Comparative Evaluation of Argument Extraction Algorithms in Discourse Relation Parsing

Evgeny A. Stepanov and **Giuseppe Riccardi**
Department of Information Engineering and Computer Science,
University of Trento, Trento, Italy
{stepanov,riccardi}@disi.unitn.it

Abstract

Discourse relation parsing is an important task with the goal of understanding text beyond the sentence boundaries. One of the subtasks of discourse parsing is the extraction of argument spans of discourse relations. A relation can be either intra-sentential – to have both arguments in the same sentence – or inter-sentential – to have arguments span over different sentences. There are two approaches to the task. In the first approach the parser decision is not conditioned on whether the relation is intra- or inter-sentential. In the second approach relations are parsed separately for each class. The paper evaluates the two approaches to argument span extraction on Penn Discourse Treebank explicit relations; and the problem is cast as token-level sequence labeling. We show that processing intra- and inter-sentential relations separately, reduces the task complexity and significantly outperforms the single model approach.

1 Introduction

Discourse analysis is one of the most challenging tasks in Natural Language Processing, that has applications in many language technology areas such as opinion mining, summarization, information extraction, etc. (see (Webber et al., 2011) and (Taboada and Mann, 2006) for detailed review). With the availability of annotated corpora, such as Penn Discourse Treebank (PDTB) (Prasad et al., 2008), statistical discourse parsers were developed (Lin et al., 2012; Ghosh et al., 2011; Xu et al., 2012).

PDTB adopts non-hierarchical binary view on discourse relations: Argument 1 (*Arg1*) and Argument 2 (*Arg2*), which is syntactically attached to a discourse connective. Thus, PDTB-based discourse parsing can be roughly partitioned into discourse relation detection, argument position classification, argument span extraction, and relation sense classification. For discourse relations signaled by a connective (explicit relations), discourse relation detection is cast as classification of connectives as discourse and non-discourse. Argument position classification involves detection of the location of *Arg1* with respect to *Arg2*: usually either the same sentence (SS) or previous ones (PS).[1] Argument span extraction, on the other hand, is extraction (labeling) of text segments that belong to each of the arguments. Finally, relation sense classification is the annotation of relations with the senses from PDTB.

Since arguments of explicit discourse relations can appear in the same sentence or in different ones (i.e. relations can be intra- or inter-sentential); there are two approaches to argument span extraction. In the first approach the parser decision is not conditioned on whether the relation is intra- or inter-sentential (e.g. (Ghosh et al., 2011)). In the second approach relations are parsed separately for each class (e.g. (Lin et al., 2012; Xu et al., 2012)). In the former approach argument span extraction is applied right after discourse connective detection, while the latter approach also requires argument position classification.

The decision on argument span can be made on different levels: from token-level to sentence-level. In (Ghosh et al., 2011) the decision is made on token-level, and the problem is cast as sequence labeling using conditional random fields (CRFs) (Lafferty et

[1] We use the term *inter-sentential* to refer to a set of relations that includes both previous sentence (*PS*) and following sentence (*FS*) *Arg1*. *Intra-sentential* and same sentence (*SS*) relations, on the other hand, are the same set.

al., 2001). In this paper we focus on argument span extraction, and extend the token-level sequence labeling approach of (Ghosh et al., 2011) with the separate models for arguments of intra-sentential and inter-sentential explicit discourse relations. To compare to the other approaches (i.e. (Lin et al., 2012) and (Xu et al., 2012)) we adopt the immediately previous sentence heuristic to select a candidate *Arg1* sentence for the inter-sentential relations. Additionally to the heuristic, we train and test CRF argument span extraction models to extract *exact* argument spans.

The paper is structured as follows. In Section 2 we briefly present the corpus that was used in the experiments – Penn Discourse Treebank. Section 3 describes related works. Section 4 defines the problem and assesses its complexity. In Section 5 we describe argument span extraction cast as the token-level sequence labeling; and in Section 6 we present the evaluation of the two approaches – either single or separate processing of intra- and inter-sentential relations. Section 7 provides concluding remarks.

2 The Penn Discourse Treebank

The Penn Discourse Treebank (PDTB) (Prasad et al., 2008) is a corpus that contains discourse relation annotation on top of WSJ corpus; and it is aligned with Penn Treebank (PTB) syntactic tree annotation. Discourse relations in PDTB are binary: *Arg1* and *Arg2*, where *Arg2* is an argument syntactically attached to a discourse connective. With respect to *Arg2*, *Arg1* can appear in the same sentence (SS case), one of the preceding (PS case) or following (FS case) sentences.

A discourse connective is a member of a well defined list of 100 connectives and a relation expressed via such connective is an *Explicit* relation. There are other types of discourse and non-discourse relations annotated in PDTB; however, they are out of the scope of this paper. Discourse relations are annotated using 3-level hierarchy of senses. The top level (level 1) senses are the most general: *Comparison*, *Contingency*, *Expansion*, and *Temporal* (Prasad et al., 2008).

3 Related Work

Pitler and Nenkova (2009) applied machine learning methods using lexical and syntactic features and achieved high classification performance on discourse connective detection task (F_1: 94.19%, 10 fold cross-validation on PDTB sections 02-22). Later, Lin et al. (2012) achieved an improvement with additional lexico-syntactic and path features (F_1: 95.76%).

After a discourse connective is identified as such, it is classified into relation senses annotated in PDTB. Pitler and Nenkova (2009) classify discourse connectives into 4 top level senses – *Comparison*, *Contingency*, *Expansion*, and *Temporal* – and achieve accuracy of 94.15%, which is slightly above the inter-annotator agreement. In this paper we focus on the parsing steps after discourse connective detection; thus, we use gold reference connectives and their senses as features.

The approaches used for the argument position classification even though useful, are incomplete as they do not make decision on argument spans. (Wellner and Pustejovsky, 2007) and (Elwell and Baldridge, 2008), following them, used machine learning methods to identify head words of the arguments of explicit relations expressed by discourse connectives. (Prasad et al., 2010), on the other hand, addressed a more difficult task of identification of sentences that contain *Arg1* for cases when arguments are located in different sentences.

Dinesh et al. (2005) and Lin et al. (2012) approach the problem of argument span extraction on syntactic tree node-level. In the former, it is a rule based system that covers limited set of connectives; whereas in the latter it is a machine learning approach with full PDTB coverage. Both apply syntactic tree subtraction to get argument spans. Xu et al. (2012) approach the problem on a constituent-level: authors first decide whether a constituent is a valid argument and then whether it is *Arg1*, *Arg2*, or neither. Ghosh et al. (2011) (and further (Ghosh et al., 2012a; Ghosh et al., 2012b)), on the other hand, cast the problem as token-level sequence labeling. In this paper we follows the approach of (Ghosh et al., 2011).

4 Problem Definition

In the introduction we mentioned Immediately Previous Sentence Heuristic for *Arg1* of inter-sentential explicit relations and Argument Position Classification as a prerequisite for processing intra- and inter-sentential relations separately. In this section we analyze PDTB to assess the complexity and potential accuracy of the heuristic and the classification task.

	SingFull		SingPart		MultFull		MultPart		Total	
ARG1										
IPS	**3,192**	(44.2%)	**1,880**	(26.0%)	370	(5.1%)	107	(1.5%)	5,549	(76.8%)
NAPS	993	(13.8%)	551	(7.6%)	71	(1.0%)	51	(0.7%)	1,666	(23.1%)
FS	2	(0.0%)	0	(0.0%)	1	(0.0%)	5	(0.0%)	8	(0.1%)
Total	4,187	(58.0%)	2,431	(33.7%)	442	(6.1%)	163	(2.3%)	7,223	(100%)
ARG2										
SS/Total	5,181	(71.7%)	1,936	(26.8%)	84	(1.2%)	22	(0.3%)	7,223	(100%)

Table 1: Distribution of *Arg1* with respect to the location (rows) and extent (columns) (partially copied from (Prasad et al., 2008)); and distribution of *Arg2* with respect to extent **in inter-sentential explicit discourse relations**.
SS = same sentence as the connective; IPS = immediately previous sentence; NAPS = non-adjacent previous sentence; FS = some sentence following the sentence containing the connective; SingFull = Single Full sentence; SingPart = Part of single sentence; MultFull = Multiple full sentences; MultPart = Parts of multiple sentences.

4.1 Immediately Previous Sentence Heuristic

According to Prasad et al. (2008)'s analysis of explicit discourse relations annotated in PDTB, out of 18,459 relations, 11,236 (60.9%) have both of the arguments in the same sentence (SS case), 7,215 (39.1%) have *Arg1* in the sentences preceding the *Arg2* (PS case), and only 8 instances have *Arg1* in the sentences following *Arg2* (FS case). Since FS case has too few instances it is usually ignored. For the PS case, the *Arg1* is located either in Immediately Previous Sentences (IPS: 30.1%) or in some Non-Adjacent Previous Sentences (NAPS: 9.0%).

CRF-based discourse parser of Ghosh et al. (2011), which processes SS and PS cases with the same model, uses ± 2 sentence window as a hypothesis space (5 sentences: 1 sentence containing the connective, 2 preceding and 2 following sentences). The window size is motivated by the observation that it entirely covers arguments of 94% of all explicit relations. The authors also report that the performance of the parser on inter-sentential relations (i.e. mainly PS case) has F-measure of 36.0. However, since in 44.2% of inter-sentential explicit discourse relations *Arg1* fully covers the sentence immediately preceding *Arg2* (see Table 1 partially copied from (Prasad et al., 2008)), the heuristic that selects the immediately previous sentence and tags all of its tokens as *Arg1* already yields F-measure of 44.2 over all PDTB (the performance on the test set may vary).

The same heuristic is mentioned in (Lin et al., 2012) and (Xu et al., 2012) as a majority classifier for the relations with *Arg1* in previous sentences.

Compared to the ± 2 window, the heuristic covers *Arg1* of only 88.4% explicit discourse relations (60.9% SS + 27.5% PS); since it ignores all the relations with *Arg1* in Non-Adjacent Previous Sentences (NAPS) (9.0% of all explicit relations), and does not accommodate *Arg1* spanning multiple immediately preceding sentences (2.6% of all explicit relations). Nevertheless, 70.2% of all PS explicit relations have *Arg1* entirely inside the immediately previous sentence. Thus, the integration of the heuristic is expected to improve the argument span extraction performance for inter-sentential *Arg1*.

In 98.5% of all PS cases *Arg2* is within the sentence containing the connective (remaining 1.5% are multi-sentence *Arg2*); and in 71.7% of all PS cases it fully covers the sentence containing the discourse connective (see Table 1). Thus, similar heuristic for *Arg2* is to tag all the tokens of the sentence except the connective as *Arg2*.

For the heuristics to be applicable, a discourse connective has to be classified as requiring its *Arg1* in the same sentence (SS) or the previous ones (PS), i.e. it requires argument position classification.

4.2 Argument Position Classification

Explicit discourse connectives, annotated in PDTB, belong to one of the three syntactic categories: (1) subordinating conjunctions (e.g. *when*), (2) coordinating conjunctions (e.g. *and*), and (3) discourse adverbials (e.g. *for example*). With few exceptions, a discourse connective belongs to a single syntactic category (see Appendix A in (Knott, 1996)). Each of these syntactic categories has a strong preference on the po-

	Sentence Initial				Sentence Medial			
	SS		PS		SS		PS	
Coordinating	10	(0.05%)	2,869	(15.54%)	3,841	(20.81%)	202	(1.09%)
Subordinating	1,402	(7.60%)	114	(0.62%)	5,465	(29.61%)	83	(0.45%)
Discourse Adverbial	13	(0.07%)	1,632	(8.84%)	495	(2.68%)	2,325	(12.60%)

Table 2: Distribution of discourse connectives in PDTB with respect to syntactic category (rows) and position in the sentence (columns) and the location of *Arg1* as in the same sentence (SS) as the connective or the previous sentences (PS). The case when *Arg1* appears in some following sentence (FS) is ignored, since it has only 8 instances.

sition of *Arg1*, depending on whether the connective appears sentence-initially or sentence-medially. Here, a connective is considered sentence-initial if it appears as the first sequence of words in a sentence. Table 2 presents the distribution of discourse connectives in PDTB with respect to the syntactic categories, their position in the sentence, and having *Arg1* in the same or previous sentences. The distribution of sentence-medial discourse adverbials, which is the most ambiguous class, between SS and PS cases is 17.5% to 82.5%; for all other classes it higher than 90% to 10%. Thus, the overall accuracy of the SS vs. PS majority classification using just syntactic category and position information is already 95.0%.

When analyzed on per connective basis, the observation is that some connectives require *Arg1* in the same or previous sentence irrespective of their position in the sentence. For instance, sentence-initial subordinating conjunction *so* always has its *Arg1* in the previous sentence; and the parallel sentence-initial subordinating conjunction *if..then* in the same sentence. Others, such as sentence-medial adverbials *however* and *meanwhile* mainly require their *Arg1* in the previous sentence. Even though low, there is still an ambiguity: e.g. for sentence-medial adverbials *also*, *therefore*, *still*, *instead*, *in fact*, etc. *Arg1* appears in SS and PS cases evenly. Consequently, assigning the position of the *Arg1* considering the discourse connective, together with its syntactic category and its position in the sentence, for PDTB will be correct in more than 95% of instances.

In the literature, the task of argument position classification was addressed by several researchers (e.g. (Prasad et al., 2010), (Lin et al., 2012)). Lin et al. (2012), for instance, report F_1 of 97.94% for a classifier trained on PDTB sections 02-21, and tested on section 23. The task has a very high baseline and even higher performance on supervised machine learning, which is an additional motivation to process intra- and inter-sentential relations separately.

Feature	**ABBR**	**Arg2**	**Arg1**
Token	TOK	Y	Y
POS-Tag	POS		
Lemma	LEM	Y	Y
Inflection	INFL	Y	Y
IOB-Chain	IOB	Y	Y
Connective Sense	CONN	Y	Y
Boolean Main Verb	BMV		Y
Prev. Sent. Festure	PREV		Y
Arg2 Label	ARG2		Y

Table 3: Feature sets for *Arg2* and *Arg1* argument span extraction in (Ghosh et al., 2011)

5 Parsing Models

We replicate and evaluate the discourse parser of (Ghosh et al., 2011), then modify it to process intra- and inter-sentential explicit relations separately. This is achieved by integrating Argument Position Classification and Immediately Previous Sentence heuristic into the parsing pipe-line.

Since the features used to train argument span extraction models for both approaches are the same, we first describe them in Subsection 5.1. Then we proceed with the description of the single model discourse parser (our baseline) and separate models discourse parser, Subsections 5.2 and 5.3, respectively.

5.1 Features

The features used to train the models for *Arg1* and *Arg2* are given in Table 3. Besides the token itself (*TOK*), the rest of the features is described below.

Lemma (*LEM*) and *inflectional* affixes (*INFL*) are extracted using *morpha* tool (Minnen et al., 2001), that requires token and its POS-tag as input. For instance, for the word *flashed* the lemma and infection

features are '*flash*' and '*+ed*', respectively.

IOB-Chain (*IOB*) is the path string of the syntactic tree nodes from the root node to the token, prefixed with the information whether a token is at the beginning (B-) or inside (I-) the constituent. The feature is extracted using the *chunklink* tool (Buchholz, 2000). For example, the IOB-Chain '*I-S/B-VP*' indicates that a token is the first word of the verb phrase (*B-VP*) of the main clause (*I-S*).

PDTB Level 1 Connective sense (*CONN*) is the most general sense of a connective in PDTB sense hierarchy: one of *Comparison*, *Contingency*, *Expansion*, or *Temporal*. For instance, a discourse connective *when* might have the CONN feature '*Temporal*' or '*Contingency*' depending on the discourse relation it appears in, or '*NULL*' in case of non-discourse usage. The value of the feature is '*NULL*' for all tokens except the discourse connective.

Boolean Main Verb (*BMV*) is a feature that indicates whether a token is a main verb of a sentence or not (Yamada and Matsumoto, 2003). For instance in the sentence *Prices collapsed when the news flashed*, the main verb is *collapsed*; thus, its BMV feature is '1', whereas for the rest of tokens it is '0'.

Previous Sentence Feature (*PREV*) signals if a sentence immediately precedes the sentence starting with a connective, and its value is the first token of the connective (Ghosh et al., 2011). For instance, if some sentence *A* is followed by a sentence *B* starting with discourse connective *On the other hand*, all the tokens of the sentence *A* have the *PREV* feature value '*On*'. The feature is similar to a heuristic to select the sentence immediately preceding a sentence starting with a connective as a candidate for *Arg1*.

Arg2 Label (*ARG2*) is an output of *Arg2* span extraction model, and it is used as a feature for *Arg1* span extraction. Since for sequence labeling we use IOBE (Inside, Out, Begin, End) notation, the possible values of *ARG2* are IOBE-tagged labels, i.e. '*ARG2-B*' – if a word is the first word of *Arg2*, '*ARG2-I*' – if a word is inside the argument span, '*ARG2-E*' – if a word is in the last word of *Arg2*, and '*O*' otherwise.

CRF++[2] – conditional random field implementation we use – allows definition of feature templates. Via templates these features are enriched with n-grams: tokens with 2-grams in the window of ± 1 tokens, and the rest of the features with 2 & 3-grams in the window of ± 2 tokens.

[2]https://code.google.com/p/crfpp/

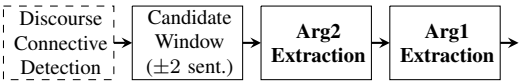

Figure 1: Single model discourse parser architecture of (Ghosh et al., 2011). CRF argument span extraction models are in bold.

For instance, labeling a token as *Arg2* is an assignment of one of the four possible labels: ARG2-B, ARG2-I, ARG2-E and O (ARG2 with IOBE notation). The feature set (token, lemma, inflection, IOB-chain and connective sense (see Table 3)) is expanded by CRF++ via template into 55 features ($5 * 5$ unigrams, 2 token bigrams, $4 * 4$ bigrams and $4 * 3$ trigrams of other features).

5.2 Single Model Discourse Parser

The discourse parser of (Ghosh et al., 2011) is a cascade of CRF models to sequentially label *Arg2* and *Arg1* spans (since *Arg2* label is a feature for *Arg1* model) (see Figure 1). There is no distinction between intra- and inter-sentential relations, rather the single model jointly decides on the position and the span of an argument (either *Arg1* or *Arg2*, not both together) in the window of ± 2 sentences (the parser will be further abbreviated as *W5P* – Window 5 Parser).

The single model parser achieves F-measure of 81.7 for *Arg2* and 60.3 for *Arg1* using CONNL evaluation script. The performance is higher than (Ghosh et al., 2011) – *Arg2*: F_1 of 79.1 and *Arg1*: F_1 of 57.3 – due to improvements in feature and instance extraction, such as the treatment of multi-word connectives. These models are the baseline for comparison with separate models architecture. However, we change the evaluation method (see Section 6).

5.3 Separate Models Discourse Parser

Figure 2 depicts the architecture of the discourse parser processing intra- and inter-sentential relations separately. It is a combination of argument position classification with specific CRF models for each of the arguments of SS and PS cases, i.e. there are 4 CRF models – SS *Arg1* and *Arg2*, and PS *Arg1* and *Arg2* (following sentence case (FS) is ignored). SS models are applied in a cascade and, similar to the

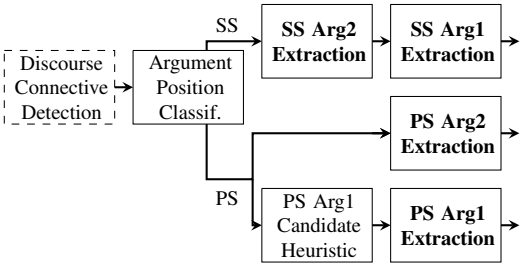

Figure 2: Separate models discourse parsing architecture. CRF argument span extraction models are in bold.

baseline single model parser, *Arg2* label is a feature for *Arg1* span extraction. These SS models are trained using exactly the same features, with the exception of *PREV* feature: since we consider only the sentence containing the connective, it naturally falls out.

For the PS case, we apply a heuristic to select candidate sentences. Based on the observation that in PDTB for the PS case *Arg2* span is fully located in the sentence containing the connective in 98.5% of instances; and *Arg1* span is fully located in the sentence immediately preceding *Arg2* in 71.7% of instances; we select sentences in these positions to train and test respective CRF models. The feature set for *Arg2* remains the same, whereas, from *Arg1* feature set we remove *PREV* and *Arg2* label (since in PS case *Arg2* is in different sentence, the feature will always have the same value of 'O').

For *Argument Position Classification* we train unigram BoosTexter (Schapire and Singer, 2000) model with 100 iterations[3] on PDTB sections 02-22 and test on sections 23-24; and, similar to other researchers, achieve high results: $F_1 = 98.12$. The features are connective surface string, POS-tags, and IOB-chains. The results obtained using automatic features ($F_1 = 97.87$) are insignificantly lower (McNemar's $\chi^2(1, 1595) = 0.75, p = 0.05$); thus, this step will not cause deterioration in performance with automatic features. Here we used Stanford Parser (Klein and Manning, 2003) to obtain POS-tags and automatic constituency-based parse trees.

Since both argument span extraction approaches are equally affected by the discourse connective detection step, we use gold reference connectives. As an alternative, discourse connectives can be detected with high accuracy using addDiscourse tool (Pitler and Nenkova, 2009).

In the separate models discourse parser, the steps of the process to extract argument spans given a discourse connective are as follows:

1. Classify connective as SS or PS;
2. If classified as SS:
   (a) Use SS Arg2 CRF model to label the sentence tokens for *Arg2*;
   (b) Use SS Arg1 CRF model to label the sentence tokens for *Arg1* using *Arg2* label as a feature;
3. If classified as PS
   (a) Select the sentence containing the connective and use PS Arg2 CRF model to label *Arg2* span;
   (b) Select the sentence immediately preceding the *Arg2* sentence and use PS Arg1 CRF model to label *Arg1* span.

The separate model parser with CRF models will be further abbreviated as *SMP*; and with the heuristics for PS case as *hSMP*.

6 Experiments and Results

We first describe the evaluation methodology. Then present evaluation of PS case CRF models against the heuristic. In subsection 6.3 we compare the performance of the single and separate model parsers on SS and PS cases of the test set separately and together. Finally, we compare the results of the separate model parser to (Lin et al., 2012) and (Xu et al., 2012).

6.1 Evaluation

There are two important aspects regarding the evaluation. First, in this paper it is different from (Ghosh et al., 2011); thus, we first describe it and evaluate the difference. Second, in order to compare the baseline single and separate model parsers, the error from argument position classification has to be propagated for the latter one; and the process is described in 6.1.2.

Since both versions of the parser are affected by automatic features, the evaluation is on gold features only. The exception is for *Arg2* label; since it is generated within the segment of the pipeline we are in-

[3]The choice is based on the number of discourse connectives defined in PDTB.

terested in. Unless stated otherwise, all the results for *Arg1* are reported for automatic *Arg2* labels as a feature. Following (Ghosh et al., 2011) PDTB is split as Sections 02-22 for training, 00-01 for development, and 23-24 for testing.

6.1.1 CONLL vs. String-based Evaluation

Ghosh et al. (2011) report using CONLL-based evaluation script. However, it is not well suited for the evaluation of argument spans because the unit of evaluation is a chunk – a segment delimited by any out-of-chunk token or a sentence boundary. However, in PDTB arguments can (1) span over several sentences, (2) be non-contiguous in the same sentence. Thus, CONLL-based evaluation yields incorrect number of test instances: Ghosh et al. (2011) report 1,028 SS and 617 PS test instances for PDTB sections 23-24 (see caption of Table 7 in the original paper), which is 1,645 in total; whereas there is only 1,595 explicit relations in these sections.

In this paper, the evaluation is string-based; i.e. an argument span is correct, if it matches the whole reference string. Following (Ghosh et al., 2011) and (Lin et al., 2012), argument initial and final punctuation marks are removed; and precision (*p*), recall (*r*) and F_1 score are computed using the equations 1 – 3.

$$p = \frac{\text{Exact Match}}{\text{Exact Match} + \text{No Match}} \quad (1)$$

$$r = \frac{\text{Exact Match}}{\text{References in Gold}} \quad (2)$$

$$F_1 = \frac{2 * p * r}{p + r} \quad (3)$$

In the equations, *Exact Match* is the count of correctly tagged argument spans; *No Match* is the count of argument spans that do not match the reference string exactly (even one token difference is counted as an error); and *References in Gold* is the total number of arguments in the reference.

String-based evaluation of the single model discourse parser with gold features reduces F_1 for *Arg2* from 81.7 to 77.8 and for *Arg1* from 60.33 to 55.33.

6.1.2 Error Propagation

Since the single model parser applies argument span extraction right after discourse connective detection, whereas in the separate model parser there is an additional step of argument position classification; for the two to be comparable an error from the argument position classification is propagated. Even though, the performance of the classifier is very high (98.12%) there are still some misclassified instances. These instances are propagated to the counts of *Exact Match* and *No Match* of the argument span extraction. For example, if the argument position classifier misclassified an SS connective as PS; in the SS evaluation its *Arg1* and *Arg2* are considered as not recalled regardless of argument span extractor's decision (i.e. neither *Exact Match* nor *No Match*); and in the PS evaluation, they are both considered as *No Match*.

The separate model discourse parser results are reported without error propagation for in-class comparison of the heuristic and CRF models, and with error propagation for cross-class comparison with the single model parser.

	Arg2			Arg1		
	P	**R**	**F1**	**P**	**R**	**F1**
hSMP	74.19	74.19	74.19	39.19	39.19	39.19
SMP	78.61	78.23	78.42	46.81	37.90	41.89

Table 4: Argument span extraction performance of the heuristics (*hSMP*) and the CRF models (*SMP*) on inter-sentential relations (PS case). Results are reported as precision (**P**), recall (**R**) and F-measure (**F1**)

6.2 Heuristic vs. CRF Models

The goal of this section is to assess the benefit of training CRF models for the extraction of exact argument spans of PS *Arg1* and *Arg2* on top of the heuristics. The performance of the heuristics (immediately previous sentence for *Arg1* and the full sentence except the connective for *Arg2*) and the CRF models is reported in Table 4. CRF models perform significantly better for *Arg2* (McNemar's $\chi^2(1, 620) = 7.48, p = 0.05$). Even though, they perform 2.7% better for *Arg1*, the difference is insignificant (McNemar's $\chi^2(1, 620) = 0.66, p = 0.05$). For both arguments, the CRF model results are lower than expected.

6.3 Single vs. Separate Models

To compare the single and the separate model parsers, the results of the former must be split into SS and PS cases. For the latter, on the other hand, we propagate

	Arg2			Arg1		
	P	**R**	**F1**	**P**	**R**	**F1**
W5P	87.57	84.51	86.01	71.73	62.97	67.07
SMP	90.36	87.49	88.90	70.27	66.67	68.42

Table 5: Performance of the single ± 2 window (*W5P*) and separate model (*SMP*) parsers on argument span extraction of SS relations; reported as precision (**P**), recall (**R**) and F-measure (**F1**). For the *SMP* results are with error propagation from argument position classification.

	Arg2			Arg1		
	P	**R**	**F1**	**P**	**R**	**F1**
W5P	71.12	59.19	64.61	40.06	22.74	29.01
hSMP	74.67	72.23	73.94	38.98	38.23	38.60
SMP	79.01	77.10	78.04	46.23	36.61	40.86

Table 6: Performance of the single model parser (*W5P*) and the separate model parser with the heuristics (*hSMP*) and CRF models (*SMP*) on argument span extraction of PS relations; reported as precision (**P**), recall (**R**) and F-measure (**F1**). For the separate model parsers, results include error propagation from argument position classification.

error from the argument position classification step. For the PS case we also report the performance of the heuristic with error propagation.

Table 5 reports the results for the SS case, and Table 6 reports the results for the PS case. In both cases the separate model parser with error propagation from argument position classification step significantly outperforms the single model parser.

The performance of the separate model parsers (reported in Table 7) with heuristics and CRF models on all relations (SS + PS) both are significantly better than the performance of single ± 2 window model parser (for *SMP* McNemar's $\chi^2(1, 1595) = 17.75$ for *Arg2* and $\chi^2(1, 1595) = 19.82$ for *Arg1*, $p = 0.05$).

	Arg2			Arg1		
	P	**R**	**F1**	**P**	**R**	**F1**
W5P	81.47	74.42	77.79	61.90	46.96	53.40
hSMP	84.21	81.94	83.06	57.86	55.61	56.71
SMP	85.93	83.45	84.67	61.94	54.98	58.25

Table 7: Performance of the single model parser (*W5P*) and the separate model parser with the heuristics (*hSMP*) and CRF models (*SMP*) on argument span extraction of all relations; reported as precision (**P**), recall (**R**) and F-measure (**F1**). For the separate model parsers, results include error propagation from argument position classification.

	Arg2	Arg1
Lin et al. (2012)	82.23	59.15
Xu et al. (2012)	81.00	**60.69**
hSMP	80.04	54.37
SMP	**82.35**	57.26

Table 8: Comparison of the separate model parsers (with heuristics (*hSMP*) and CRFs (*SMP*)) to (Lin et al., 2012) and (Xu et al., 2012) reported as F-measure (**F1**). Trained on PDTB sections 02-21, tested on 23.

6.4 Comparison of Separate Model Parser to (Lin et al., 2012) and (Xu et al., 2012)

The separate model parser allows to compare argument span extraction cast as token-level sequence labeling to the syntactic tree-node level classification approach of (Lin et al., 2012) and constituent-level classification approach of (Xu et al., 2012); since now the complexity and the hypothesis spaces are equal. For this purpose we train models on sections 02-21 and test on 23.

Unfortunately, the authors do not report the results on SS and PS cases separately, but only the combined results that include the heuristic. Moreover, the performance of the heuristic is mentioned to be 76.9% instead of 44.2% for the exact match (see IPS x SingFull cell in Table 1 or Table 1 in (Prasad et al., 2008)). Thus, the comparison provided here is not definite. Since all systems have different components up the pipe-line, the only possible comparison is without error propagation.

From the results in Table 8, we can observe that all the systems perform well on *Arg2*. As expected, for the harder case of *Arg1*, performances are lower.

7 Conclusion

In this paper we compare two strategies for the argument span extraction: to process intra- and inter-sentential explicit relations by a single model, or separate ones. We extend the approach of (Ghosh et al., 2011) to argument span extraction cast as token-level sequence labeling using CRFs and integrate argument position classification and immediately previous sentence heuristic. The evaluation of parsing strategies on the PDTB explicit discourse relations shows that the models trained specifically for intra- and inter-sentential relations significantly outperform the single ± 2 window models.

References

Sabine Buchholz. 2000. Readme for perl script chunklink.pl.

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*.

Robert Elwell and Jason Baldridge. 2008. Discourse connective argument identification with connective specific rankers. In *Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2008)*.

Sucheta Ghosh, Richard Johansson, Giuseppe Riccardi, and Sara Tonelli. 2011. Shallow discourse parsing with conditional random fields. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*.

Sucheta Ghosh, Richard Johansson, Giuseppe Riccardi, and Sara Tonelli. 2012a. Improving the recall of a discourse parser by constraint-based postprocessing. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*.

Sucheta Ghosh, Giuseppe Riccardi, and Richard Johansson. 2012b. Global features for shallow discourse parsing. In *Proceedings of the SIGDIAL 2012 Conference, The 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

Dan Klein and Christopher D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15 (NIPS 2002)*.

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

John Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of 18th International Conference on Machine Learning*.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2012. A pdtb-styled end-to-end discourse parser. *Natural Language Engineering*, 1:1 – 35.

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 ACL-IJCNLP Conference*.

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 Language Resources and Evaluation (LREC 2008)*.

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

Robert E. Schapire and Yoram Singer. 2000. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

Maite Taboada and William C. Mann. 2006. Applications of rhetorical structure theory. *Discourse Studies*, (8):567–88.

Bonnie L. Webber, Markus Egg, and Valia Kordoni. 2011. Discourse structure and language technology. *Natural Language Engineering*, pages 1 – 54.

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 2007)*.

Fan Xu, Qiao Ming Zhu, and Guo Dong Zhou. 2012. A unified framework for discourse argument identification via shallow semantic parsing. In *Proceedings of 24th International Conference on Computational Linguistics (COLING 2012): Posters*.

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