes and part-of-speech nodes are removed from the path. If the argument lies in a different sentence, the path from the connective to the argument consists of the path from the connective to the top node of its sentence, followed by a series of virtual *SENT* nodes for the intervening sentences and then ending with the path from the top node of the sentence containing the argument to the argument head itself.

Dependency Path Features. We experimented with a number of syntactic features based on a *dependency* parse representation. The primary motivation here being that it provides for a more compact and natural representation of the syntax, providing for better syntactic features with less data sparseness than constituent path features. The dependency representation we use is that put forth in de Marneffe et al. (2006) and we apply their approach to deriving the dependency structure from the constituent parse. The features used here include the (shortest) dependency path from the connective to the prospective argument and two collapsed versions removing coordination links as well as repeated links of the same type. For argument candidates in prior sentences, we introduce *SENT* links for each intervening sentence.

Connective Features. Different discourse connectives behave differently depending on their type. A potentially important feature then involves capturing the connective type: (coordinating, subordinating or adverbial). We use the categorized lists of discourse connectives found in (Knott, 1996); further, any connectives not appearing in these lists are considered discourse adverbials. As we would expect different syntax associated with different connectives we introduce conjunctive features such as the connective type and syntactic path.

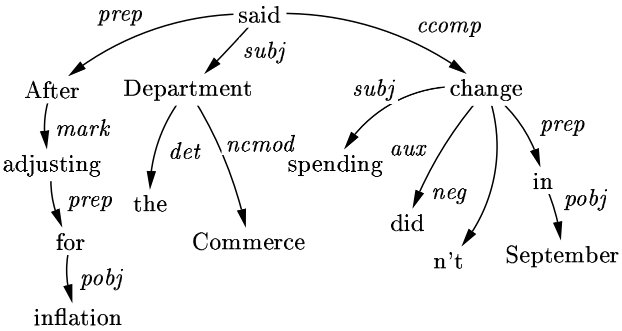

Figure 2: Dependency structure.

Lexico-Syntactic Features. One of the prime difficulties in identifying the correct non-anaphoric argument has to do with *attribution.* In this situation the argument is the complement of a verb indicating attribution of the proposition denoted by the complement to an individual other than the writer. Figure 1 provides an example of this where the ARG1 of "After" is the complement of the verb "said" being attributed to "the Commerce Department". To model this situation we introduce features capturing whether the argument is a potentially attribution-denoting verb, whether it has a clausal complement, whether it is the clausal complement of another verb and whether the complementing verb is attributing.

A full listing of the features used for identifying arguments is shown in Table 1.

4 Experiments with Independent Argument Identification

For all of our experiments, we use sections 02-22 for training, sections 00-01 for development and sections 23-24 for testing. The development data was used to customize our features and to tune the Gaussian prior used to prevent over-

Baseline Features	
A	Where in the sentence (beginning, middle, end) the connective resides
B	Whether the argument is in the same sentence as the connective (yes,no)
C	Connective phrase
D	Downcase connective phrase
E	Argument head word
F	Argument head prior or after connective
G	A & B
Constitutent Features	
H	Path from argument to connective through the constituent tree
I	Length of path
J	Collapsed path without part-of-speech
K	Collapsed path removing repetitions of the same node type (e.g. VP-VP-VP → VP)
L	C & H
Dependency Features	
M	Dependency path from argument to connective
N	Path + head word of first link from connective
O	Collapsed path removing coordinating links
P	Collapsed path removing repetitions of links
Q	C & M
Connective Features	
R	coordinating, subordinating or adverbial connective
S	A & R
T	M & R
Lexico-Syntactic Features	
U	Argument is an attributing verb
V	Argument has a clausal complement
W	U & V
X	Argument is a clausal complement of a verb
Y	X & governing verb is an attributing verb

Table 1: Feature types for discourse connective argument identification

fitting in the log-linear models (at $\sigma = 0.25$ for both the local and the re-ranking models). All results are reported on the testing data, sections 23-24. We report results using both gold-standard parses and automatic parses using the Charniak-Johnson parser (Charniak and Johnson, 2005).

For evaluating Arg1 and Arg2 argument identification performance we report *accuracy* — i.e., the percentage of arguments correctly identified. An argument is correct if and only if it is the same head-word as derived from the argument extent as annotated in the PDTB (as described in Section 2.3). We also report *Connective Accuracy* which is the percentage of connectives for which *both* arguments were correctly identified.

	Accuracy		
FeatureSet	Arg1	Arg2	Conn.
A-G	32.7	60.7	21.6
A-L	60.6	85.5	53.6
A-G;M-Q	73.7	94.2	70.2
A-Y	75.0	94.2	71.7
A-Y(auto)	67.9	90.6	62.7

Table 2: Results for argument identification on the testing data (WSJ sections 23-24) gold standard parses (with various feature sets) and Charniak-Johnson parses (auto) for the full feature set A-Y.

Our results for the task of identifying arguments are shown in Table 4 for various feature combinations. It is interesting to compare the performance of the constituent parse features (A-L) vs. the dependency parse features (A-G;M-Q). The dependency parse features perform markedly better: 70.2 vs. 53.6 Connective Accuracy with gold-standard parses.

5 Experiments With Re-ranking

A drawback to the above approach is that the two arguments are identified independently. Ideally, one would like to consider both arguments and the connective simultaneously, taking into account global properties such as the pattern of the argument structure (e.g. Connective-Arg2-Arg1 vs. Arg1 Connective Arg2) or properties of compatibility between the two arguments (e.g. agreement in tense). Considering all pairs of arguments outright, however, presents scalability issues as the number of such pairs can be very large (especially with anaphoric Arg1s). Indeed, a huge advantage of the lexicalized approach taken with the PDTB is that we *can* identify arguments independently using the connectives as anchors. Nevertheless, there is obvious potential gain from modeling pairs of arguments jointly.

One way to model these dependencies in a tractable fashion is to use a *re-ranking* approache (Collins, 2000) which has proven successful in a variety of NLP tasks. The basic idea is to use a model with strong independenc as-

	Accuracy		
N	Arg1	Arg2	Conn.
1	74.5	94.5	71.4
5	83.1	97.4	81.8
10	90.5	97.9	89.2
20	93.8	97.9	92.1
30	94.6	97.9	92.9

Table 3: N-best upper-bounds for different values of N according to a product of independent argument ranker probabilities with the full feature set (A-Y)

sumption, $GEN(\pi)$, in this case based on the independent argument models described above, to generate N candidate argument pairs for a given connective, π. Then, the re-ranking model is used to re-rank these candidate pairs; the top-ranked pair is then selected.

In our setting for a given connective, π, we define the *local probability* for a candidate argument pair, $\langle \alpha_i, \alpha_j \rangle$ as:

$$P_{loc}(\alpha_i, \alpha_j|\pi, x) = P_{\text{ARG1}}(\alpha_i|\pi, x) \cdot P_{\text{ARG2}}(\alpha_j|\pi, x)$$

Thus, $GEN(\pi)$ generates the top N argument pairs according to the P_{loc}. In practice, we also assert that $P_{loc}(\alpha_j, \alpha_k|\pi, x) = 0$ when $j = k$.

For different values of N, Table 3 shows the oracle upper bounds on performance - the performance achieved by selecting the correct argument pair from $GEN(\pi)$ if it is in the list of argument pairs and otherwise selecting the first pair with one correct argument if such a pair exists. Note that performance on Arg2 plateaus at 97.9. This is due to 2.1 percent of the Arg2s not being reachable because they are not considered candidates (they are more than 10 "parse steps" away or an invalid part-of-speech).

5.1 Modeling Inter-Argument Dependencies

The model for re-ranking pairs of arguments is given by

$$P_r(\alpha_i, \alpha_j|\pi, x) = \frac{\exp\left(\sum_k \lambda_k f_k(\alpha_i, \alpha_j, \pi, x)\right)}{\sum_{\alpha_i, \alpha_j \in GEN(\pi)} \exp\left(\sum_k \lambda_k f_k(\alpha_i, \alpha_j, \pi, x)\right)}$$

Following previous work (Collins, 2000; Toutanova et al., 2005), we mix the local model into the final score along with the re-ranking model as:

$$P(\alpha_i, \alpha_j|\pi, x) = P_{loc}(\alpha_i, \alpha_j|\pi, x)^{\gamma} \cdot P_r(\alpha_i, \alpha_j|\pi, x)$$

where γ indicates the degree to which the local model influences the final score. Tuning γ on the development data, we set $\gamma = 0.4$ for all our re-ranking experiments.

The re-ranking model is able to accommodate features over *both* candidate arguments. For example, we can test whether the two arguments are the same predicate or whether they are both reporting verbs. Another set of features consists of triples denoting the relative order of the arguments and the connective. For example, the feature *CONN*_Arg2_Arg1 indicates the connective and both arguments lie in the same sentence with the connective first, followed by Arg2 and then Arg1. The feature *Prev_CONN*_Arg2 indicates Arg1 is in the previous sentence and the connective precedes Arg2 within the sentence containing the connective. Other slight variations capture configurations where the Arg1 candidate lies further back in the discourse. Finally, we found some utility in comparing the syntactic arguments (e.g., subject, direct object) of the candidate argument pairs. For example, the arguments of the discourse adverbial *also* not only frequently involve the same predicate but also involve the same entities that appear as arguments to the predicate. Currently, we simply introduce features testing whether the argument strings are identitical as a proxy for full co-reference.

Table 4 shows the results incorporating the re-ranking model for the different feature sets described earlier. The re-ranking models in each case are constructed from the features that would naturally be available to the re-ranker. For example, the re-ranking model for feature set A-Y uses a feature testing whether both candidate arguments are reporting verbs, whereas the re-ranking model for A-L doesn't.

	Accuracy			
Features	Arg1	Arg2	Conn.	Err.
A-G	44.1	59.6	30.6	11.5%
A-L	64.7	85.6	58.1	9.6%
A-G;M-Q	74.2	94.4	71.8	5.4%
A-Y	76.4	95.4	74.2	8.8%
A-Y(auto)	69.8	90.8	64.6	5.4%

Table 4: Re-ranking results for argument identification on the testing data using gold-standard and Charniak-Johnson parses for the full feature set, A-Y (auto). The error reduction (Err.) is relative to the results in Table 2.

5.2 Discussion and Error Analysis

Not surprisingly, performance at identifying Arg2s is much higher than for Arg1s as the former are syntactically bound to the connective. Indeed, performance for identifying Arg2s may be at or very close to human levels of performance using gold-standard parses. Miltsakaki et al. (2004a) indicate 94.1% inter-annotator agreement for Arg2, 86.3% on Arg1 and 82.8% agreement per discourse connective with respect to the full argument extents for a set of 10 connectives. The disagreement rates, however, would likely be reduced considerably using our head-based representation since almost half of the disagreements reported were due to argument extent disagreements.

Many of the Arg2 errors we found had to do with attribution, such as:

(5) ..“We pretty much *have* a policy of not commenting on rumors, $\boxed{\text{and}}$ I **think(?)** that **falls** in that category.

where the system proposed “think” as the Arg2 and the annotated argument was “falls”.

The Arg1 errors were much more diverse with many involving arguments in previous sentences, such as the following case in which the system proposed *owned* as the argument yet the correct argument was *completed* found three sentences prior in the discourse.

(6) ..Quantum *completed* in August an acquisition of Petrolane... Petrolane is the second-largest... The largest, Suburban Propane, was already *owned(?)* by Quantum. $\boxed{\text{Still}}$, Quantum **has** a crisis to get past right now.

Conn. Type	Freq.	Indep. Acc.	Rerank Acc.	Err.
Coord.	662	75.5	78.3	11.4%
Subord.	547	87.2	86.8	-3.0%
Adv.	386	42.2	49.0	11.8%
Total	1595	71.7	74.2	8.8%

Table 5: Frequency of each connective type and connective accuracy for the independent and reranking approaches using gold-standard parses and features (A-Y).

An examination of the errors by connective type is shown in Table 5. The re-ranking model provides considerable improvement for coordinating and adverbial connectives, but slighly *lowers* performance for subordinating connectives. Overall performance on discourse adverbials remains below 50% however.

6 Related Work

Given the formulation of discourse relations as predicate-argument structures anchored on discourse connectives, our work here bears some resemblance to work in semantic role labeling that has focused on identifying semantic frames for verbs (Toutanova et al., 2005). The task of identifying discourse relations is simpler in that there are only and exactly two arguments for each predicate; yet it is more difficult due to many more candidate arguments not contained within a single sentence.

Within discourse parsing, our work is similar to that of Soricut and Marcu (2003) but they focus only on identifying (and labeling the type of) all intra-sentential discourse relations whereas we attempt to identify discourse relations spanning multiple sentences, provided they are lexicalized by a connective. While not directly comparable to our results, they report 73.0 F-measure at identifying intra-sentential discourse relations and segments using gold-standard parses. With gold-standard discourse segments provided, their system achieves human-levels of performance (96.2

F-measure), broadly comparable to our near-human levels of performance on identifying Arg2s with gold-standard parses. Sporleder and Lapata (2005) address intra-sentential discourse modeling with a chunking approach. They achieve 88.7 F-measure on identifying discourse segment boundaries and 76.3 F-measure when also labeling each segment as a nucleus or satellite. Webber et al. (2003) provide a discourse parsing model, DLTAG, which is an extension of Lexicalized Tree Adjoining Grammars. Baldridge and Lascarides (2005) present a discourse parser for dialogue in the framework of SDRT (Asher, 1993) and achieve 67.9 F-measure on identifying and segmenting discourse relations.

7 Conclusions and Future Work

We have presented a fully automated system capable of identifying the arguments of discourse connectives. Rather than identifying the full argument extents in the PDTB, we have proposed here an alternative problem formulation: that of identifying the heads of discourse arguments. [3] With such a representation our system achieves 74.2% accuracy using gold-standard parses and 64.6% accuracy using automatic parses on the task of correctly identifying both arguments of discourse connectives. We found that syntactic features based on a dependency parse representation provide more discriminative features over those based on a constituent tree representation. Additionally, we found a notable improvement by exploiting joint features over argument pairs in a re-ranking model in comparison to modeling the arguments independently.

We have provided here, to our knowledge, the first rigorous empirical results on identifying the arguments of discourse connectives in the PDTB. Accordingly, many avenues remain for future work. Further feature engineering, particularly work capturing the lexico-semantic, attributive and predicate-argument properites of arguments appears necessary to better identify the Arg1s of anaphoric discourse adverbials, in particular. Introducing separate models and feature sets for each of the three connective types may also prove beneficial since phenomena involved vary according to connective type.

While we have demonstrated some encouraging results by modeling both arguments jointly, we hypothesize more gains are possible by modeling *inter-connective* dependencies. The discourse arguments of one connective are not independent of other (nearby) connectives and their arguments. For example, it is very rare to see crossing argument links. Capturing these inter-connective dependencies and constraints is likely to be even more important when considering the task of identifying the rhetorical *types* associated with the connectives or when considering non-lexicalized relations between adjacent sentences.

Finally, jointly modeling PropBank and the PDTB is another interesting area we plan to investigate, something to which the head-based approach and dependency parse representation we advocate here would be well-suited.

Acknolwedgements

The first author was supported in part by MITRE Sponsored Research. The authors thank Bonnie Webber, Aravind Joshi, Alan Lee and Inderjeet Mani for useful discussions and commments on an earlier draft of this paper.

References

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

Jason Baldridge and Alex Lascarides. 2005. Probabilistic head-driven parsing for discourse structure. In *Proceedings of the Ninth Conference on Computational Natural Language Learning CoNLL-2005.*

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2003. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Current Directions in Discourse and Dialogue*, pages 85–112. Kluwer Academic Publishers.

[3] Software for producing the head-based representation of the PDTB, an augmented version of the Charniak-Johnson parser that a produces dependency representation, and the log-linear ranking code are available at: http://www.cs.brandeis.edu/w̃ellner/pdtb-emnlp/

E. Charniak and M. Johnson. 2005. Coarse-to-fine n-best parsing and maxent reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*.

M. Collins. 1999. Head-driven statistical models for natural language parsing.

Michael Collins. 2000. Discriminative reranking for natural language parsing. In *Proceedings of ICML-2000*.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *In Proceedings of the 5th International Conference on Language Resources and Evaluation*, Genoa, Italy.

Pascal Denis and Jason Baldridge. 2007. A ranking approach to pronoun resolution. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*.

Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2005. Attribution and the (non)-alignment of syntactic and discourse arguments of connectives. In *Proceedings of the ACL Workshop on Frontiers in Corpus Annotation II: Pie in the Sky*, Ann Arbor, Michigan, USA.

Jerry Hobbs. 1985. On the coherence and structure of discourse.

Zheng Ping Jiang and Hwee Tou Ng. 2006. Semantic role labeling of nombank: A maximum entropy approach. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, Sydney, Australia.

Alistair Knott. 1996. *A Data-Driven Methodology for Motivating a Set of Coherence Relations.* Ph.D. thesis, University of Edinburgh.

Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2004a. Annotating discourse connectives and their arguments. In *Proceedings of the HLT/NAACL Workshop on Frontiers in Corpus Annotation*.

Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2004b. The penn discourse treebank. In *LREC 2004*.

Deepak Ravichandran, Eduard Hovy, and Franz Josef Och. 2003. Statistical qa - classifier vs. re-ranker: What's the difference? In *Proceedings of the ACL Workshop on Multilingual Summarization and Question Answering-Machine Learning and Beyond*.

Radu Soricut and Daniel Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, Edmonton, Canada.

Caroline Sporleder and Mirella Lapata. 2005. Discourse chunking and its application to sentence compression. In *Proceedings of the HLT/EMNLP*, pages 257–264, Vancouver.

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. 2005. Joint learning improves semantic role labeling. In *Proceedings of the Association for Computational Linguistics(ACL)*, Ann Arbor, Michigan, USA.

Bonnie Webber, Aravind Joshi, Alistair Knott, and Matthew Stone. 2003. Anaphora and discourse structure. *Computational Linguistics*.

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