Inferring Discourse Relations from PDTB-style Discourse Labels for Argumentative Revision Classification

Fan Zhang **Diane Litman** **Katherine Forbes Riley**
University of Pittsburgh
Pittsburgh, PA, 15260
{zhangfan,litman}@cs.pitt.edu, katherineforbesriley@gmail.com

Abstract

Penn Discourse Treebank (PDTB)-style annotation focuses on labeling local discourse relations between text spans and typically ignores larger discourse contexts. In this paper we propose two approaches to infer discourse relations in a paragraph-level context from annotated PDTB labels. We investigate the utility of inferring such discourse information using the task of revision classification. Experimental results demonstrate that the inferred information can significantly improve classification performance compared to baselines, not only when PDTB annotation comes from humans but also from automatic parsers.

1 Introduction

Widely used in discourse research, Penn Discourse Treebank (PDTB)-style annotation (Prasad et al., 2008) adopts a lexically grounded approach by anchoring discourse relations according to discourse connectives. In a typical PDTB annotation process, an annotator first locates discourse connectives (explicit or implicit) then annotates text spans as their arguments. While the process of manual PDTB annotation has been demonstrated to yield reliable results (Alsaif and Markert, 2011; Danlos et al., 2012; Zhou and Xue, 2015; Zeyrek et al., 2013), it yields more shallow annotation when compared to another widely-used discourse scheme, namely Rhetorical Structure Theory (RST) (Mann and Thompson, 1988; Carlson et al., 2002). This is because when using RST a text is represented as a hierarchical discourse tree, while when using PDTB the relations exist only locally (typically between sentences or clauses).

The lack of discourse information across larger contexts potentially limits the utility of PDTB-style labels. Feng et al. (2014) found that when applied to the tasks of sentence ordering and essay scoring, an RST-style discourse parser outperformed a PDTB-style parser. Performance on both tasks was also likely impacted by parsing errors. To address both the local nature of PDTB-style annotations as well as the errors introduced by state-of-the-art discourse parsers, we propose to first build paragraph-level discourse structures from annotated PDTB labels, then to infer discourse relations based on these structures. We hypothesize that features extracted from inferred relations will improve performance in downstream applications, compared to features extracted from only original annotations.

To verify our hypotheses, we choose the task of argumentative revision classification (Zhang and Litman, 2015; Zhang and Litman, 2016), which aims to identify the purpose of an author's revisions during argumentative writing. This task first detects the differences between two drafts of a paper at the sentence level, then labels each revision using one of five categories: *Claim/Ideas*, *Warrant/Reasoning/Backing*, *Evidence*, *General Content* and *Surface*. While Zhang and Litman (2016) demonstrated decent classification performance without using discourse structure, an error analysis of their results suggests that discourse relations might improve performance (e.g., their current system has trouble differentiating between changing reasoning (*Warrant/Reasoning/Backing*) versus smoothing transitions (*General Content*)). One of the corpora used in their work has recently been annotated with PDTB-style relations, which allows us to explore the utilization of both manually and automatically-produced discourse relations for revision classification. Revision classification also allows us to utilize not only the discourse structure of each version of a paper, but also the differences in discourse structure across versions.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2615–2624, Osaka, Japan, December 11-17 2016.

Draft2 Essay	(1) The lustful are those who long and crave for one another. (2) The person guilty of lust is put in this layer of hell because of his over indulgence of sexual-pleasure. (3) The man who is stuck in this layer is Hue Heffner. (4) He has devoted his entire life for other people 's lustful pleasure and his own. (5) He has spent millions on working on his mansion which is for the purpose of other lustful desires. (6) People who were stuck in this layer are constantly whipped around and " banging " into one another. (7) What you do in your Earthly presence follows with you into Hell. (8) For him and like many others he is now tortured in a whirlwind of torment with others lustful accommodators with himself.
Annotated PDTB	(1->2, EntRel), (2->3, Expansion), (3->4, Contingency), (4->5, Expansion), (5->6, EntRel), (6->7, Contingency), (7->8, Contingency)

Table 1: A paragraph from an essay about putting contemporaries into different levels of hell (top), and annotated PDTB relations between sentences (bottom). The paragraph can be divided into two segments (Section 3.2). In the first segment (sentences (1) to (3)) the author introduces the person to be put in the lustful layer. In the second segment (sentences (4) to (8)), the author states why this person belongs there and how he will be treated. PDTB relations are processed from PDTB annotations ignoring the discourse connectives, e.g. (1->2, EntRel) represents the discourse information: (Arg1: Sentence1, Arg2: Sentence2, Relation Type: EntRel).

2 Related Work

Previous applications of PDTB-style annotations have typically been based on the extraction of PDTB relation occurrence patterns. Lin et al. (2011) encoded PDTB information into the entity-grid (Barzilay and Lapata, 2008) model for textual coherence evaluation. Mithun and Kosseim (2013) utilized PDTB relations for blog summarization. A limitation of the pattern approach is that since it targets a whole paragraph/essay, it is not straightforward to use for prediction tasks on individual sentences. In our work we propose to infer contextual PDTB discourse relations for each sentence, thus enabling the utilization of less-local PDTB information during single sentence prediction.

The construction of our discourse structure looks similar to the building of an RST tree (Duverle and Prendinger, 2009), and there are also prior efforts in combining the benefits of RST and PDTB (e.g., when building a Chinese discourse corpus (Li et al., 2014)). However, the focus of our work is different. The construction of an RST tree aims at creating a discourse representation of the whole sentence/paragraph/essay. We focus on grouping semantically similar text and assigning higher priority to specific local discourse relations. We expect our structure to be able to select the relations that should be propagated to improve performance in downstream applications.

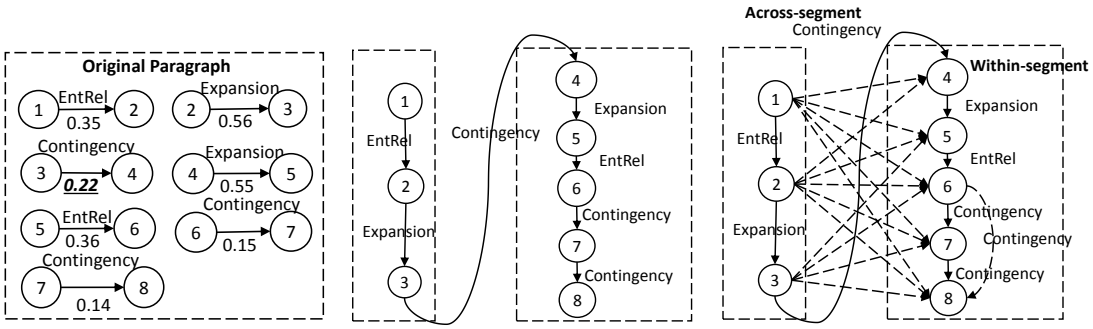
3 Inferring Discourse Information from PDTB-style Labels

3.1 Intuitions for PDTB relation inference

Different from other discourse annotations, the PDTB annotation schema anchors at the labeling of discourse connectives and labels text spans around the connective. The annotator either locates the "Explicit" connectives or manually fills in the "Implicit" connectives between two text spans. The text span where the connective structurally attaches to is called **Arg2**, while the other text span is called **Arg1**. The spans are usually used at the level of sentence/phrase. In (Prasad et al., 2014), five relation types are annotated: *Explicit*, *Implicit*, *AltLex*, *EntRel* and *NoRel*. Within the *Explicit*/*Implicit* relations, the senses of relations are further categorized at multiple levels. In Level-1, the relations are categorized to 4 senses: *Comparison*, *Contingency*, *Expansion* and *Temporal*. In this paper we focus on the type/sense of Level-1 relations only and ignore the discourse connectives[1]. *Arg1*, *Arg2* and the discourse relation type/sense are used as demonstrated in Table 1. For the *Explicit*/*Implicit* relations, we use the sense of the relation directly to represent the relation. Below we explain our intuitions for inferring new discourse relations within the paragraph. Sections 3.2 and 3.3 then detail our corresponding computational approaches.

Intuition 1. Latent discourse relations can be inferred from annotated discourse relations. In this paper we explore two possible cases: **1) Same type transition**: If sentence A has relation type T with sentence

[1]We plan to explore connectives in future work.



(a) Original PDTB annotation (b) After linear segmentation (c) After relation propagation

Figure 1: The construction of PDTBSegment structure of the example in Table 1. As sentence similarity between 3 and 4 is 0.22, smaller than the value 0.56 (before) and 0.55 (after), the paragraph is segmented to segment(1-3) and segment (4-8). Afterwards relations are inferred both within the segment and across the segments. The dashed lines represent the propagated relations.

B and sentence B has the same relation type with sentence C, we can infer that A has relation type T with C. In the example in Table 1, a *Contingency* relation between sentences 6 and 8 will be inferred from the *Contingency* relationships between sentences (6,7) and sentences (7,8). **2) Across segment propagation**: If a paragraph can be segmented to text segments semantically dissimilar to each other (i.e. the two text segments are serving different semantic purposes), the discourse relation of sentences on the boundary of two segments can be propagated to infer weaker relations between all sentences in the segments. In the example in Table 1, due to the discourse relation between sentences 3 and 4 and the segment boundary between them, the segment from 4 to 8 will also be viewed as a reasoning (*Contingency*) of the segment from 1 to 3 (why and how Hue Heffner belongs to the lustful level), and weak relations are inferred between sentences (1,2,3) and (4,5,6,7,8).

Intuition 2. The importance of discourse relations to argumentation varies even if the relation types are the same. The relations connecting the semantically dissimilar segments are likely to be more important than the relations within a segment. In Table 1, the *Contingency* relation between sentences 3 and 4 transits the thesis introduction to the arguments supporting the thesis. The *Contingency* relation between sentences 6 and 7 is just a transition to smooth the description of how Hue Heffner is going to be treated.

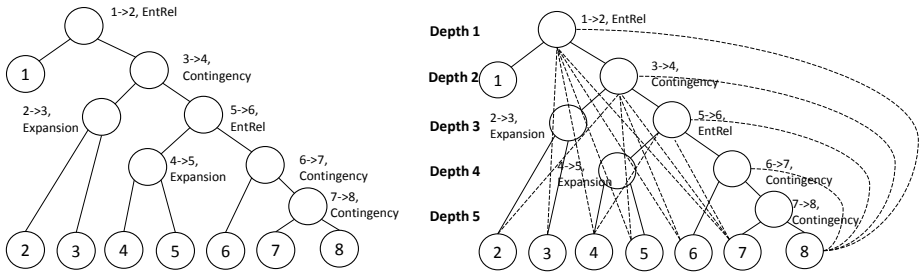
3.2 PDTBSegment

Based on intuition 1, the *PDTBSegment* approach emphasizes the inference of discourse relations.

Step1. Linear segmentation. Inspired by the TextTiling algorithm (Hearst, 1997) for linear segmentation, we utilize the "valley" of semantic similarity scores between sentences as the segmentation boundary. The summed word-embedding vector is calculated for each sentence[2] and cosine value between vectors is used as the similarity score. Similarity scores indicates a possible segmentation boundary. In the example of Figure 1(a), the similarity between (2,3) and the similarity between (4,5) are larger than the similarity between (3,4), in other words, sentence 3 and 4 has a low similarity score preceded by and followed by high similarity scores, thus the paragraph is first segmented into segment (1,2,3) and segment (4,5,6,7,8) as in Figure 1(b).

Step2. Relation inference. **1) Within segment.** We conduct "same type transition" for sentences within the same segment. As in Figure 1(c), there exists relation *Contingency* between 6 and 8 as the same relationship exists between 6, 7 and 7, 8. **2) Across segment.** "Across-segment propagation" is conducted for sentences in different segments. If there exists relation (type T) between two segments Seg1 and Seg2, a relation with type T is inferred for each sentence in Seg1 and each sentence in Seg2. In Figure 1(c), we propagate the *Contingency* relations between sentence (1,2,3) and sentences (4,5,6,7,8).

[2]Pre-trained word2vec vectors from (Mikolov et al., 2013).



(a) After tree structure construction (b) After relation propagation

Figure 2: PDTBTree structure of Table 1 example. The dashed lines represent the propagated relations.

Segment	1(Arg2)	2	3	4	5	6	7	8
1(Arg1)	N/A[a]	Ent	N/A[b]	Cont(2,0)[c]	Cont(2,1)	Cont(2,2)	Cont(2,3)	Cont(2,4)
2	N/A	N/A	Expan	Cont(1,0)	Cont(1,1)	Cont(1,2)	Cont(1,3)	Cont(1,4)
3	N/A	N/A	N/A	Cont	Cont(0,1)	Cont(0,2)	Cont(0,3)	Cont(0,4)
4	N/A	N/A	N/A	N/A	Expan	N/A	N/A	N/A
5	N/A	N/A	N/A	N/A	N/A	EntRel	N/A	N/A
6	N/A	N/A	N/A	N/A	N/A	N/A	Cont	Cont(1)
7	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Cont
8	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Tree	1(Arg2)	2	3	4	5	6	7	8
1(Arg1)	N/A	Ent-1	Ent-1(0,1)[d]	Ent-1(0,1)	Ent-1(0,2)	Ent-1(0,2)	Ent-1 (0,3)	Ent-1(0,3)
2	N/A	N/A	Expan-3	Cont-2(1,0)	Cont-2(1,1)	Cont-2(1,1)	Cont-2(1,2)	Cont-2(1,3)
3	N/A	N/A	N/A	Cont-2	Cont-2(0,1)	Cont-2(0,1)	Cont-2(0,2)	Cont-2(0,3)
4	N/A	N/A	N/A	N/A	Expan-4	Ent-3(1,0)	Ent-3(1,1)	Ent-3(1,2)
5	N/A	N/A	N/A	N/A	N/A	Ent-3	Ent-3(0,1)	Ent-3(0,2)
6	N/A	N/A	N/A	N/A	N/A	N/A	Cont-4	Cont-4(0,1)
7	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Cont-5(0,1)
8	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 2: Relation matrix constructed for the PDTBSegment approach (Upper) and the PDTBTree approach (Below). Ent is short for EntRel, Expan short for Expansion and Cont short for Contingency.

[a]Relationship between clauses within the sentence is not used in relation inference.

[b]No relations can be inferred between 1 and 3.

[c]Cont(2,0) means distance to real Arg1 is 2 and distance to real Arg2 is 0. Here the inferred relation is coming from the labeled relation (3->4, Contingency). 3 is the real Arg1 and 4 is the real Arg2. Distance to real Arg1 is 2 as the the distance between 1 and 3 is 2.

[d]Ent-1(0,1) stands for Depth 1 distance, distance=0 for Arg1 and distance = 1 for Arg2.

3.3 PDTBTree

PDTBTree focuses on intuition 2 using sentence aggregation. To better separate important discourse relations, a hierarchical tree structure is constructed for each paragraph and relations then inferred.

Step 1. Tree construction. As in Figure 2(a), the tree is constructed iteratively starting with each sentence constructed as a leaf node. Semantic similarities between adjacent sentences are calculated in the same manner as the *PDTBSegment* approach. In each round, the two most similar nodes are selected and merged into one node and similarities between the merged node and its adjacent nodes are calculated[3]. The selection and merge of nodes repeats until there is only one root node left.

Discourse relations are assigned to the non-leaf nodes after tree construction. For each tree node, sentences in its left and right child are listed as $Nodes_{left}$ and $Nodes_{right}$. Relations with Arg1 in $Nodes_{left}$ and Arg2 in $Nodes_{right}$ are assigned to the merged node. For example, the discourse relation (1->2, EntRel) is assigned to the root node as sentence 1 is in its left child and sentence 2 is in its right child. After this step we bind each non-leaf node with one or several discourse relations.

Step 2. Relation inference. Relations are first assigned different levels of importance as depths. As in Figure 2(b), the assignment starts at the root node and traverses the whole tree until all the non-leaf nodes

[3]The similarity between merged nodes is calculated as the average of the similarity between their child leaf nodes.

are labeled. Depth starts from 1 and smaller number indicates larger importance. As in the example, we notice that the transition from the thesis to its reasoning (3->4) is recognized as a depth-2 relation while the transitions between sentences 6,7,8 are recognized as depth-4 and depth-5 relations.

Afterwards discourse relations are inferred by traversing up from the leaf nodes back to their parent nodes. The parent node is used as the discourse connector and its child leaf nodes are used as Arg1 and Arg2. For example, in Figure 2(b), sentence 3 is the left child of the node (3->4, Contingency) and sentence 5 is the right child. Thus we infer the discourse relation between 3 and 5 as (3->5, Contingency).

3.4 Constructing the relation matrix

For both approaches, relation matrices are constructed to represent the discourse information as in Table 2. Extraction of features using the matrix is described in the next section. Relations already labeled by the annotator/parser are directly recorded in the matrix. Observing that the reliability of an inferred relation decreases as the number of annotated relations connecting the arguments increases, we record not only the relation types but also the **"distance"** information for the inferred relations.

For the **PDTBSegment** approach, distances are recorded separately for within-segment relations and across-segment relations. For within-segment relations, the distance is recorded according to the number of sentences between Arg1 to Arg2. For example, distance for sentence 6 and 8 is recorded as 2 as there is one sentence between the two sentences. For across-segment relations, distances are recorded for both Arg1 and Arg2 according to their distances to the real Arg1/Arg2 of the across-segment relation as (Dist1, Dist2). For example, distance between sentence 1 and 5 is recorded as (2,1) as there is the distance of 2 between sentence 1 and 3 and there is the distance of 1 between sentence 4 and 5.

For the **PDTBTree** approach, we traverse up from Arg1 and Arg2 to their closest common parent node and count the distances for both arguments as (Dist1, Dist2). In Table 2, distance between sentence 2 and 5 is recorded as (1,1) as we back trace both nodes to their parent node (3->4). As sentence 2 is the real text span in relation node (2->3) and sentence 5 is in the node (5->6), we get distance 1 for sentence 2 as the distances between (2->3) and (3->4) in the tree is 1; similarly, we get distance 1 for sentence 5.

4 Task and Data Description

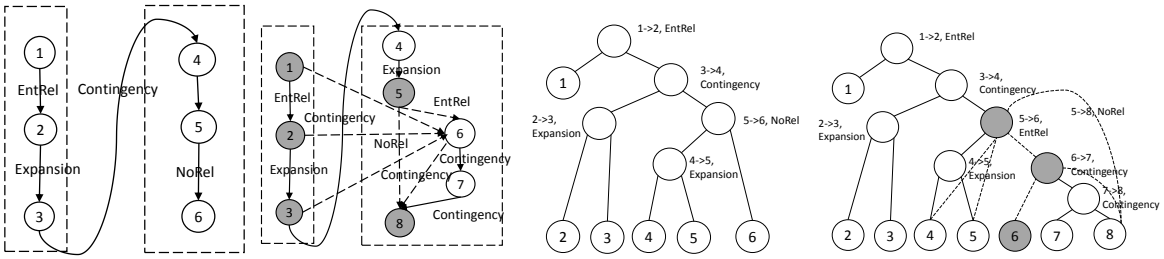
Argumentative revision classification. The task of revision classification aims to detect then categorize an author's changes to their writing. Revision research has been conducted on Wikipedia articles (Bronner and Monz, 2012; Daxenberger and Gurevych, 2013) and argument-oriented study essays (Zhang and Litman, 2015). The example in Table 1 contains a paragraph from a revised student essay. Comparing to its previous draft, the changes are the addition of sentence 6 and sentence 7. The addition of sentence 6 is labeled as (Null->6, "Add", "General Content"). The full classification process involves the alignment of sentences/clauses to locate changes (recognizing the alignment Null->6 and the revision operation "Add") and the classification of change types (identifying the revision type "General Content"). In this work we assume perfect alignment and focus on improving classification performance (the recognition of "General Content") by using inferred PDTB information.

Annotated revision dataset. To evaluate revision classification performance, we use the two corpora used in (Zhang and Litman, 2016)[4]. Each student wrote two essays: *Draft 1* where the students initially write the essay, and *Draft 2* where the students revise Draft 1 after receiving comments from other students. Corpus A contains 47 students (94 essays) and 1267 revised sentence pairs, talking about placing contemporaries into Dante's Inferno. Corpus B contains 63 students (126 essays) and 1044 revised sentence pairs, where students explain the rhetorical strategies used by the speaker/author of a previously read lecture/essay. Distribution of revisions is shown in the first columns of Tables 4 and 5.

PDTB annotation. Recently PDTB discourse information was annotated on corpus A by one of the early developers of the D-LTAG environment (which engendered the PDTB framework)[5] (Forbes et al., 2003; Webber, 2004; Forbes-Riley et al., 2016). Five relation types were annotated: *Explicit*, *Implicit*, *EntRel*, *AltLex* and *NoRel*. Within the *Explicit* and *Implicit* types, four level-1 senses were labeled:

[4]Both corpora are from high school AP English classes.

[5]Thus considered as an expert annotator.



(a) Segment for Draft 1 (b) Segment for Draft 2 (c) Tree for Draft 1 (d) Tree for Draft 2

Figure 3: The change of discourse structure from Draft 1 (D1) to Draft 2 (D2). The gray nodes are the affected nodes and the dashed lines are the affected relations. Sentences are aligned as (1->1), (2->2), (3->3), (4->4), (5->5), (6->8), (Null->6), (Null->7).

Features	Example
Loc	D1-Arg1[a]: N/A, D1-Arg2: N/A, D2-Arg1: Contingency, D2-Arg2: EntRel
Seg	**Individual**: WithinSegment: D1-Arg1: N/A, D1-Arg2: N/A, D2-Arg1: Contingency, D2-Arg2: EntRel, AcrossSegment: D2-Arg2: (Contingency, $\frac{1}{3}$)[b] **Structure**: WithinSegment[c]: (-1, 1, 0, 0, 0.5, 0, 0), AcrossSegment: (0, 0, 0, 0, $\frac{1}{3}$, 0, 0)
Tree	**Individual**: D1-Arg1: N/A, D1-Arg2: N/A, D2-Arg1: Contingency-4, D2-Arg2: (EntRel-1,$\frac{1}{8}$), (Contingency-2, $\frac{1}{4}$), EntRel-3 **Structure**: Depth1Vector: (0,0,0,0,0,0), Depth2Vector:(0,0,0,0,0,0), Depth3Vector: (-1,1,0,0,0,0), Depth4Vector: (0,0, 0,0,1,0,0)

Table 3: Examples of the features extracted for the added sentence 6 in Table 1.

[a]D1-Arg1 means features of sentence acting as Arg1 in the first Draft.
[b](Contingency, $\frac{1}{3}$) represents relation type Contingency with weight $\frac{1}{3}$.
[c]The columns of the change vector are (NoRel, EntRel, AltLex, Comparison, Contingency, Expansion, Temporal).

Comparison, *Contingency*, *Expansion* and *Temporal*. Similarly to (Prasad et al., 2011), the annotator relaxed the structural adjacency constraint, allowing the annotation of relations between non-adjacent sentences. Annotated and inferred PDTB information will be used to construct features for revision classification as described next.

5 Using Inferred PDTB for Classification

In this section we investigate whether using the annotated PDTB information itself can improve the classification performance, and whether such performance can be improved with the inferred PDTB information. For both of our inference approaches, we extract features individually to utilize the inferred relation type and further extract structure change features to utilize the PDTB structures built.

5.1 Features

The *PDTBSegment* and *PDTBTree* structures are constructed for both drafts as in Figure 3. Table 3 shows the PDTB features extracted for the added sentence 6 in Table 1, with features explained below.

Baseline features (Base). Features in (Zhang and Litman, 2015) are used as a baseline. Besides *Unigram* features, *Location* features encode the location of the revised sentence. *Textual* features encode revision operation, sentence length, edit distance between revised sentences and punctuation. *Language* features encode part of speech (POS) unigrams and difference in POS tag counts.

Features using the labeled local PDTB information (Local). Features are extracted as the types of relations a sentence is involved with (i.e. the relation where the sentence acts as Arg1 or Arg2.) Features are extracted for sentences in both drafts. If a sentence is added or deleted, the features for the empty sentence are marked as N/A.

Features using PDTBSegment (Segment).

Individual features Within each draft, the features of sentences are extracted based on the relation ma-

trix. Similar to **Local**, we extract the discourse relation type of each sentence acting as Arg1 and Arg2[6]. Features for across-segment relations are extracted separately since the discourse relations across segments are likely to be more important than relations within segments. Weights are assigned to relations according to their distance information. A within-segment relation with distance (d1) is assigned weight $\frac{1}{d1+1}$; an across-segment relation with distance (d1, d2) is assigned weight $\frac{1}{(d1+1)*(d2+1)}$. If a sentence is involved with multiple same-type relations, the relation with the largest weight is chosen.

Structure change features For these across draft features, the segment structures created for draft 1 and draft2 are compared. Nodes of segment structures are aligned according to the sentence alignment information. After comparison, the aligned nodes that are affected by the revision are selected, where the change of their related relations with the revised sentence are counted. For example in Figure 3(b), sentences 1, 2, 3, 5, 8 are affected by the addition of sentence 6. For sentence 1, 2, 3, sentence 6 brings addition of three across-segment relations. For sentence 5, the original "NoRel" label between sentence 5 and sentence 8 is removed. For sentence 8, relation between sentence 6 and sentence 8 is added. A vector of relation changes is thus created according to the relation matrix.

Features using PDTBTree (Tree).

Individual features Features are collected in a similar manner as the **PDTBSegment** approach. To enlarge the difference of different-depth relations, weight $\frac{1}{2^{d1+d2}}$ is assigned to a relation with distance (d1, d2) .

Structure change features Due to the complexity of the tree structure, only the non-leaf nodes that are directly related to the revised sentence (i.e. the sentence as Arg1 or Arg2 of the relation) are considered in the extraction of structure changes. As in Figure 3(d), the added sentence 6 acts as Arg2 in node (5->6) and Arg1 in node (6->7). Change of relations (4->6), (5->6) are considered as the changed relations of node (5->6). Change of relations (5->8) and (6->8) are the changed relations of node (6->7). Change vectors are calculated in similar manners as the **PDTBSegment** approach at each depth. To avoid data sparsity, the depth number is limited to 4 to reduce the number of features[7].

It is important to notice that as in the standard PDTB annotations, the spans of arguments may cover only a part of a sentence or multiple sentences. If the span covers only a part of the sentence and the two spans of the relation come from two different sentences, we use the relationship as the relation between the two sentences. If the two spans come from one sentence, we consider it as a self-relation. Such information is used in the extraction of local PDTB features but not used in relation inference. If a span of a relation covers multiple sentences, we infer relations as in the PDTBSegment approach (consider the multiple sentences span as a text segment). Also, it is possible to have more than one relation annotation between two sentences. In that case all the relations are kept and used in relation inference.

5.2 Experiments and Results

We repeated the experiment in (Zhang and Litman, 2016) using our new proposed feature group[8]. 5-class revision category classification[9] was conducted with the SVM[10] classifier. Two hypotheses are proposed: 1) For manually labeled PDTB relations, using features extracted from inferred relations has better performance than using baseline features or baseline with only local PDTB features. 2) For automatically labeled PDTB relations, using features extracted from the inferred relations reduces the noise introduced by the PDTB parser and has better performance than using only local PDTB features.

Based on the hypotheses, two experiments were conducted. In both experiments we compared the results using inferred information to the baseline results, and to the results with baseline features plus each individual feature group[11]. In Experiment 1, we experimented with the PDTB features extracted from manual labels on Corpus A (where we have manual annotations). In Experiment 2, we experimented

[6]The row of the sentence in the relation matrix corresponds to Arg1 and the column corresponds to Arg2.

[7]If the depth of tree is larger than 4, the depth of the relation is still considered as 4.

[8]In this paper we focus on the comparison of features and thus do not directly compare our approach with the sequence labeling approach used in their work

[9]*Claim*, *Warrant*, *Evidence*, *General* and *Surface*

[10]SVM model implemented with Weka (Hall et al., 2009).

[11]We also experimented mixing all the features groups together but did not observe significant improvement.

Corpus A	Base	Base+Local	Base+Segment	Base+Tree
Claim(111)	0.540	0.530	0.500	**0.578**‡*
Warrant(390)	0.680	0.693	**0.715**‡*	0.713‡*
Evidence(110)	0.288	0.347*	0.387‡*	**0.415**‡*
General(356)	0.694	0.715	**0.746**‡*	0.724*
Surface(300)	0.868	**0.872**	0.869	0.870
Average(1267)	0.614	0.630	0.642*	**0.658**‡*

Table 4: Experiment 1. With manually labeled PDTB. The average F-measure of 10-fold (student) cross-validation is reported, average represents the unweighted average F1 of all 5 categories. $*$ indicates significantly better than the baseline (paired T-test, $p<0.05$), ‡ indicates significantly better than (Base+local), **bold** indicates best.

Corpus A	Base	B+Local	B+Segment	B+Tree	B-	(B-)+Local	(B-)+Segment	(B-)+Tree
Claim(111)	**0.540**	0.517	0.516	0.518	0.466	0.475⋆	0.501⋆	0.504⋆
Warrant(390)	0.680	0.669	**0.698**‡	0.682	0.658	0.647	0.686⋆	0.671⋆
Evidence(110)	0.288	0.299	0.276	**0.306**	0.274	0.266	0.261	0.282
General(356)	0.694	0.683	**0.702**‡	0.683	0.621	0.643⋆	0.696⋆	0.682⋆
Surface(300)	**0.868**	0.865	**0.868**	0.863	0.841	0.835	0.843	0.844
Average(1267)	0.614	0.605	**0.617**‡	0.609	0.572	0.573	0.597⋆	0.596⋆
Corpus B	Base	B+Local	B+Segment	B+Tree	B-	(B-)+Local	(B-)+Segment	(B-)+Tree
Claim(76)	0.504‡	0.471	0.496‡	**0.512**‡	0.421	0.433	0.443⋆	0.451⋆
Warrant(327)	0.611	0.620	**0.635***	0.609	0.588	0.591	0.610⋆	0.605⋆
Evidence(34)	0.024	0.088	**0.094**	0.044	0.024	0.088	0.088	0.088
General(216)	**0.505**‡	0.459	0.503‡	0.484‡	0.451	0.455	0.477⋆	0.469⋆
Surface(391)	0.867	0.853	**0.872**‡	0.865	0.848	0.851	0.855	0.853
Average(1044)	0.503	0.495	**0.520**‡	0.503	0.466	0.483	0.495⋆	0.493⋆

Table 5: Experiment 2. With automatically labeled PDTB. B short for Base, B- is a weaker baseline using unigram and *Textual* features from the baseline approach. ⋆ indicates significantly better than B-. ‡ indicates significantly better than (B+local), **bold** indicates best.

with the PDTB features extracted from labels generated with Lin's automatic PDTB parser (Lin et al., 2014) on Corpora A[12] and B. All experiments were conducted using 10-fold (student) cross-validation with 300 features selected[13] using learning gain ratio. Tables 4 and 5 demonstrate the results.

Experiment 1 results provide support for our first hypothesis. Comparing to the baseline, Base+Local (using only features from the labeled PDTB relations) yields a significant improvement only when classifying *Evidence* revisions and a non-significant overall average improvement. In contrast, both Base+Segment and Base+Tree (our inference-based approaches) yield several significant improvements over the baseline[14]. Comparing to the baseline, the **PDTBSegment** approach yields significant improvement in the classification of *Warrant*, *Evidence* and *General Content* revisions and the **PDTBTree** approach yields significant improvement in the classification of all revisions except *Surface*. For the minority category *Evidence*, the **PDTBTree** approach improved F1 from 0.288 to 0.415. Comparing to the results using only labeled PDTB, the **PDTBSegment** approach yields significant improvement in the classification of *Warrant*, *Evidence* and *General*, while the **PDTBTree** approach yields significant improvement in the classification of *Claim*,*Warrant* and *Evidence* and a significant overall F1 improvement.

Experiment 2 results support our second hypothesis. On corpus A, we observe a significant performance drop ($p<0.05$) in average F1 score for all the (B+) feature groups comparing to the Experiment 1 results in Table 4, which indicates that the noisy output generated by the parser impacts the performance[15]. While we do not gain significant improvement over the baseline using the inferred relations, we still observe significantly better performance using the **PDTBSegment** approach comparing to using

[12]The same fold is used as Experiment 1.

[13]Features selected only on the training folds each round.

[14]We also tested using just individual features (without the structure change features) and both approaches still significantly outperform the baseline.

[15]We compared the output of the PDTB parser and the manual annotation on Corpus A, indicating that the F-measure for the relation prediction is 0.45.

only the automatically labeled relations. Meanwhile, we observe that the inferred PDTB features can still significantly improve the performance of a weaker baseline with *Unigram* and *Textual* features while the Local features can't, suggesting that the our approach is still effective even with the noisy parser output.

6 Conclusion and Future Works

This paper presented two approaches to construct a paragraph-level discourse structure from local PDTB discourse labels, and infer discourse relations within the structure. We applied our approaches on the task of argumentative revision classification with manually/automatically labeled PDTB information. Results demonstrated that using features extracted from inferred PDTB relations yields better revision classification performance than using features from original PDTB annotations. Results also showed that our method reduced the impact of the automatic PDTB parsing errors.

We believe our work can be expanded from two perspectives. 1) From the PDTB perspective, we can investigate our approach on other traditional PDTB applications such as coherence evaluation to see if we can still observe an improvement on these tasks. 2) From the argumentative revision classification perspective, we plan to explore what aspects of our proposed approach yields most robustness to errors from automatic PDTB parsers. We also want to try other discourse parsers in recent shallow discourse parsing shared tasks (CoNLL, 2016). We also plan to investigate whether the RST-style discourse information can also improve the classification performance and compare the results with our approach. We also plan to investigate whether our approaches can be applied to other type of writings besides argumentative writings.

Acknowledgments

We want to thank the members of the SWoRD and ITSPOKE groups for their helpful feedback and all the anonymous reviewers for their suggestions.

This research is funded by NSF Award #1550635 and the Learning Research and Development Center of the University of Pittsburgh.

References

Amal Alsaif and Katja Markert. 2011. Modelling discourse relations for Arabic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 736–747. Association for Computational Linguistics.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Amit Bronner and Christof Monz. 2012. User edits classification using document revision histories. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 356–366. Association for Computational Linguistics.

Lynn Carlson, Mary Ellen Okurowski, Daniel Marcu, Linguistic Data Consortium, et al. 2002. *RST discourse treebank*. Linguistic Data Consortium, University of Pennsylvania.

CoNLL. 2016. Conll 2016 shared task. Accessed: 2016-07-15.

Laurence Danlos, Diégo Antolinos-Basso, Chloé Braud, and Charlotte Roze. 2012. Vers le FDTB: French Discourse Tree bank. In *Proceedings of TALN 2012: 19ème conférence sur le Traitement Automatique des Langues Naturelles*, volume 2, pages 471–478. ATALA/AFCP.

Johannes Daxenberger and Iryna Gurevych. 2013. Automatically classifying edit categories in Wikipedia revisions. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 578–589, Seattle, Washington, USA, October. Association for Computational Linguistics.

David A Duverle and Helmut Prendinger. 2009. A novel discourse parser based on support vector machine classification. 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: Volume 2-Volume 2*, pages 665–673. Association for Computational Linguistics.

Vanessa Wei Feng, Ziheng Lin, Graeme Hirst, and Singapore Press Holdings. 2014. The impact of deep hierarchical discourse structures in the evaluation of text coherence. In *Proceedings of International Conference on Computer Linguistics*, pages 940–949.

Katherine Forbes, Eleni Miltsakaki, Rashmi Prasad, Anoop Sarkar, Aravind Joshi, and Bonnie Webber. 2003. D-LTAG system: Discourse parsing with a lexicalized tree-adjoining grammar. *Journal of Logic, Language and Information*, 12(3):261–279.

Kate Forbes-Riley, Fan Zhang, and Diane Litman. 2016. Extracting PDTB discourse relations from student essays. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 117–127, Los Angeles, September. Association for Computational Linguistics.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Marti A Hearst. 1997. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, 23(1):33–64.

Yancui Li, Wenhe Feng, Jing Sun, Fang Kong, and Guodong Zhou. 2014. Building Chinese Discourse Corpus with connective-driven dependency tree structure. In *Proceedings of Empirical Methods of Natural Language Processing*, pages 2105–2114.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2011. Automatically evaluating text coherence using discourse relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 997–1006. Association for Computational Linguistics.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, 20(02):151–184.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in neural information processing systems*, pages 3111–3119.

Shamima Mithun and Leila Kosseim. 2013. Measuring the effect of discourse relations on blog summarization. In *Proceedings of International Joint Conference of Natural Language Processing*, pages 1401–1409.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The Penn Discourse Treebank 2.0. In *Proceedings of Language Resources and Evaluation Conference*.

Rashmi Prasad, Susan McRoy, Nadya Frid, Aravind Joshi, and Hong Yu. 2011. The biomedical discourse relation bank. *BMC bioinformatics*, 12(1):1.

Rashmi Prasad, Bonnie Webber, and Aravind Joshi. 2014. Reflections on the Penn Discourse Treebank, comparable corpora, and complementary annotation. *Computational Linguistics*.

Bonnie Webber. 2004. D-LTAG: extending lexicalized tag to discourse. *Cognitive Science*, 28(5):751–779.

Deniz Zeyrek, Işın Demirşahin, AB Sevdik-Çallı, and Ruket Çakıcı. 2013. Turkish Discourse Bank: Porting a discourse annotation style to a morphologically rich language. *Dialogue and Discourse*, 4(2):174–184.

Fan Zhang and Diane Litman. 2015. Annotation and classification of argumentative writing revisions. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 133–143, Denver, Colorado, June. Association for Computational Linguistics.

Fan Zhang and Diane Litman. 2016. Annotation and classification of argumentative writing revisions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*, page to appear, San Diego, California, June. Association for Computational Linguistics.

Yuping Zhou and Nianwen Xue. 2015. The Chinese Discourse Treebank: A Chinese corpus annotated with discourse relations. *Language Resources and Evaluation*, 49(2):397–431.