Discourse connective argument identification with connective specific rankers

Robert Elwell and Jason Baldridge
The University of Texas at Austin
Department of Linguistics
{relwell,jbaldrid}@mail.utexas.edu

Abstract

Automatically identifying the arguments of discourse connectives (e.g., and, because, however*) is an important part of modeling discourse structure. Previous work used a single, general classifier for different connectives; however, connectives differ in their distribution and behavior, so conflating them this way loses discriminative power. Here, we show that using models for specific connectives and types of connectives and interpolating them with a general model improves performance. We also describe additional features that provide greater sensitivity to morphological, syntactic, and discourse patterns, and less sensitivity to parse quality. Our best model achieves a 3.6% absolute improvement over the state-of-the-art on identifying both arguments of discourse connectives when using features from gold-standard parses, and a 9.0% improvement when using automatically produced parses.*

1 Introduction

Automated analysis of discourse structure for raw text is a challenging but important problem. For example, determining the full discourse structure of a text has uses in practical applications such as generation [11], text summarization [17], and automated essay evaluation [10]. In this paper, we consider discourse connective argument identification, a subtask of determining full discourse structure that involves identifying the arguments of explicit discourse connectives. Connectives are words with rhetorical functions such as *and*, *because*, and *however*; their arguments are the text spans that they relate. For example, in the sentence *John left* ***because*** *he was mad*, the first argument of *because* is *John left* and the second is *he was mad*.

Accurate identification of the arguments of connectives could be useful for many semantic computing tasks in addition to those mentioned above, including sentiment analysis, textual entailment and temporal reasoning. In essence, they can act as a lightweight approximation of a full rhetorical analysis that can inform models for other tasks. For example, knowing that a particular connective relates two text spans could be used in the structured model of sentiment analysis of [19] to define a small set of rhetorically important dependencies beyond simple adjacency. Also, connectives include conditionals like *if* and negation (e.g., *not*) that affect monotonicity of inferences in textual entailment [14]. Finally, many discourse connectives–such as *then* and *after*–relate their arguments temporally: connective argument identification can thus aid in the prediction of when the events evoked by those arguments occur with respect to one another, e.g., in an approach such as that of [16].

Following Wellner and Pustejovksy (2007) (henceforth, W&P), we use the Penn Discourse Treebank (PDTB, [20, 22]), a layer of annotations for discourse connectives and their arguments over the Wall Street Journal portion of the Penn Treebank [18]. W&P use maximum entropy rankers combined with a reranking step to jointly select the two arguments of each connective. Their best results show that this strategy works well but leaves much room for improvement on identifying first arguments.

In this paper, we improve on W&P's results by using models tuned to specific connectives and connective types (*subordinating conjunctions*, *coordinating conjunctions*, and *discourse adverbials*). Like W&P, we use maximum entropy rankers. They use a single model that handles all connectives. This allows it to capture the general behavior of all connectives, but it ignores the fact that different connectives, and in particular, different types of connectives behave differently. For instance, syntactic constituency relationships generally hold between coordinating and subordinating connectives and their first argument spans, but adverbial connectives find their first arguments anywhere in the prior discourse. A general model will have difficulty resolving the tension on certain features, such as distance, since adverbials will express a preference for further attachments while the others will prefer closer ones.

Specialized models that capture the distribution for a given connective or type of connective do not have to deal with such possibly conflicting evidence from the events for

978-0-7695-3279-0/08 $25.00 © 2008 IEEE
DOI 10.1109/ICSC.2008.50

other connectives. We show here that these specialized models do indeed improve performance for this task, similarly to using specialized models for different types of mentions in automated coreference resolution [7]. However, the general model does have access to more training instances, so we interpolate the specialized models with this one to create a combined model that takes advantage of the strengths of both strategies.

We also describe new features that improve performance over those used by W&P. Our additional features encode (1) morphological properties of connectives and their arguments, (2) additional syntactic configurations, and (3) the wider context of preceding and following connectives. The latter help create more coherent assignments by indirectly adding greater sensitivity to other decisions.

2 The Penn Discourse Treebank

Several annotated corpora for discourse structure have been created and used for creating discourse parsers [5, 28, 2]. These resources assume an underlying theory of abstract discourse relations, such as *Elaboration*, *Explanation*, and *Narration*. They also assume that discourse structure is encompassed by some specific sort of directed graph (e.g. trees [5] versus more general acyclic graphs [2] versus fully general graphs [28]). The goal of parsers for these resources is to identify how different utterances, or elementary discourse units, are hierarchically connected to one another via these relations. Despite these efforts, there is still no general agreement as to what constitutes an adequate representation of discourse structure, especially for annotating texts. The issues include determining a precise set of discourse relations, questions about whether trees are adequate, and whether it is necessary to use underspecified representations of discourse structure.

The Penn Discourse Treebank (PDTB, [20]) is another annotated resource that circumvents such issues. The design of the PDTB seeks to retain a close link between the representation being constructed and the text being annotated. The focus is on explicit connectives— which signal discourse relations—and the spans of text which are the arguments of those connectives. The argument structure of each connective is strictly binary and non-recursive. This produces a shallow representation that can be annotated reliably, in large part because it remains tightly connected to the text itself and does not posit higher level structure. It also makes it straightforward to evaluate performance on discourse connective argument identification.

Knott (1996) provides an extensive study of discourse connectives and their properties[13]. An important dimension on which they vary is their syntactic type: *subordinating conjunctions*, *coordinating conjunctions*, *discourse adverbials*, *prepositional phrases*, and *phrases taking sentence complements* [23]. Examples of connectives grouped by syntactic types are given in Table 1.

Coordinating	Subordinating	Other
and	*because*	*afterwards*
or	*when*	*previously*
but	*since*	*nonetheless*
yet	*even though*	*actually*
then	*except when*	*again*

Table 1. Examples of connectives, grouped by syntactic function (from [13]).

To see how different connectives behave differently, it is useful to consider some actual examples and their annotations from the PDTB. We follow the labeling convention used in W&P, which adds head identification to the standard PDTB conventions: the connective is in a box, Arg1 in *italics*, Arg2 in **bold**, and head words of each argument are underlined. The following examples show coordinating (1), subordinating (2), and discourse adverbial (3) connectives and their arguments as annotated in the PDTB:

(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.

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

(3) *France's second-largest government-owned insurance company, Assurances Generales de France, is building its own Navigation 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.

Subordinating and coordinating connectives are typically connected to both of their arguments syntactically in the same sentence or their argument is in the immediately preceding sentence: they are structural [26]. Adverbials, on the other hand, can be anaphorically linked to their Arg1, as is clear in (3). It is exactly this kind of difference that suggests different models for different types of connectives should help.

Annotators dealt with each connective independently. As a result, two connectives can have interleaved, overlapping arguments. For example, the connectives *because* and *then* occur next to one another in the following passage; the Arg2 of *because* subsumes that of *then* while the Arg1 of *then* is before that of *because*:

(4) John loves Barolo. He ordered three cases of the '97. But *he had to cancel the order* because **then he discovered he was broke**.

(5) John loves Barolo. *He ordered three cases of the '97*. But he had to cancel the order because then **he discovered he was broke**.

Some of our new features help capture such overlaps. For example, we include features that state the previous and following connectives and whether or not there is overlap in their candidates and those for the current connective.

It is important to select a word with some syntactic motivation to represent an argument span, but due to the lack of consistent alignment between syntax and discourse, we must enforce a syntactic relationship between the various nodes in the span forest. Following W&P, we find the least common ancestor (LCA) node which governs all terminal nodes within the argument span and evaluate on correctly selecting the terminal head of the LCA.

3 Models

There are two stages in identifying the arguments of a discourse connective: the heads of candidate arguments must be identified and then the best candidate must be chosen. For the first stage, W&P select candidates only within a distance of ten *steps* of the connective. A step is defined as the traversal of a sentence boundary or dependency link. The heuristic may traverse sentence boundaries for ARG1S, but stays within the same sentence as the connective for ARG2S, which are generally syntactically dependent upon the connective. This prunes the number of candidates which must be considered during classification.

To identify the best candidates, W&P use a maximum entropy ranker with a large feature set which considers syntax, dependency, and lexical semantics. Maximum entropy models are widely used for classification tasks in natural language processing—they are accurate, can incorporate non-independent, overlapping features, and are reasonably fast to train. The advantage to using rankers (as opposed to classifiers) for this particular task is that in the set of candidates generated, we may assume there is only one correct answer. The model considers all candidates for a given connective simultaneously and selects only one candidate. A classifier for this task would have to consider each candidate independently, predicting "Yes this is an argument" or "No, it isn't" for each—similarly to the standard classification approach for coreference resolution [25]. Rankers have been shown to improve accuracy for tasks with a similar structure, such as question answering [24], pronoun resolution [8], and coreference resolution [7].[1]

In the composition of the PDTB, there cannot be more than one first argument and second argument for a single connective, so ranking is thus a good fit for this task as well. Like W&P, we also use a maximum entropy ranker. Models are trained for ARG1 and ARG2 selection separately. The model for ranking with respect to the identity of a candidate head α_i as an argument head $\hat{\alpha}$ given a connective π and a document *x* is:

$$P_{\hat{\alpha}}(\alpha_i|\pi, x) = \frac{exp(\sum_k \lambda_k f_k(\alpha_i, \pi, x))}{\sum_{\alpha_j \in C_{\hat{\alpha}}(\pi,x)} exp(\sum_k \lambda_k(\alpha_j, \pi, x))} \qquad (1)$$

where the f_k are feature functions, the λ_k are their respective weights, and $C_{\hat{\alpha}}(\pi, x)$ is the set of candidate arguments for connective π within a document *x*. During the training phase, the incorrect alternatives for this discriminative model are sampled as in W&P: all candidates within ten steps of the connective, as mentioned above. During testing, the same criterion is used for identifying the candidate set; the chosen candidate is then that which has the highest probability according to (1). For the rest of this paper, we refer to this approach as the general connective model, or GC. The number of training instances coincides with the number of connectives in the training set.

We use the Toolkit for Advanced Discriminative Modeling[2] [15] to determine weights for this model and the specialized models described in the next two sections. For all models, we use a Gaussian prior with a variance of 100, determined on the development set.

Connective Specific Models. Discourse encompasses many levels of analysis (e.g., lexical semantics, pragmatics, temporal semantics, and world knowledge), so we consider the value of training models specific to the connective in question so as to better model their individual properties. For example, *then* typically relates two temporally related arguments whereas *but* usually contrasts two propositions regardless of temporal considerations.

For each individual connective, we train models using all its instances. During testing, the prediction for a connective is determined by the specific model trained for it and no others. In a few cases, a connective unseen in the training

[1] A common point of confusion is how rankers differ from classifiers. The main difference is that all candidates are considered together in a single decision. The features in a classifier are a combination of a *contextual predicate*, like **candidate_head=Choose** and a class *label*, like **Yes**: so the resulting *feature* is **(Yes,candidate_head=Choose)**. A decision is then about a *single* candidate with respect to how the contextual predicates identified in it pair with the different labels in consideration. In a ranker, the contextual predicate *is* the feature, and multiple candidates are compared together. A consequence of this is that items in a ranking decision can have features in common, whereas those in classifiers are disjoint. For example, in a parse selection task [3], several parses of a sentence might share a feature that encodes the fact that the **S** → **NP VP** rule was used.

[2] http://tadm.sf.net

material occurs in the test material; for such cases, the GC model is uses as a backoff model. We refer to the collection of connective specific models as SC.

Type Specific Models. Connectives link arguments in primarily three different ways [13]. Coordinating connectives (i.e. *and*) assume argument spans are generally syntactically similar. Subordinating connectives (i.e. *because*) are dominated or adverbially linked to the ARG1 and dominate the ARG2. Adverbial connectives (i.e. *nonetheless*) can appear in several places in the sentence and have no necessary syntactic relationship to their ARG1. Because of this tendency, this subset of connectives more closely mirrors the task of coreference resolution. As a result, we explore modeling each set independently to more closely reflect the differing behaviors of each connective group. Connective types are determined from the list of connectives and their properties given in Appendix A of [13]. As a default, any connective not mentioned there is assumed to be adverbial. This arrives at a slightly different frequency of the three types than outlined in W&P, but stays fairly close to the distribution amongst all types. We refer to this set of (three) models as the type connective model, or TC.

Interpolated Models. The more specific models described in the previous sections trade off precise modeling of the distribution for specific connectives or types of connectives with the extra evidence (and larger training set) available with the GC model. We thus combine models of different specificity using standard linear interpolation as a simple way of getting the benefits of both. We use two different interpolated models: (1) TC with GC and (2) SC combined with the interpolation of TC and GC:

$$P_{TG}(a|c_i) = \lambda_{t_i} P_{t_i}(a) + (1 - \lambda_{t_i}) P_g(a) \tag{2}$$

$$P_{SGT}(a|c_i) = \lambda_{c_i} P_{c_i}(a) + (1 - \lambda_{c_i}) P_{TG}(a|c_i) \tag{3}$$

where a is a candidate argument, c_i is the specific connective under consideration, P_{c_i} is the specialized model for connective c_i, t_i is the type of that connective, P_{t_i} is the model for that type, P_g is the general connective model, and λ_{c_i} and λ_{t_i} are interpolation weights for connective c_i and connective type t_i, respectively, controlling the influence of the more general models.

The interpolation weights λ_{c_i} and λ_{t_i} are determined in a very simple manner that ensures that connectives or types which have been seen fewer times leave more mass for P_{TG} or P_{SGT}, respectively. For example:

$$\lambda_{c_i} = \frac{freq(c_i)}{freq(c_i) + C} \tag{4}$$

C is a constant chosen based on performance on held-out development data. λ_{t_i} is set similarly. For λ_{c_i}, we set C at 99, the number of different connectives in the corpus. For λ_{t_i}, C is set as 3, the number of different connective types.

More complex model combinations could of course be used. Nonetheless, we find that this simple approach works well with these base models and is robust to many different values of C for both connectives and connective types.

Why use specific models? One reviewer wondered about the connection between specialized models and other common natural language processing tasks: e.g., why are word-specific models not used for part-of-speech tagging, where a general model is used and features always include the word itself? Actually, basically all POS taggers *do* use a word-specific model, albeit in the form of a tag dictionary. Tag dictionaries restrict the possible tags that can be assigned to a word to be only those tags that have been seen with that word in training. In effect, the tag dictionary acts like a word-specific model in that it is a hard filter on what the tagger can assign to each seen word. However, it should also be noted that tasks like the one considered here are quite different from POS tagging in that they do not involve selecting a label from a small predefined set, but instead involve relating arbitrary text spans. They are thus far more unconstrained, and as we have argued based on the data, they behave differently in terms of the spans they associate with and the distance with which they find their arguments. This is similar to coreference, where pronouns are usually coreferent with a noun phrase in the current or preceding sentence, but proper nouns are often coreferent with other mentions that are many sentences away [7].

The most extreme case of the use of specialized models is word-specific models for word-sense disambiguation [12] where the possible set of senses (the labels to be predicted) is different for each word. Creating general models is actually quite challenging and there are very few approaches that go beyond the word level; nonetheless, improvements have been achieved by learning correlations across the predictions for different words [1].

4 Features

Discourse tasks such as connective argument identification are influenced by many aspects of the text and its underlying content. W&P use a wide-range of features that encode aspects of the surface text, such as the connective string itself, phrase structure and dependency path information, and some lexical semantics with respect to the behavior of the candidate and connective.

Many other features can be extracted for the task. We consider an additional set that attempts to account better for the prediction of one connective based on other neighboring connectives and that would be especially useful for models for specific connectives or connective types. We also use

α	Previous connective string
β	Following connective string
γ	Connective in quotes
δ	Candidate in quotes
ϵ	γ & δ
ζ	γ & δ AND same quote
η	Word to the left of the candidate
θ	Word to the right of the candidate
ι	Word to the left of the connective
κ	Word to the right of the connective
λ	The unique set of node labels in the constituent path
μ	Lemmatized candidate
ν	Inflection features of candidate (-ing/-ed, etc.)
ξ	Candidate a constituent of another connective
o	Candidate argument position in dependency graph
π	Connective argument position in dependency graph
ρ	Candidate in a copular sentence (boolean)

Table 2. Additional features.

morphological stemming and additional syntactic features not considered by W&P. In part, we seek to provide greater robustness to the disconnect between syntactic constituents and connective arguments in the PDTB [9].

We modify or do without some of W&P's features, based on performance observed on the development material, as follows. Directional information for constituent paths is omitted, making the features simply serial part-of-speech strings. Collapsed constituent path without part-of-speech (their feature J) is removed. This is replaced by the path of constituent lexical heads from the candidate to the connective. This helped particularly in improving ARG1 scores on the development data.

Our additional features are summarized in Table 2. Because a ranking instance occurs for each connective with all candidate heads considered at once, there is more room to consider dependencies between features. This is particularly important for cases of time-sensitive connectives, where a connective such as *seven months after* would most likely not have an ARG1 in the present tense and an ARG2 in the past tense.

Features α through κ are intended to introduce a higher level of context-sensitivity using simple, surface-level cues. Depending on the surroundings of the connectives, there may be preferences for different types of arguments which a model dealing with a specific connective or type of connective might more easily capture. This is also the reasoning behind feature σ: certain connectives are unlikely to take copular arguments. Similarly for feature λ; certain phrasal nodes may block well-formed candidate heads within the discourse–for example, a subordinate or adverbial clause.

Features γ through ζ deal with within-quotes discourse as opposed to document-level discourse. In some ways, these features are elaborations on W&P's feature U, regarding whether the candidate is an attributing verb: they may help to block narrative document-level discourse from being related to discourse stemming from a specific speaker whose dialogue is being directly attributed.

The output of the `morpha` lemmatizer [21] is used for several features to allow generalization over lemmata as well as introduce weak tense-based features as relations between the first and second argument, as in feature ν. This is an important aspect to include as a weak indicator of temporal factors, which should help for connectives such as *fully eight months before*, *four days after*, and *ever since*.

Feature ξ should discourage selection of head words which are immediate constituents of other connectives. These are much more likely to be ARG2s of other connectives than first arguments of the connective in question.

In all, the additional features seek to capture a greater degree of discourse-level context and should also be helpful for connective specific models. Features which consider tense and adjacent connectives should improve sensitivity to the greater discourse; intuitively, as a connective's argument selection behavior is a consequence of the full discourse, this should improve performance.

5 Experiments

We use version 1.0 of the Penn Discourse Treebank. As is standard when using the Wall Street Journal portion of the Penn Treebank, we train on sections 02-22, develop on sections 00 and 01, and test on sections 23 and 24. This allows us to directly compare the performance of our approach to W&P. All models were developed on 00-01 and run once on 23-24. Like W&P, we evaluate our approach for both ARG1 and ARG2 identification and for identifying both correctly (referred to as CONN accuracy). We also break down performance in terms of connective types (see section 3 for how we determined these types from [13]).

We also evaluate the performance of models on automatically produced parses using the Bikel parser [4].[3] The parsed data for the training portion is created using 5-fold training and labeling with the parser.

Results for ARG1, ARG2, and CONN accuracy for the various models are given in Table 3 for gold parses and Table 4 for auto-parses. Scores are given for W&P's best simple ranking model (W&P-BASE) and their reranker (W&P-RERANK). Our models are simple ranking models, like W&P-BASE,[4] and would all likely improve with reranking.[5] A factor that is clear in all the models is that ARG2

[3] W&P obtained auto-parses from the Charniak parser [6].

[4] We replicated this model and obtained the scores reported by W&P.

[5] As with the classifier versus ranker question, there is at times confusion between a ranker and a *re*ranker. Rerankers take the n-best output of a previous model (thus not considering all possibilities the first model did), and typically take advantage of a wider range of features than the first did.

selection is considerably easier than Arg1. This is unsurprising since Arg1 identification requires considering multiple sentences, and thus more candidates. In addition, as the average distance between the connective and the candidate argument increases, features such as syntactic path and dependency path between nodes becomes sparser and less useful. Conversely, connectives generally have a direct syntactic relationship with Arg2s.

Base model results: new features. Both of the W&P scores use all of their features. By changing some of the features and removing others, as described in Section 4, we obtain a model, GC-W&P-Revised, that has better Arg1 accuracy (78.1 vs. 75.0), worse Arg2 accuracy (91.8 vs 94.2), but is overall more accurate at getting both connectives right (73.0 vs 71.7). When we add our additional features (Table 2), performance improves on all measures (indicated by the row for GC-All), and in particular the Conn accuracy nearly rivals that of W&P-Rerank (73.9 vs 74.2). These results show the utility of our additional features for Arg1 accuracy, but also demonstrate that W&P's original set is better for Arg2 identification. This suggests that the two tasks should be tackled with feature sets tuned to each, rather than a single feature set as W&P did and we have done here.

As seen on Table 4, the models implemented here do not suffer as severely using auto-parse data as W&P's. One likely explanation may be that the revised feature set removes some of the constituency based features and adds features that do not reference syntactic structure, thus reducing reliance on parse quality. It is also possible that the output of the different parsers, Bikel's in our case and Charniak's in W&P's case, contributes to this difference.

Base model results: specialized models. The results for the connective-type model, TC-All, shows that further gains are made by using a model which treats each type of connective separately. In particular, this improves Arg1 accuracy (by 3%, from 78.7 to 81.7). This is as expected, since it is with respect to Arg1 that the different connective types behave most differently. As discussed in Section 2, subordinating and coordinating connectives usually find their Arg1s structurally, whereas adverbial connectives find them anaphorically (and usually at a greater distance).

Model	Arg1	Arg2	Conn
W&P-Base	75.0	94.2	71.7
W&P-Rerank	76.4	95.4	74.2
GC-W&P-Revised	78.1	91.8	73.0
GC-All	78.7	92.1	73.9
TC-All	81.7	92.6	76.1
SC-All	80.3	93.1	75.8
TC-GC-Interp	81.7	93.2	77.2
SC-TC-GC-Interp	82.0	93.7	77.8

Table 3. Accuracy scores on gold-standard parses.

Model	Arg1	Arg2	Conn
W&P-Base	67.9	90.6	62.7
W&P-Rerank	69.8	90.8	64.6
GC-W&P-Revised	76.9	89.0	70.2
GC-All	77.1	89.1	70.2
TC-All	78.9	89.7	72.4
SC-All	78.7	85.4	68.4
TC-GC-Interp	79.8	89.9	73.1
SC-TC-GC-Interp	80.0	90.2	73.6

Table 4. Accuracy scores on Bikel parses.

The connective specific model, SC-All, does not do as well as TC-All on Arg1, but is still better than the general GC-All model. It is our best simple model on Arg2. This suggests that TC-All is still too course-grained: Arg2s are found in the same sentence, and both GC-All and TC-All must have some connectives which overpower the preferences of others. On the other hand, because Arg1s are further away for adverbial connectives, TC-All may be better able to use the evidence from all adverbial connectives while not suffering from the great reduction in training instances incurred by SC-All (consider, for example, that many connectives—especially the adverbial ones—are found only once in the training material). In particular, this means that the general problem of sparser features for Arg1s is greatly aggravated in the SC-All model.

The most notable detail of the performance of specialized models on the auto-parse data is the decrease in performance for SC-All. The weakest performance for this model is among subordinating connectives (see Table 6).

Interpolated model results. Clearly, there is value in using more specific models, but they must be balanced by more general models to protect against sparsity. The simple linear interpolation described in section 3 is a straightforward way to combine these models. As indicated by

For example, in parsing, a generative model that uses very limited, local features produces an n-best list of parses, and a reranker uses features that span large portions of entire trees [6]. W&P use a ranker in the following manner: their initial ranker predicts Arg1s and Arg2s separately, then an n-best list of *pairs* of arguments is created by multiplying the probability of each argument, and finally, the resulting n-best list creates a training instance (with the gold-standard providing the truth) or a test instance to be evaluated with the trained reranking model. This allows an approximation of a joint model over both arguments. See W&P for more details.

Model	Subord.	Coord.	Adverb.
GC-W&P-REVISED	80.8	74.2	58.7
GC-ALL	80.9	75.5	59.8
TC-ALL	82.8	77.5	67.5
SC-ALL	83.9	78.0	59.0
TC-GC-INTERP	82.8	77.5	67.8
SC-TC-GC-INTERP	83.9	78.1	67.5

Table 5. Breakdown of CONN scores by connective type on gold parses.

Model	Subord.	Coord.	Adverb.
GC-W&P-REVISED	76.2	73.3	55.4
GC-ALL	76.9	73.5	54.0
TC-ALL	78.2	75.4	58.2
SC-ALL	67.9	77.4	53.2
TC-GC-INTERP	77.3	76.5	60.7
SC-TC-GC-INTERP	78.0	77.1	60.7

Table 6. Breakdown of CONN scores by connective type on Bikel parses.

the row for TC-GC-INTERP, interpolating TC-ALL with GC-ALL does help. In particular, ARG2 accuracy improves from 92.6 for TC-ALL on its own to 93.2. Because ARG2s are more similar across the different types of connectives, the interpolated model is able to incorporate additional—and more importantly, relevant, appropriate, and more numerous—evidence from GC-ALL.

Interpolating the three levels together, SC-TC-GC-INTERP, provides our best results. This combined model can use SC-ALL when it has a specific connective that had many training instances and thus can be modeled well by the single specialized model while relying more on TC-GC-INTERP for connectives which were observed just a few times in the training material. (These contributions are governed by the connective specific λ_{c_i} weights, as defined in section 3). This model achieves the best performance in all three metrics and provides a 3.6% relative improvement over W&P's reranking model. On auto-parsed data, this model achieves a 9% improvement over W&P's reranking model. We would expect to get even better results by reranking the output of SC-TC-GC-INTERP.

Results by connective type. It is instructive to consider the results for each of the models on CONN accuracy for each different type of connective. This is given in Table 5 and Table 6 for gold parses and auto-parses, respectively.[6] The most salient number is 67.5 for adverbial accuracy for TC-ALL. Clearly, this single model represents the most important split in specialization since it allows the structural dependencies of subordinating and coordinating connectives to be modeled differently from the anaphoric dependencies of adverbials. It thus captures the longer distance adverbial ARG1s more accurately than GC-ALL. Although SC-ALL also specializes (even further than TC-ALL), it is hurt by subsequent sparsity since many of the models that constitute it are trained on just a few examples. TC-ALL thus provides a good balance between both extremes of GC-ALL and SC-ALL.

SC-ALL, on the other hand, does best of all single models for subordinating and coordinating connectives. As mentioned above, sparsity is less of an issue for these connectives since their arguments are usually found much closer and in stricter syntactic relationships to them. Of course, the interpolated models straightforwardly incorporate the benefits of all these strategies and perform better across the three types than any of the single models.

6 Conclusion

We have shown that accuracy in identifying the arguments of discourse connectives can be improved by building models for specific connectives and/or different connective types. These models allow the specific distributions for a connective or connective type to be modeled more closely, but suffer from not having as much training material as a general model that uses all connectives. This is similar to what has been found for specialized models for coreference resolution [7], a task with a very similar structure. We additionally show that the strengths of both the specialized models and the general model can be realized by combining them with simple interpolation.

We also demonstrate the utility of additional features for the task. These features use morphological analysis, further syntactic patterns, and information about the distribution of other connectives in relation to the connective under consideration. By using these new features and interpolating a connective specific model, connective type model, and general model, we achieve 77.8% accuracy for identifying *both* arguments of connective, a 3.6% absolute accuracy improvement over the state-of-the-art result of W&P [27].

An immediate way to improve our results is to use separately tuned feature sets for ARG1 and ARG2 identification, as can be seen in the difference between W&P's basic model and our revised version of their feature set: the former does better on ARG2, whereas the latter is better on ARG1. Our subsequent models—using specialization—improve ARG1 accuracy using the same features as the lat-

[6] Note that our connective type classification is slightly different from W&P's. In the test material, they counted 662 coordinating, 547 subordinating, and 386 adverbial connectives; according to our connective types, based on [13], there are 654, 577, and 366, respectively.

ter, and they nearly rival that of W&P's basic model for ARG2. Thus, building on W&P's features for ARG2 identification would likely improve the results of our specialized models. And of course, we expect our best models would additionally benefit from W&P's reranking approach.

Acknowledgments The authors would like to thank Ben Wellner for providing feedback and data that helped replicate his results. We would also like to thank David Beaver, Katrin Erk and the anonymous reviewers for their feedback. This work was supported by NSF grant IIS-0535154.

References

[1] R. K. Ando. Applying alternating structure optimization to word sense disambiguation. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 77–84, New York City, June 2006. Association for Computational Linguistics.

[2] J. Baldridge, N. Asher, and J. Hunter. Annotation for and robust parsing of discourse structure on unrestricted texts. *Zeitschrift fur Sprachwissenschaft*, 26(213–239), 2007.

[3] J. Baldridge and M. Osborne. Active learning and logarithmic opinion pools for HPSG parse selection. *Natural Language Engineering*, 14(2):199–222, 2008.

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

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

[6] E. Charniak and M. Johnson. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 173–180, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[7] P. Denis. *New Learning Models for Robust Reference Resolution*. PhD thesis, The University of Texas at Austin, 2007.

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

[9] N. Dinesh, A. Lee, E. Miltsakaki, R. Prasad, A. Joshi, and B. Webber. Attribution and the (non-)alignment of syntactic and discourse arguments of connectives. In *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, pages 29–36, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

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

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

[12] N. Ide and J. Véronis. Introduction to the special issue on word sense disambiguation: the state of the art. *Computational Linguistics*, 24(1):1–40, 1998.

[13] A. Knott. *A Data-Driven Methodology for Motivating a Set of Coherence Relations*. PhD thesis, University of Edinburgh, 1996.

[14] B. MacCartney and C. D. Manning. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200, Prague, June 2007.

[15] R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002)*, pages 49–55, 2002.

[16] I. Mani, M. Verhagen, B. Wellner, C. M. Lee, and J. Pustejovsky. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 753–760, Sydney, Australia, July 2006.

[17] D. Marcu. The rhetorical parsing of unrestricted natural language texts. In *Proceedings of ACL/EACL*, pages 96–103, July 1997.

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

[19] R. McDonald, K. Hannan, T. Neylon, M. Wells, and J. Reynar. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 432–439, Prague, Czech Republic, June 2007.

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

[21] G. Minnen, J. Carroll, and D. Pearce. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223, 2001.

[22] R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber. The penn discourse treebank 2.0. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, 2008.

[23] R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik. *A Grammar of Contemporary English*. Longman, London, 1972.

[24] D. Ravichandran, E. Hovy, and F. J. Och. 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*, 2003.

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

[26] B. L. Webber, A. Knott, M. Stone, and A. Joshi. Anaphora and discourse structure. *Computational Linguistics*, 29(4):589–637, 2003.

[27] B. Wellner and J. Pustejovsky. Automatically identifying the arguments of discourse connectives. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 92–101, 2007.

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