Shallow Discourse Parsing with Conditional Random Fields

Sucheta Ghosh Richard Johansson Giuseppe Riccardi Sara Tonelli

Department of Information Engineering and Computer Science, University of Trento FBK-IRST

{ghosh,johansson,riccardi}@disi.unitn.it satonelli@fbk.eu

Abstract

Parsing discourse is a challenging natural language processing task. In this paper we take a data driven approach to identify arguments of explicit discourse connectives. In contrast to previous work we do not make any assumptions on the span of arguments and consider parsing as a token-level sequence labeling task. We design the argument segmentation task as a cascade of decisions based on conditional random fields (CRFs). We train the CRFs on lexical, syntactic and semantic features extracted from the Penn Discourse Treebank and evaluate feature combinations on the commonly used test split. We show that the best combination of features includes syntactic and semantic features. The comparative error analysis investigates the performance variability over connective types and argument positions.

1 Introduction

Automatic discourse processing is considered one of the most challenging NLP tasks due to its dependency on lexical and syntactic features and on the inter-sentential relations. While automatic discourse processing of structured documents or free text is still in its infancy, a number of applications of this technology in practical NLP systems have been proposed. For instance, Somasundaran et al. (2009) describe the use of discourse structure for opinion analysis. Other applications include conversational analysis and dialog systems (Tonelli et al., 2010).

In this work we divide the whole task of discourse parsing into two sub-tasks: connective classification and argument segmentation and classification. Several successful attempts have already been made in the direction of automatic classification of connectives, while token-level argument segmentation has not been explored. Therefore in this paper we will focus on the segmentation and labeling of discourse arguments (`Arg1` and `Arg2`) with full spans, as defined in the annotation protocol of the Penn Discourse Treebank (PDTB) (Prasad et al., 2008).

We present a methodology that, given explicit discourse connectives, automatically extracts discourse arguments by identifying `Arg1` and `Arg2` including the corresponding text spans. We call this approach *shallow* following Prasad et al. (2010) as opposed to tree-like representations of discourse, as in Rhetorical Structure Theory (Mann and Thompson, 1988). Indeed, we provide a flat chunk classification of discourse relations, building a non-hierarchical representation of the relations in a text.

The discourse parser is designed as a cascade of argument-specific CRFs trained on different sets of lexical, syntactic and semantic features. The evaluation is made in terms of exact and partial match of arguments. The partial match condition may be useful in the case of noisy input or for applications that do not require exact alignment.

The paper is structured as follows: in Section 2 we present related work to discourse parsing. In Section 3 we detail argument annotation in PDTB and we report some statistics about the PDTB corpus. In Section 4 the pipeline implemented for the argument segmentation and classification task is presented while in Section 5 two different feature sets used for classification are detailed and compared. In Section 6 the experimental setup is described, together with an extensive evaluation and error analysis. Finally, we draw some conclusions in Section 7.

2 Related Work

The task that we address in this paper – automatic extraction of discourse arguments for given ex-

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 1071–1079,
Chiang Mai, Thailand, November 8 – 13, 2011. ©2011 AFNLP

plicit discourse connectives – has been attempted a number of times. Soon after the initial release of the PDTB, it was realized that sentence-internal arguments may be located and classified using techniques similar to semantic role detection and classification methods. Wellner and Pustejovsky (2007) were the first to carry out such an experiment on the PDTB, and Elwell and Baldridge (2008) later improved over their results. However, their task was limited to retrieving the argument *heads*. In contrast, we integrate discourse segmentation in the parsing pipeline because we believe that spans are necessary when using the discourse arguments as input to applications such as opinion mining, where attributions need to be explicitly marked. Besides, no gold data are available for head-based discourse parsing evaluation and they have to be automatically derived from parse trees with a further processing step. With our approach, instead, we can directly use PDTB argument spans both for training and for testing.

Dinesh et al. (2005) extracted complete arguments with boundaries, but only for a restricted class of connectives. The recent work by Prasad et al. (2010) is also limited, since their system only extracts the *sentences* containing the arguments.

In our work, we assume that explicit discourse connectives are given beforehand, either taken directly from a gold standard or automatically identified. The second task based on PDTB was tackled among others by Pitler et al. (2008) and Pitler and Nenkova (2009).

In addition to the work on finding explicit connectives and their arguments, there has been recent work on classification of *implicit* discourse relations, see for instance Lin et al. (2009). In a similar classification experiment, Pitler et al. (2009) investigated features ranging from low-level word pairs to high-level linguistic cues, and demonstrated that it is useful to model the sequence of discourse relations using a sequence labeler. Although they both outperformed their respective baselines, this task is very difficult and performances are still very low.

3 The Penn Discourse Treebank (PDTB)

The Penn Discourse Treebank (Prasad et al., 2008) is a resource including one million words from the Wall Street Journal (Marcus et al., 1993), annotated with discourse relations.

Based on the observation that "no discourse connective has yet been identified in any language that has other than two arguments" (Webber et al. (2010), p. 15), connectives in the PTDB are treated as discourse predicates taking two text spans as *arguments*, i.e. parts of the text that describe events, propositions, facts, situations. Such two arguments in the PDTB are just called `Arg1` and `Arg2` and are chosen according to syntactic criteria: `Arg2` is the argument syntactically bound to the connective, while `Arg1` is the other one. This means that the numbering of the arguments does not necessarily correspond to their order of appearance in text.

In the PDTB, discourse relations can be overtly expressed either by *explicit* connectives, or by *alternative lexicalizations* (AltLex). The first group of connectives corresponds primarily to a few well-defined syntactic classes, while alternative lexicalizations are generally non-connective phrases used to express discourse relations, such that the insertion of an explicit connective would lead to redundancy. There is also a third type of relations - the *implicit* ones - which can be inferred between adjacent sentences, even if no discourse connective is overtly realized.

Every kind of relation (i.e. explicit, implicit and AltLex) in the PDTB is assigned a sense label based on a three-layered hierarchy: the top-level *classes* are the most generic ones and include EXPANSION, CONTINGENCY, COMPARISON and TEMPORAL labels (see below resp. examples from *a* to *d*). Then, each class is further specified at *type* and *subtype* level. Since the state of the art in automatic surface-sense classification (at *class* level) has already reached the upper bound of inter-annotator agreement (Pitler and Nenkova, 2009), we do not include this task in our pipeline. Instead, we use the *class* label as one of our features, because we can expect to achieve similar performance both with gold standard and with automatically assigned classes.

As for the relations considered, we focus here exclusively on *explicit* connectives and the identification of their arguments, including the exact spans. This kind of classification is very complex, since `Arg1` and `Arg2` can occur in many different configurations. Consider for example the following explicit relations annotated in the PDTB[1]:

[1] In all examples of this paper, `Arg1` is reported in italics, `Arg2` appears in bold and discourse connectives are underlined. At the end of the sentence we specify the *class* label

(a) *I never gamble too far.* In particular **I quit after one try, whether I win or lose.** [Expansion]

(b) Since **McDonald's menu prices rose this year**, *the actual deadline may have been more.* [Contingency]

(c) As an indicator of the tight grain supply situation in the U.S., market analysts said **that late Tuesday the Chinese government**, *which often buys U.S. grains in quantity,* **turned** instead **to Britain to buy 500,000 metric tons of wheat.** [Comparison]

(d) When **Mr. Green won a $240,000 verdict in a land condemnation case against the State in June 1983**, he says, *Judge O'Kicki unexpectedly awarded him an additional $100,000.* [Temporal]

An explicit connective can occur between two arguments (a) or before them (b). It can also appear inside the argument as shown in (c), where `Arg2` is composed of three discontinuous text spans and `Arg1` is interpolated. Furthermore, `Arg1` and `Arg2` need not to be adjacent, as shown in (d), where "he says" does not belong to any argument span. The latter case is annotated as an *Attribution* in the PDTB, because it ascribes the assertion in text to the agent making it. Attributions occur in 34% of all explicit relations in the PDTB, and represent one of the major challenges in identifying exact argument spans, especially for `Arg2`. However, given the fact that `Arg2` is syntactically bound to the connective, its identification is generally considered an easier task than the detection of `Arg1` (Prasad et al., 2010). As shown in Table 1, the position of `Arg1` w.r.t. the discourse connective is highly variable and, when it does not occur in the same sentence of the connective, it can be very distant from `Arg2`, even in a preceding paragraph.

Explicit connectives (tokens)	$18,459$
Explicit connectives (types)	100
`Arg1` in same sentence as connective	60.9%
`Arg1` in previous, adjacent sentence	30.1%
`Arg1` in previous, non adjacent sentence	9.0%

Table 1: Statistics about PDTB annotation from Prasad et al. (2008).

Another element increasing the complexity of `Arg1` and `Arg2` identification is the fact that discourse connectives can be expressed by subordinating and coordinating conjunctions as well as by discourse adverbials, and each type is subject to different discourse constraints. Furthermore, argument spans range from clauses, even single verb phrases, to multiple sentences, and they do not necessarily match single constituents in the syntax because they can be discontinuous. For all these reasons, the identification of `Arg1` has been only partially addressed in previous works (see for instance Prasad et al. (2010).

The PDTB achieved high-valued inter-annotator agreement. Overall agreement for identifying both the arguments (`Arg1` and `Arg2`) of explicit connectives reached 90.2%, with a general tendency of lower scores for `Arg1` and higher scores for `Arg2`. When considering a matching technique that gives credit also to partial overlap, the agreement reaches 94.5% for explicit connectives (Wellner, 2009).

4 Processing pipeline

We show that discourse annotation can be performed *in a pipeline* handling all types of explicit connectives and argument positions. The fundamental idea is to divide the whole complex task into several small and simpler independent subtasks, in order to feed the output of each step into the following one. An overview of the pipeline is given in Fig. 1. Note that, this representation includes data pre-processing, training and testing.

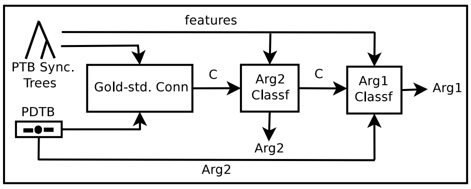

Figure 1: Argument parsing pipeline given Gold-Std Connective(C)

In contrast to previous works, our shallow parsing strategy combines the identification of non-overlapping sequences as connective arguments and the tagging of such text chunks with `Arg1` and `Arg2` labels.

Since our experiments are based on gold-standard parse trees, we take advantage of the overlap between the PDTB and the Penn Treebank documents (Marcus et al., 1993) in order to map PDTB discourse annotation onto PTB parse trees. We extract the gold-standard connectives with the corresponding top-level sense label from PDTB relations, since this sense label is also one of the features used by our system. This feature is denoted as C in Fig. 1. Besides, we also extract from the PTB trees all syntactic features needed by the system for the first parsing subtask, which is the identification of `Arg2`.

After the identification of `Arg2` given the connective sense label and feature(s) from the gold parse trees, we proceed with the classification of `Arg1`. This step-by-step methodology is different from previous approaches like the one by Wellner and Pustejovsky (2007), where the authors select *pairwise* the best heads of `Arg1` and `Arg2` in order to capture their dependencies, and also by Elwell and Baldridge (2008), who additionally develop *connective-specific* models. Our approach is motivated by two intuitions: first, the identification of `Arg2` and `Arg1` may require different features, since the two arguments have different syntactic and discourse properties, as discussed in Section 3. Second, the identification of `Arg2` is much easier than the identification of `Arg1`, because the former is syntactically bound to the connective. For this reason, a two-step decision architecture seems more appropriate, because we can start with the easier classification task and then exploit additional output information to tackle the second task.

5 Feature description

We report in Table 2 the list of all features considered in the argument labeling task and we explain them in the light of the example in Fig. 2.

Despite the complex task, the feature set is quite small for both arguments. For the identification of `Arg1`, we include one additional features which corresponds to `Arg2` gold standard labels. Note that the best performing set of features does not include all those listed in the table (see feature analysis in Tables 4 and 5).

Features used for `Arg1` and `Arg2` segmentation and labeling.	
F1.	Token (T)
F2.	Sense of Connective (CONN)
F3.	IOB chain (IOB)
F4.	PoS tag
F5.	Lemma (L)
F6.	Inflection (INFL)
F7.	Main verb of main clause (MV)
F8.	Boolean feature for MV (BMV)
F9.	Previous sentence feature (PREV)
Additional feature used only for `Arg1`	
F10.	`Arg2` Labels

Table 2: Feature sets for `Arg1` and `Arg2` segmentation and labeling.

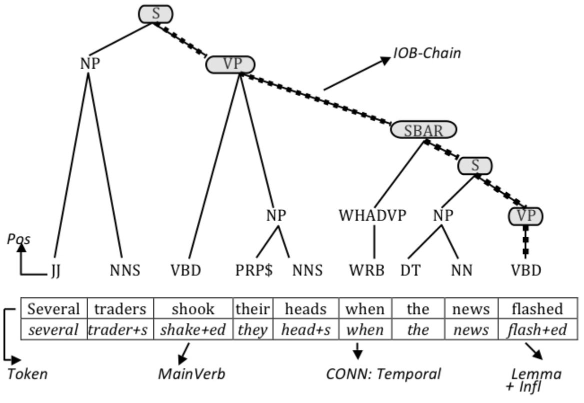
S
NP
VP
IOB-Chain
SBAR
S
NP
WHADVP
NP
VP
Pos
JJ
NNS
VBD
PRP$
NNS
WRB
DT
NN
VBD
Several traders shook their heads when the news flashed
several trader+s shake+ed they head+s when the news flash+ed
Token
MainVerb
CONN: Temporal
Lemma
+ Infl

Figure 2: Example sentence with system features

The sense of the connective (F2) refers to one of the four top-level classes in PDTB sense hierarchy, namely TEMPORAL, COMPARISON, CONTINGENCY and EXPANSION. In the sentence reported in Fig. 2, for example, only "when" bears the *temporal* label, while all other tokens are assigned as a "null".

The IOB(Inside-Outside-Begin) chain[2] (F3) is extracted from a full parse tree and corresponds to the syntactic categories of all the constituents on the path between the root note and the current leaf node of the tree. Experiments with other syntactic features proved that IOB chain conveys all deep syntactic information needed in the task, and makes all other syntactic information redundant, for example clause boundaries, token distance from the connective, constituent label, etc. In Fig. 2 the path between "flashed" and the root node is highlighted. The corresponding feature would be *I-S/E-VP/E-SBAR/E-S/C-VP*, where B-, I-, E- and C- indicate whether the given token is respectively at the beginning, inside, at the end of the constituent, or a single token chunk. In this case, "flashed" is at the end of every constituent in the chain, except for the last VP, which dominates one single leaf.

In order to extract the morphological features needed, we use the *morpha* tool (Minnen et al., 2001), which outputs lemma (F5) and inflection information (F6) of the candidate token. The latter is the ending usually added to the word root to convey inflectional information. It includes for example the *-ing* and *-ed* suffixes in verb endings as well as the *-s* to form the plural of nouns. In our

[2] We extracted this feature using the Chunklink.pl script made available by Sabine Buchholz at `http://ilk.uvt.nl/team/sabine/chunklink/README.html`

example sentence, this feature would be for example *s* for "traders" and "heads", etc.

As for features (F7) and (F8), they rely on information about the main verb of the current sentence. More specifically, feature (F7) is the main verb token (i.e. *shook* in our example), extracted following the head-finding strategy by Yamada and Matsumoto (2003), while feature (F8) is a boolean feature that indicates for each token if it is the main verb in the sentence or not.[3]

The previous sentence feature "Prev" (F9) is a connective-surface feature and is used to capture if the following sentence begins with a connective. Our intuition is that it may be relevant to detect Arg1 boundaries in inter-sentential relations. The feature value for each candidate token of a sentence corresponds to the connective token that appears at the beginning of the following sentence, if any. Otherwise, it is equal to 0.

We also add gold-standard Arg2 labels (F10) as an extra information for Arg1 identification.

6 Experiments

All data used in our experiments are taken from PTB and PDTB. In particular, folders $02-22$ are used to train the model, while folders $00-01$ belong to the development set, and folders 23 and 24 are meant for testing. Our goal is to classify discourse arguments given the connectives by focusing on one relation at time. Since this results in a large search space for the classifier, we prune the search space trying to preserve the relevant contextual information related to the arguments. For this reason, the data given as input to the classifier include a window of two sentences before and after the given connective. This allows us to reduce the search space by more than 90%. In Table 3 we give the statistics of the explicit relation instances for the whole PDTB corpus and span limit sets. Most of the explicit relations (95%) occur within the five sentence window (two preceding and two following the sentence including the connective token).

We use the CRF++ tool (http://crfpp.sourceforge.net/) for sequence labeling classification (Lafferty et al., 2001), with second-order Markov dependency between tags. Beside the individual specification of a feature in the feature description template, the features in various combinations are also represented. We used this tool because the output of CRF++ is compatible to CoNLL 2000 chunking shared task, and we view our task as a discourse chunking task. On the other hand, linear-chain CRFs for sequence labeling offer advantages over both generative models like HMMs and classifiers applied at each sequence position. Also Sha and Pereira (2003) claim that, as a single model, CRFs outperform other models for shallow parsing.

Number of all explicit relations in PDTB	18459
Number of explicit relations with Arg1 entirely *inside* the window	94%
Number of explicit relations with Arg1 entirely *inside or overlapping* the window	95%

Table 3: Statistics about explicit relations and Arg1 extension.

6.1 Evaluation methodology

We present our results using precision, recall and F1 measures. Following Johansson and Moschitti (2010), we use three scoring schemes: *exact*, *intersection* (or *partial*), and *overlap* scoring. In the exact scoring scheme, a span extracted by the system is counted as correct if its extent exactly coincides with one in the gold standard. However, we also use the two other scoring schemes since exact scoring may be uninformative in some situations where it is enough to have a rough approximation of the argument spans. In the overlap scheme, an expression is counted as correctly detected if it overlaps with a gold standard argument, i.e. if their intersection is nonempty. The intersection scheme assigns a score between 0 and 1 for every predicted span based on how much it overlaps with a gold standard span, so unlike the other two schemes it will reward close matches.

6.2 Feature analysis

Our feature set includes a small set of lexical, syntactic and semantic features, which convey the essential information needed to represent the arguments' position and the clausal boundaries, as well as the internal clause structure. We first take into account the features commonly used in similar works, for example by Wellner and Pustejovsky (2007) and Elwell and Baldridge (2008), and then carry out a selection step in order to identify only the feature combination that performs best in our parsing task. Note that both Wellner and Puste-

[3] We used the head rules by Yamada & Matsumoto (http://www.jaist.ac.jp/~h-yamada/)

jovsky (2007) and Elwell and Baldridge (2008) limit their classification to argument heads, thus they may employ features that are not very relevant to our approach.

We follow the hill-climbing (greedy) feature selection technique proposed by Caruana and Freitag (1994). In this optimization scheme, the best-performing set of features is selected on the basis of the best F1 "exact" scores. Therefore, we increase the number of features at each step, and report the corresponding performance. In order to understand better the contribution of each feature and also to avoid sub-optimal solutions, we also run an ablation test by leaving out one feature in turn from the best-performing set. We use the development split to generate results for the feature analysis to find the best performing feature set, whereas the train split is used to built model. Final results are generated using only the test split.

The results of our feature analysis are reported in Table 4 for `Arg2` and Table 5 for `Arg1`. We do not report the scores having zero as F1-measure.

Features	P	R	F1
Features in Isolation			
Token (T)	0.25	0.08	0.13
Connective (CONN)	0.58	0.50	0.54
IOB_Chain (IOB)	0.22	0.06	0.10
PoS	0.26	0.03	0.05
Lemma (L)	0.26	0.09	0.13
Morph(L+INFL)	0.27	0.05	0.09
Hill-Climbing Feature Analysis			
T+CONN	0.80	0.73	0.76
T+CONN+IOB	0.83	0.75	0.79
T+CONN+IOB+Morph	**0.84**	**0.76**	**0.80**
T+CONN+IOB+Morph+Prev	0.83	0.75	0.79
T+CONN+IOB+Morph+Prev+PoS	0.85	0.75	0.79
Token+CONN+IOB+PoS +Morph+BMV+Prev	0.84	0.74	0.78
Token+CONN+IOB+PoS +Morph+MV+BMV+Prev	0.82	0.72	0.77
Feature Ablation			
T+CONN+IOB	0.83	0.75	0.79
T+CONN+Morph	0.80	0.69	0.74
IOB+CONN+Morph	0.84	0.72	0.77
T+IOB+Morph	0.29	0.16	0.20

Table 4: Results with Single and Combined Features for `Arg2`

Both the feature-in-isolation procedure and the ablation test show that the connective sense feature is the most relevant feature for `Arg1` and `Arg2`, whereas the analysis results for `Arg1` show that the "Prev" feature is also important.

We observe that the performance of the lemma increases if integrated with the inflection feature, while inflection in isolation scores a null Precision, Recall and F1. Therefore, we consider lemma and inflection together as a single feature, which we call *Morph*.

We show that the best performing set for `Arg1` includes eight features, whereas the best feature combination for `Arg2` classification is achieved using only four features, namely token, IOB chain, connective sense and *Morph*.

Features	P	R	F1
Features in Isolation			
Token (T)	0.29	0.03	0.05
Connective (CONN)	0.40	0.08	0.14
IOB_Chain (IOB)	0.18	0.04	0.06
PoS	0.14	0.00	0.01
Lemma (L)	0.26	0.03	0.05
Morph(L+INFL)	0.27	0.02	0.03
Prev_feat(PREV)	0.57	0.09	0.16
Hill-Climbing Feature Analysis			
T+CONN	0.62	0.30	0.40
T+CONN+IOB	0.65	0.32	0.44
T+CONN+IOB+Prev	0.69	0.45	0.55
T+CONN+IOB+Arg2+Prev	0.69	0.50	0.58
T+CONN+IOB+BMV+Arg2+Prev	0.70	0.50	0.58
T+CONN+IOB+BMV +Arg2+Prev+Morph	**0.73**	**0.50**	**0.60**
T+CONN+IOB+BMV+Prev +Morph+PoS+Arg2	0.72	0.51	0.59
Token+CONN+IOB+PoS+Prev +Morph+MV+BMV+Arg2	0.69	0.50	0.58
Feature Ablation			
T+CONN+IOB+BMV+Morph+Prev	0.70	0.44	0.54
T+CONN+IOB+BMV+Prev+Arg2	0.70	0.50	0.58
T+CONN+IOB+BMV+Morph+Arg2	0.69	0.38	0.50
T+CONN+IOB+Prev+Morph+Arg2	0.72	0.51	0.60
T+CONN+BMV+Morph+Prev+Arg2	0.69	0.46	0.55
T+IOB+BMV+Morph+Prev+Arg2	0.62	0.36	0.45
CONN+IOB+BMV+Morph+Prev+Arg2	0.70	0.50	0.59

Table 5: Results with Single and Combined Features for `Arg1`

The best combination for `Arg1` classification includes all features from our initial set described in Table 2, except MV and PoS. This is probably due to the fact that PoS information becomes redundant for the classifier and BMV and MV convey the same kind of information.

6.3 Results

We compute a baseline (Table 6 between parenthesis) for each parsing subtask, i.e. `Arg1` and `Arg2` identification with the test dataset. To obtain this baseline, we take into account that *i*) `Arg2` is the argument immediately adjacent to the connective and *ii*) 90% of the relations in PDTB are either intra-sentential or involve two contiguous sentences. Thus, Arg2 baseline is computed by labeling as Arg2 the text span between the connective and the beginning of the next sentence. The other baseline, on the other hand, is computed by labeling as Arg1 all tokens in the text span from the end of the previous sentence to the connective position. In case the connective occurs at the beginning of a sentence, then the baseline classifier tags the previous sentence as Arg1.

		P	R	F1
Arg2	**Exact**	**0.83** (0.53)	**0.75** (0.46)	**0.79** (0.49)
	Partial	0.93 (0.80)	0.84 (0.85)	0.88 (0.82)
	Overlap	0.97 (0.98)	0.88 (0.85)	0.92 (0.91)
Arg1 +Prev	**Exact**	**0.70** (0.19)	**0.48** (0.19)	**0.57** (0.19)
	Partial	0.83 (0.50)	0.62 (0.68)	0.71 (0.58)
	Overlap	0.91 (0.70)	0.63 (0.68)	0.74 (0.69)
Arg1 -Prev	**Exact**	**0.70** (0.19)	**0.38** (0.19)	**0.50** (0.19)
	Partial	0.83 (0.50)	0.49 (0.68)	0.62 (0.58)
	Overlap	0.92 (0.70)	0.50 (0.68)	0.65 (0.69)

Table 6: Results of Arg1 and Arg2 extraction with test dataset. Baseline results between parentheses.

In Table 6 we report for each parsing subtask Precision, Recall and F1 achieved with the best performing feature set (see Section 6.2) using the test split, with the corresponding baseline between parenthesis. Note that before evaluation, all spans were normalized by removing leading or trailing punctuation. The best results and features are highlighted in Table 4 and 5 for Arg2 and Arg1 respectively.

We compute the confidence intervals using a resampling method (Hjorth, 1993). For Arg1 identification, we observe that the confidence interval (95%) without "Prev" feature ranges from 0.48 to 0.52 and the same interval is between 0.55 and 0.59 with "Prev" feature, if the exact F1 measure is taken into account. For Arg2 identification the confidence interval (95%) is between 0.78 and 0.81, when the exact F1 measure is taken into account. A statistical significance test run on previous and current results of Arg1 identification shows also that the difference is significant ($p < 0.0001$).

We observe in the results that recall is consistently lower than precision in all tables. This is probably due to the fact that CRF is more conservative while tagging data with argument label compared to other classifiers, which may lead to a lower coverage.

As expected, Arg2 parsing subtask achieves a better performance than Arg1 subtask because Arg2 position and extension are easier to predict. This is confirmed by the fact that the baseline precision of Arg2 *overlap* is 0.98. Also, the major improvement w.r.t. the baseline is achieved in the *exact* setting.

6.4 Error Analysis

We carry out a further analysis on the test set in order to characterize parser errors on different test set partitions. Since Arg1 may occur in a previous sentence w.r.t. the connective, we want to assess the impact of Arg1 position on the parsing task. Therefore, we separately evaluate Arg1 precision, recall and F1 on intra-sentential and inter-sentential discourse relations. Results are reported in Table 7. We also show the changes before and after adding the lexical feature targeting inter-sentential cases.

		Arg1-Results		
		P	R	F1
Intra-Sentential w/o Prev_feat	Exact	0.73	0.61	0.66
	Partial	0.86	0.77	0.81
	Overlap	0.95	0.78	0.86
Inter-Sentential w/o Prev_feat	Exact	0.19	0.01	0.02
	Partial	0.27	0.02	0.04
	Overlap	0.31	0.02	0.04
Intra-Sentential with Prev_feat	Exact	0.77	0.61	0.68
	Partial	0.88	0.79	0.81
	Overlap	0.96	0.77	0.85
Inter-Sentential with Prev_feat	Exact	0.52	0.27	0.36
	Partial	0.68	0.40	0.50
	Overlap	0.79	0.40	0.54

Table 7: Results of Arg1 parsing for intra- and inter- sentential partitions. In the test set, the number of intra- and inter- sentential relations are 1028 and 617 respectively.

The "Prev" feature is critical to the parser to achieve reasonable baseline Arg1 performance for the inter-sentential partition of the test set.

We also carry out a comparative analysis of the parsing performance in the *exact* evaluation setting by considering separately coordinating, subordinating and adverbial connectives. We make the above-mentioned distinction following the suggestion by Elwell and Baldridge (2008), because each connective type has a different behavior w.r.t. its arguments: coordinating connectives (e.g. *and*, *but*) usually have syntactically similar arguments, subordinating ones (e.g. *since*, *before*) are dominated or adverbially linked to Arg1 and are syntactically bound to Arg2, while adverbial connectives (i.e. *nevertheless*, *for instance*) can occur in different positions in the sentence and are not necessarily bound to Arg1.

The evaluation results are presented in Table 8.

In previous works, e.g. Elwell and Baldridge (2008), adverbial connectives were usually considered the most difficult connective type to classify. This is confirmed by our results obtained on Arg1, which show that adverbial connectives negatively affect both precision and recall, with a higher impact on recall. As for Arg2, the parsing results on the three connective types are more homogeneous.

We also observe that the "Prev" feature significantly improves Arg1 parsing with any connective type because it increases recall, while precision decreases with coordinating and adverbial connectives.

Conn. Type	P	R	F1
Results for Arg2			
Coordinating	0.81	0.75	0.78
Subordinating	0.86	0.78	0.82
Adverbial	0.83	0.74	0.78
Results for Arg1(w/o Prev)			
Coordinating	0.73	0.42	0.54
Subordinating	0.73	0.45	0.56
Adverbial	0.68	0.26	0.37
Results for Arg1 (with Prev)			
Coordinating	0.69	0.59	0.64
Subordinating	0.76	0.50	0.61
Adverbial	0.64	0.34	0.44

Table 8: Exact evaluation for each connective type. Coordinating connectives appear in around 40% of the relations, while subordinating and adverbials are respectively 25% and 35% of all connectives.

In order to understand the most common mistakes done by the classifier, we present two example relations where resp. Arg1 (e) and Arg2 (f) are wrongly identified[4]. Note that in example (f) Arg1 appears in the previous sentence, which we do not report here.

(e) Many analysts said the September increase was a one-time event, *coming* as **dealers introduced their 1990 models** [CONTINGENCY]

(f) However, Jeffrey Lane, president of Shearson Lehman Hutton, said **that Friday's plunge is "going to set back" relations with customers**, "because it reinforces the concern of volatility [COMPARISON]

In (e), the classifier tagged the whole text from "the September" to "coming" as Arg1 instead of only "coming", since it takes clausal boundaries as a relevant factor for identifying the argument spans. In (f) the classifier is unable to detect Arg2 probably because the argument does not occur immediately next to the connective.

A manual inspection of misclassified relations confirms that the parser is more accurate in the identification of the sentences containing the arguments rather than in the detection of their exact spans. Also, mistakes concern mostly the classification of inter-sentential relations (especially as regards the Arg1 classifier), thus we will need to focus on these specific cases for future improvements.

7 Conclusions

We cast the complex task of discourse argument parsing as a set of cascading subtasks to be tackled in sequence, and we showed that in this way we achieved a reasonable parser accuracy by handling the whole labeling process in a pipeline.

Since we consider this discourse parsing task as a token-level sequence-labeling task, we were able to detect connective arguments and the corresponding boundaries avoiding the computationally complex approaches described in previous works.

We trained a CRF classifier with lexical, syntactic and semantic features extracted from PDTB and PTB gold annotation. We tested these features both in isolation and in different combinations in order to achieve an optimized performance. To make training time manageable, we pruned the search space by 90%, though leaving out only around 5% of all Arg1 in PDTB.

We also presented a comparative error analysis (subsection 6.4), where we showed that Arg1 classification on intra-sentential relations achieves a performance comparable to Arg2 classification (Table 6). Since the main open issue in our approach is the correct classification of Arg1 in inter-sentential relations, we plan to improve it through more feature engineering. We already extended our experimental framework by including automatically annotated parse trees and connectives in the pipeline (Ghosh et al., 2011).

8 Acknowledgements

This work was partially funded by the LiveMemories project (www.livememories.org). Richard Johansson was funded by the EC FP7 under grant 231126: LivingKnowledge – Facts, Opinions and Bias in Time.

[4] The examples show the gold standard annotation.

References

Rich Caruana and Dayne Freitag. 1994. Greedy attribute selection. In *Proceedings of the Eleventh International Conference on Machine Learning*.

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 Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, pages 29–36, Ann Arbor, Michigan, June.

Robert Elwell and Jason Baldridge. 2008. Discourse connective argument identification with connective specific rankers. In *Proceedings of ICSC-2008*, Santa Clara, United States.

Sucheta Ghosh, Sara Tonelli, Giuseppe Riccardi, and Richard Johansson. 2011. End-to-end discourse parser evaluation. In *Proceedings of 5th IEEE International Conference on Semantic Computing*, Palo Alto, CA, USA.

J. S. Urban Hjorth. 1993. *Computer Intensive Statistical Methods*. Chapman and Hall, London.

Richard Johansson and Alessandro Moschitti. 2010. Syntactic and semantic structure for opinion expression detection. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 67–76.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *18th International Conf. on Machine Learning*. Morgan Kaufmann.

Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing implicit discourse relations in the Penn Discourse Treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 343–351, Singapore.

William Mann and Sara Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*.

Emily Pitler and Ani Nenkova. 2009. Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing*.

Emily Pitler, Mridhula Raghupathy, Hena Mehta, Ani Nenkova, Alan Lee, and Aravind Joshi. 2008. Easily identifiable discourse relations. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 87–90, Manchester, United Kingdom.

Emily Pitler, Annie Louis, and Ani Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 683–691, Suntec, Singapore.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse Treebank 2.0. In *Proceedings of the 6th International Conference on Languages Resources and Evaluations (LREC 2008)*, Marrakech, Morocco.

Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2010. Exploiting scope for shallow discourse parsing. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of HLT/NAACL*, pages 213–220.

Swapna Somasundaran, Galileo Namata, Janyce Wiebe, and Lise Getoor. 2009. Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 170–179, Singapore.

Sara Tonelli, Giuseppe Riccardi, Rashmi Prasad, and Aravind Joshi. 2010. Annotation of discourse relations for conversational spoken dialogs. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

Bonnie Webber, Markus Egg, and Valia Kordoni. 2010. Discourse Structure and Language Technology. *Natural Language Engineering*, 1(1):1–49.

Ben Wellner and James Pustejovsky. 2007. 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, Prague, Czech Republic.

Ben Wellner. 2009. *Sequence Models and Ranking Methods for Discourse Parsing*. Ph.D. thesis, Brandeis University.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of 8th International Workshop on Parsing Technologies*.