Reducing Sparsity Improves the Recognition of Implicit Discourse Relations

Junyi Jessy Li
University of Pennsylvania
ljunyi@seas.upenn.edu

Ani Nenkova
University of Pennsylvania
nenkova@seas.upenn.edu

Abstract

The earliest work on automatic detection of implicit discourse relations relied on lexical features. More recently, researchers have demonstrated that syntactic features are superior to lexical features for the task. In this paper we re-examine the two classes of state of the art representations: syntactic production rules and word pair features. In particular, we focus on the need to reduce sparsity in instance representation, demonstrating that different representation choices even for the same class of features may exacerbate sparsity issues and reduce performance. We present results that clearly reveal that lexicalization of the syntactic features is necessary for good performance. We introduce a novel, less sparse, syntactic representation which leads to improvement in discourse relation recognition. Finally, we demonstrate that classifiers trained on different representations, especially lexical ones, behave rather differently and thus could likely be combined in future systems.

1 Introduction

Implicit discourse relations hold between adjacent sentences in the same paragraph, and are not signaled by any of the common explicit discourse connectives such as *because*, *however*, *meanwhile*, etc. Consider the two examples below, drawn from the Penn Discourse Treebank (PDTB) (Prasad et al., 2008), of a causal and a contrast relation, respectively. The italic and bold fonts mark the arguments of the relation, i.e the portions of the text connected by the discourse relation.

Ex1: *Mrs Yeargin is lying.* [Implicit = BECAUSE] **They found students in an advanced class a year earlier who said she gave them similar help.**

Ex2: *Back downtown, the execs squeezed in a few meetings at the hotel before boarding the buses again.* [Implicit = BUT] **This time, it was for dinner and dancing - a block away.**

The task is undisputedly hard, partly because it is hard to come up with intuitive feature representations for the problem. Lexical and syntactic features form the basis of the most successful studies on supervised prediction of implicit discourse relations in the PDTB. Lexical features were the focus of the earliest work in discourse recognition, when cross product of words (word pairs) in the two spans connected via a discourse relation was studied. Later, grammatical productions were found to be more effective. Features of other classes such as verbs, inquirer tags, positions were also studied, but they only marginally improve upon syntactic features.

In this study, we compare the most commonly used lexical and syntactic features. We show that representations that minimize sparsity issues are superior to their sparse counterparts, i.e. the better representations are those for which informative features occur in larger portions of the data. Not surprisingly, lexical features are more sparse (occurring in fewer instances in the dataset) than syntactic features; the superiority of syntactic representations may thus be partially explained by this property.

More surprising findings come from a closer examination of instance representation approaches in prior work. We first discuss how choices in prior work have in fact exacerbated the sparsity problem of lexical features. Then, we introduce a new syntactically informed feature class, which is less sparse than prior lexical and syntactic features, and improves significantly the classification of implicit discourse relations.

Given these findings, we address the question if any lexical information at all should be preserved in discourse parsers. We find that purely syntactic representations show lower recognition

Proceedings of the SIGDIAL 2014 Conference, pages 199–207,
Philadelphia, U.S.A., 18-20 June 2014.

for most relations, indicating that lexical features, albeit sparse, are necessary for the task. Lexical features also account for a high percentage of the most predictive features.

We further quantify the agreement of predictions produced from classifiers using different instance representations. We find that our novel syntactic representation is better for implicit discourse relation prediction than prior syntactic feature because it has higher overall accuracy and makes correct predictions for instances for which the alternative representations are also correct. Different representation of lexical features however appear complementary to each other, with markedly higher fraction of instances recognized correctly by only one of the models.

Our work advances the state of the art in implicit discourse recognition by clarifying the extent to which sparsity issues influence predictions, by introducing a strong syntactic representation and by documenting the need for further more complex integration of lexical information.

2 The Penn Discourse Treebank

The Penn Discourse Treebank (PDTB) (Prasad et al., 2008) contains annotations for five types of discourse relations over the Penn Treebank corpus (Marcus et al., 1993). *Explicit* relations are those signaled by a discourse connective that occurs in the text, such as "because", "however", "for example". *Implicit* relations are annotated between adjacent sentences in the same paragraph. There are no discourse connectives between the two sentences, and the annotators were asked to insert a connective while marking their senses. Some pairs of sentences do not contain one of the explicit discourse connectives, but the insertion of a connective provides redundant information into the text. For example, they may contain phrases such as "the consequence of the act". These are marked *Alternative Lexicalizations* (AltLex). *Entity relations* (EntRel) are adjacent sentences that are only related via the same entity or topic. Finally, sentences where no discourse relations were identified were marked *NoRel*. In this work, we consider *AltLex* to be part of the *Implicit* relations, and *EntRel* to be part of *NoRel*.

All connectives, either explicit or implicitly inserted, are associated with two arguments of the minimal span of text conveying the semantic content between which the relation holds. This is illustrated in the following example where the two arguments are marked in bold and italic:

Ex: *They stopped delivering junk mail.* [Implicit = SO] **Now thousands of mailers go straight into the trash.**

Relation senses in the PDTB are drawn from a 3-level hierarchy. The top level relations are *Comparison* (arg1 and arg2 holds a contrast relation), *Contingency* (arg1 and arg2 are causally related), *Expansion* (arg2 further describes arg1) and *Temporal* (arg1 and arg2 are temporally related). Some of the largest second-tier relations are under *Expansion*, which include *Conjunction* (arg2 provides new information to arg1), *Instantiation* (arg2 exemplifies arg1) and *Restatement* (arg2 semantically repeats arg1).

In our experiments we use the four top level relations as well as the above three subclasses of *Expansion*. All of these subclasses occur with frequencies similar to those of the Contingency and Comparison classes, with thousands of examples in the PDTB.[1] We show the distribution of the classes below:

Temporal	1038	Comparison	2550
Contingency	4532	Instantiation	1483
Restatement	3271	Conjunction	3646
EntRel/NoRel	5464		

3 Experimental settings

In our experiments we use only lexical and syntactic features. This choice is motivated by the fact that lexical features have been used most widely for the task and that recent work has demonstrated that syntactic features are the single best type of representation. Adding additional features only minimally improves performance (Lin et al., 2009). By zeroing in only on these classes of features we are able to discuss more clearly the impact that different instance representation have on sparsity and classifier performance.

We use gold-standard parses from the original Penn Treebank for syntax features.

To ensure that our conclusions are based on analysis of the most common relations, we train binary SVM classifiers[2] for the seven relations described above. We adopt the standard practice in

[1] All other sub-classes of implicit relations are too small for general practical applications. For example the *Alternative* class and *Concession* class have only 185 and 228 occurrences, respectively, in the 16,224 implicit relation annotations of the PDTB.

[2] We use SVMLight (Joachims, 1999) with linear kernel.

prior work and downsampled the negative class so the number of positive and negative samples are equal in the training set.[3]

Our training set consists of PDTB sections 2-19. The testing set consists of sections 20-24. Like most studies, we do not include sections 0-1 in the training set. We expanded the test set (sections 23 or 23-24) used in previous work (Lin et al., 2014; Park and Cardie, 2012) to ensure the number of examples of the smaller relations, particularly of *Temporal* or *Instantiation*, are suitable for carrying out reliable tests for statistical significance.

Some of the discourse relations are much larger than others, so we report our results in term of F-measure for each relation and average unweighted accuracy. Significance tests over F scores were carried out using a paired t-test. To do this, the test set is randomly partitioned into ten groups. In each group, the relation distribution was kept as close as possible to the overall test set.

4 Sparsity and pure lexical representations

By far the most common features used for representing implicit discourse relations are lexical (Sporleder and Lascarides, 2008; Pitler et al., 2009; Lin et al., 2009; Hernault et al., 2010; Park and Cardie, 2012). Early studies have suggested that lexical features, word pairs (cross-product of the words in the first and second argument) in particular, will be powerful predictors of discourse relations (Marcu and Echihabi, 2002; Blair-Goldensohn et al., 2007). The intuition behind word pairs was that semantic relations between the lexical items, such as *drought–famine*, *child–adult*, may in turn signal causal or contrast discourse relations. Later it has been shown that word pair features do not appear to capture such semantic relationship between words (Pitler et al., 2009) and that syntactic features lead to higher accuracies (Lin et al., 2009; Zhou et al., 2010; Park and Cardie, 2012). Recently, Biran and McKeown (2013) aggregated word pair features with explicit connectives and reported improvements over the original word pairs as features.

In this section, we show that the representation of lexical features play a direct role in feature sparsity and ultimately affects prediction performance.

The first two studies that specifically addressed the problem of predicting implicit discourse relations in the PDTB made use of very different instance representations. Pitler et al. (2009) represent instances of discourse relations in a vector space defined by word pairs, i.e. the cross-product of the words that appear in the two arguments of the relation. There, features are of the form (w_1, w_2) where $w_1 \in arg1$ and $w_2 \in arg2$. If there are N words in the entire vocabulary, the size of each instance would be $N \times N$.

In contrast, Lin et al. (2009) represent instances by tracking the occurrences of grammatical productions in the syntactic parse of argument spans. There are three indicator features associated with each production: whether the production appears in $arg1$, in $arg2$, and in both arguments. For a grammar with N production rules, the size of the vector representing an instance will be $3N$. For convenience we call this "binary representation", in contrast to the word-pair features in which the cross product of words constitute the representation. Note that the cross-product approach has been extended to a wide variety of features (Pitler et al., 2009; Park and Cardie, 2012). In the experiments that follow we will demonstrate that binary representations lead to less sparse features and higher prediction accuracy.

Lin et al. (2009) found that their syntactic features are more powerful than the word pair features. Here we show that the advantage comes not only from the inclusion of syntactic information but also from the less sparse instance representation they used for syntactic features. In Table 1 we show the number of features for each representation and the average F score and accuracy for word pairs and words with binary representation (*binary-lexical*). The results for each relation are shown in Table 8 and discussed in Section 7.

	# Features	Avg. F	Avg. Accuracy
word-pairs	92128	29.46	57.22
binary-lexical	12116	31.79	60.42

Table 1: F-scores and average accuracies of paired and binary representations of words.

Using binary representation for lexical information outperforms word pairs. Thus, the difference in how lexical information is represented accounts for a considerable portion of the improvement reported in Lin et al. (2009). Most notably, for the *Instantiation* class, we see a 7.7% increase in F-score. On average, the less sparse representation

[3] We also did not include features that occurred less than 5 times in the training set.

translates into 2.34% absolute improvement in F-score and 3.2% absolute improvement in accuracy. From this point on we adopt the binary representation for the features discussed.

5 Sparsity and syntactic features

Grammatical production rules were first used for discourse relation representation in Lin et al. (2009). They were identified as the most suitable representation, that lead to highest performance in a couple of independent studies (Lin et al., 2009; Park and Cardie, 2012). The comparison representations covered a number of semantic classes related to sentiment, polarity and verb information and dependency representations of syntax.

Production rules correspond to tree chunks in the constituency parse of a sentence, i.e. a node in the syntactic parse tree with all of its children, which in turn correspond to grammar rules applied in the derivation of the tree, such as S→NP VP. This syntactic representation subsumes lexical representations because of the production rules with part-of-speech on the left-hand side and a lexical item on the right-hand side.

We propose that the sparsity of production rules can be reduced even further by introducing a new representation of the parse tree. Specifically, instead of having full production rules where a single feature records the parent and all its children, all (parent,child) pairs in the constituency parse tree are used. For example, the rule S→NP VP will now become two features, S→NP and S→VP. Note that the leaves of the tree, i.e. the part-of-speech→word features are not changed. For ease of reference we call this new representation "production sticks". In this section we show that F scores and accuracies for implicit discourse relation prediction based on production sticks is significantly higher than using full production rules.

First, Table 2 illustrates the contrast in sparsity among the lexical, production rule and stick representations. The table gives the rate of occurrence of each feature class, which is defined as the average fraction of features with non-zero values in the representation of instances in the entire training set. Specifically, let N be the total number of features, m_i be the number of features triggered in instance i, then the rate of occurrence is $\frac{m_i}{N}$.

The table clearly shows that the number of features in the three representations is comparable, but they vary notably in their rate of occurrence.

	# Features	Rate of Occurrence
sticks	14,165	0.00623
prodrules	16,173	0.00374
binary-lexical	12,116	0.00276
word-pairs	92,128	0.00113

Table 2: Number of features and rate of occurrence for binary lexical representation, production rules and sticks.

	Avg. F	Avg. Accuracy
sticks	34.73	64.89
prodrules	33.69	63.55
binary-lexical	31.79	60.42
word-pairs	29.46	57.22

Table 3: F-scores and average accuracies of production rules and production sticks.

Sticks have almost twice the rate of occurrence of that of full production rules. Both syntactic representations have much larger rate of occurrence than lexical features, and the rate of occurrence of word pairs is more than twice smaller than that of the binary lexical representation.

Next, in Table 3, we give binary classification prediction results based on both full rules and sticks. The first two rows of Table 3 compare full production rules (*prodrules*) with production sticks (*sticks*) using the binary representation. They both outperform the binary lexical representation. Again our results confirm that the better performance of production rule features is partly because they are less sparse than lexical representations, with an average of 1.04% F-score increase. Individually the F scores of 6 of the 7 relations are improved as shown in Table 8.

6 How important are lexical features?

Production rules or sticks include lexical items with their part-of-speech tags. These are the subset of features that contribute most to sparsity issues. In this section we test if these lexical features contribute to the performance or if they can be removed without noticeable degradation due to its intrinsic sparsity. It turns out that it is not advisable to remove the lexical features entirely, as performance decreases substantially if we do so.

6.1 Classification without lexical items

We start our exploration of the influence of lexical items on the accuracy of prediction by inspecting the performances of the classifiers with production rules and sticks, but without the lexical items and their parts of speech. Table 4 lists the average F

	Avg. F	Avg. Accuracy
prodrules	33.69	63.55
sticks	34.73	64.89
prodrules-nolex	32.30	62.03
sticks-nolex	33.86	63.99

Table 4: F-scores and average accuracies of production rules and sticks, with (rows 1-2) and without (rows 3-4) lexical items.

	# Features	Rate of Occurrence
prodrules	16,173	0.00374
sticks	14,165	0.00623
prodrules-nolex	3470	0.00902
sticks-nolex	922	0.0619

Table 5: Number of features and rate of occurrence for production rules and sticks, with (rows 1-2) and without (rows 3-4) lexical items.

scores and accuracies. Table 8 provides detailed results for individual relations. Here *prodrules-nolex* and *sticks-nolex* denote full production rules without lexical items, and production sticks without lexical items, respectively. In all but two relations, lexical items contribute to better classifier performance.

When lexical items are not included in the representation, the number of features is reduced to fewer than 30% of that in the original full production rules. At the same time however, including the lexical items in the representation improves performance even more than introducing the less sparse production stick representation. Production sticks with lexical information also perform better than the same representation without the POS-word sticks.

The number of features and their rates of occurrences are listed in Table 5. It again confirms that the less sparse stick representation leads to better classifier performance. Not surprisingly, purely syntactic features (without the lexical items) are much less sparse than syntax features with lexical items present. However the classifier performance is worse without the lexical features. This contrast highlights the importance of a reasonable tradeoff between attempts to reduce sparsity and the need to preserve lexical features.

6.2 Feature selection

So far our discussion was based on the behavior of models trained on a complete set of relatively frequent syntactic and lexical features (occurring more than five times in the training data). Feature selection is a way to reasonably prune out the set and reduce sparsity issues in the model. In fact feature selection has been used in the majority of prior work (Pitler et al., 2009; Lin et al., 2009; Park and Cardie, 2012).

Relation	%-nonlex	%-allfeats
Temporal	25.56	10.95
Comparison	25.40	15.51
Contingency	20.12	25.05
Conjunction	21.15	19.20
Instantiation	25.08	16.16
Restatement	22.16	17.35
Expansion	18.36	18.66

Table 6: Non-lexical features selected using feature selection. %-nonlex records the percentage of non-lexical features among all features selected; %-allfeats records the percentage of selected non-lexical features among all non-lexical features.

Here we perform feature selection and examine the proportion of syntactic and lexical features among the most informative features. We use the χ^2 test of independence, computed on the following contingency table for each feature F_i and for each relation R_j:

$$\begin{array}{c|c} F_i \wedge R_j & F_i \wedge \neg R_j \\ \hline \neg F_i \wedge R_j & \neg F_i \wedge \neg R_j \end{array}$$

Each cell in the above table records the number of training instances in which F_i and R_j are present or absent. We set our level of confidence to $p < 0.1$.

Table 6 lists the proportions of non-lexical items among the most informative features selected (column 2). It also lists the percentage of selected non-lexical items among all the 922 purely syntactic features from production rule and production stick representations (column 3). For all relations, at most about a quarter of the most informative features are non-lexical and they only take up 10%-25% of all possible non-lexical features. The prediction results using only these features are either higher than or comparable to that without feature selection (sticks-χ^2 in Table 8). These numbers suggest that lexical terms play a significant role as part of the syntactic representations.

In Table 8 we record the F scores and accuracies for each relation under each feature representation. The representations are sorted according to descending F scores for each relation. Notice that χ^2 feature selection on sticks is the best representation for the three smallest relations: *Comparison*, *Instantiation* and *Temporal*.

This finding led us to look into the selected lexical features for these three classes. We found that these most prominent features in fact capture some semantic information. We list the top ten most predictive lexical features for these three relations below, with examples. Somewhat disturbingly, many of them are style or domain specific to the Wall Street Journal that PDTB was built on.

Comparison a1a2_NN_share a1a2_NNS_cents a1a2_CC_or a1a2_CD_million a1a2_QP_$ a1a2_NP_$ a2_RB_n't a1a2_NN_% a2_JJ_year a2_IN_of

For *Comparison* (contrast), the top lexical features are words that occur in both argument 1 and argument 2. Contrast within the financial domain, such as "share", "cents" and numbers between arguments are captured by these features. Consider the following example:

Ex. *Analyst estimate the value of the BellSouth proposal at about $115 to $125 a share.* [Implicit=AND] **They value McCaw's bid at $112 to $118 a share .**

Here the contrast clearly happens with the value estimation for two different parties.

Instantiation a2_SINV_" a2_SINV_, a2_SINV_" a2_SINV_. a1_DT_some a2_S_ a2_VBZ_says a1_NP_, a2_NP_, a1_DT_a

For *Instantiation* (arg2 gives an example of arg1), besides words such as "some" or "a" that sometimes mark a set of events, many attribution features are selected. it turns out many *Instantiation* instances in the PDTB involve argument 2 being an inverted declarative sentence that signals a quote as illustrate by the following example:

Ex. *Unease is widespread among exchange members.* [Implicit=FOR EXAMPLE] **" I can't think of any reason to join Lloyd's now, "** says Keith Whitten, a British businessman and a Lloyd's member since 1979.

Temporal a1_VBD_plunged a2_VBZ_is a2_RB_later a1_VBD_was a2_VBD_responded a1a2_PRP_he a1_WRB_when a1_PRP_he a1_VBZ_is a2_VBP_are

For *Temporal*, verbs like *plunge* and *responded* are selected. Words such as *plunged* are quite domain specific to stock markets, but words such as *later* and *responded* are likely more general indicators of the relation.

The presence of pronouns was also a predictive feature. Consider the following example:

Ex. *A Yale law school graduate , he began his career in corporate law and then put in years at Metromedia Inc. and the William Morris talent agency.* [Implicit=THEN] **In 1976, he joined CBS Sports to head business affairs and, five years later, became its president.**

Overall, it is fairly easy to see that certain semantic information was captured by these features, such as similar structures in a pair of sentences holding a contrast relation, the use of verbs in a *Temporal* relation. However, it is rather unsettling to also see that some of these characteristics are largely style or domain specific. For example, for an *Instantiation* in an educational scenario where the tutor provides an example for a concept, it is highly unlikely that attribution features will be helpful. Therefore, part of the question of finding a general class of features that carry over to other styles or domains of text still remain unanswered.

7 Per-relation evaluation

Table 8 lists the F-scores and accuracies of each representation mentioned in this work for predicting individual relation classes. For each relation, the representations are ordered by decreasing F-score. We tested the results for statistical significance of the change in F-score. We compare all the representations with the best and the worse representations for the relation. A "Y" marks a significance level of $p \leq 0.05$ for the comparison with the best or worst representation, a "T" marks a significance level of $p \leq 0.1$, which means a tendency towards significance.

For all relations, production sticks, either with or without feature selection, is the top representation. Sticks without lexical items also underperform those including the lexical items for 6 of the 7 relations. Notably, production rules without lexical items are among the three worst representations, outperforming only the pure lexical features in some cases. This is a strong indication that being both a sparse syntactic representation and lacking lexical information, these features are not favored in this task. Pure lexical features give the worst or second to worst F scores, significantly worse than the alternatives in most of the cases.

In Table 7 we list the binary classification results from prior work: feature selected word pairs (Pitler et al., 2009), aggregated word pairs (Biran and McKeown, 2013), production rules only (Park and Cardie, 2012), and the best combination possible from a variety of features (Park and Cardie, 2012), all of which include production rules. We aim to compare the relative gains in performance with different representations. Note that the absolute results from prior work are not exactly comparable to ours for two reasons — the training

Sys.	Pitler et al.	Biran-McKeown
Feat.	wordpair-implicit	aggregated wp
Comp.	20.96 (42.55)	24.38 (61.72)
Cont.	43.79 (61.92)	44.03 (66.78)
Expa.	63.84 (60.28)	66.48 (60.93)
Temp.	16.21 (61.98)	19.54 (68.09)
Sys.	Park-Cardie	Park-Cardie
Feat.	prodrules	best combination
Comp.	30.04 (75.84)	31.32 (74.66)
Cont.	47.80 (71.90)	49.82 (72.09)
Expa.	77.64 (69.60)	79.22 (69.14)
Temp.	20.96 (63.36)	26.57 (79.32)

Table 7: F-score (accuracy) of prior systems. Note that the absolute numbers are not exactly comparable with ours because of the important reasons explained in this section.

and testing sets are different; how *Expansion*, *EntRel/NoRel* and *AltLex* relations are treated differently in each work. The only meaningful indicator here is the absolute size of improvement. The table shows that our introduction of production sticks led to improvements comparable to those reported in prior work.

The aggregated word pair is a less sparse version of the word pair features, where each pair is converted into weights associated with an explicit connective. Just as the less sparse binary lexical representation presented previously, the aggregated word pairs also gave better performance. None of the three lexical features, however, surpasses raw production rules, which again echoes our finding that binary lexical features are not better than the full production rules. Finally, we note that a combination of features gives better F-scores.

8 Discussion: are the features complementary?

So far we have discussed how different representations for lexical and syntactic features can affect the classifier performances. We focused on the dilemma of how to reduce sparsity while still preserving the useful lexical features. An important question remains as whether these representations are complementary, that is, how different is the classifier behaving under different feature sets and if it makes sense to combine the features.

We compare the classifier output on the test data with two methods in Table 9: the Q-statistic and the percentage of data which the two classifiers disagree (Kuncheva and Whitaker, 2003).

Representation	F (A)	sig-best	sig-worst
	Comparison		
sticks-χ^2	27.78 (62.83)	N/A	Y
prodrules	27.65 (59.5)	-	Y
sticks	27.50 (60.73)	-	Y
sticks-nolex	27.01 (59.63)	-	Y
prodrules-nolex	26.40 (58.47)	T	Y
binary-lexical	24.73 (58.32)	Y	-
word-pairs	22.68 (45.03)	Y	N/A
	Conjunction		
sticks	27.55 (63.82)	N/A	T
sticks-χ^2	27.53 (64.06)	-	T
prodrules	27.02 (63.91)	-	-
sticks-nolex	26.56 (61.03)	T	-
binary-lexical	25.90 (61.77)	Y	-
prodrules-nolex	25.20 (62.83)	T	N/A
word-pairs	25.18 (74.51)	T	-
	Contingency		
sticks	48.90 (67.49)	N/A	Y
sticks-χ^2	48.55 (67.76)	-	Y
sticks-nolex	48.08 (67.69)	-	Y
prodrules	47.14 (65.61)	T	Y
prodrules-nolex	45.79 (63.99)	Y	Y
binary-lexical	44.17 (62.68)	Y	Y
word-pairs	40.57 (50.53)	Y	N/A
	Expansion		
sticks	56.48 (61.75)	N/A	Y
sticks-χ^2	56.30 (62.26)	-	Y
sticks-nolex	55.43 (60.56)	-	Y
prodrules	55.42 (61.05)	-	Y
binary-lexical	54.20 (59.26)	Y	-
word-pairs	53.65 (56.64)	Y	-
prodrules-nolex	53.53 (58.79)	Y	N/A
	Instantiation		
sticks-χ^2	30.34 (74.54)	N/A	Y
sticks	29.93 (73.80)	-	Y
prodrules	29.59 (72.20)	-	Y
sticks-nolex	28.22 (72.66)	Y	Y
prodrules-nolex	27.83 (70.72)	Y	Y
binary-lexical	27.29 (70.05)	Y	Y
word-pairs	20.22 (51.00)	Y	N/A
	Restatement		
sticks	35.74 (61.45)	N/A	Y
sticks-χ^2	34.93 (61.42)	-	Y
sticks-nolex	34.62 (61.08)	T	Y
prodrules	33.52 (58.54)	T	Y
prodrules-nolex	32.05 (56.84)	Y	-
binary-lexical	31.27 (57.41)	Y	T
word-pairs	29.81 (47.42)	Y	N/A
	Temporal		
sticks-χ^2	17.97 (66.67)	N/A	Y
sticks-nolex	17.08 (65.27)	T	Y
sticks	17.04 (65.22)	T	Y
prodrules	15.51 (64.04)	Y	-
prodrules-nolex	15.29 (62.56)	Y	-
binary-lexical	14.97 (61.92)	Y	-
word-pairs	14.10 (75.38)	Y	N/A

Table 8: F-score (accuracy) of each relation for each feature representation. The representations in each relation are sorted in descending order. The column "sig-best" marks the significance test result against the best representation, the column "sig-worst" marks the significance test result against the worst representation. "Y" denotes $p \leq 0.05$, "T" denotes $p \leq 0.1$.

Q-statistic is a measure of agreement between two systems s_1 and s_2 formulated as follows:

$$Q_{s_1,s_2} = \frac{N_{11}N_{00} - N_{01}N_{10}}{N_{11}N_{00} + N_{01}N_{10}}$$

Where N denotes the number of instances, a subscript 1 on the left means s_1 is correct, and a subscript 1 on the right means s_2 is correct.

There are several rather surprising findings. Most notably, word pairs and binary lexical representations give very different classification results in each relation. Their predictions disagree on at least 25% of the data. This finding drastically contrast the fact that they are both lexical features and that they both make use of the argument annotations in the PDTB. A comparison of the percentages and their differences in F scores or accuracies easily shows that it is not the case that binary lexical models correctly predict instances word pairs made mistakes on, but that they are disagreeing in both ways. Thus, given the previous discussion that lexical items are useful, it is possible the most suitable representation would combine both views of lexical distribution.

Even more surprisingly, the difference in classifier behavior is not as big when we compare lexical and syntactic representations. The disagreement of production sticks with and without lexical features are the smallest, even though, as we have shown previously, the majority of production sticks are lexical features with part-of-speech tags. If we compare binary lexical features with production sticks, the disagreement becomes bigger, but still not as big as word pairs vs. binary lexical.

Besides the differences in classification, the bigger picture of improving implicit discourse relation classification is finding a set of feature representations that are able to complement each other to improve the classification. A direct conclusion here is that one should not limit the focus on features in different categories (for example, lexical or syntax), but also features in the same category represented differently (for example, word pairs or binary lexical).

Rel.	Q-stat	Disagreement
word-pairs vs. binary-lexical		
Comparison	0.65	33.55
Conjunction	0.71	28.47
Contingency	0.81	26.35
Expansion	0.69	29.38
Instantiation	0.75	31.33
Restatement	0.76	28.42
Temporal	0.25	25.34
binary-lexical vs. sticks		
Comparison	0.78	25.49
Conjunction	0.78	24.67
Contingency	0.86	20.68
Expansion	0.80	24.28
Instantiation	0.83	20.75
Restatement	0.76	26.72
Temporal	0.86	20.61
sticks vs. prodrules		
Comparison	0.88	19.77
Conjunction	0.89	18.43
Contingency	0.94	14.00
Expansion	0.88	19.18
Instantiation	0.90	16.34
Restatement	0.89	18.88
Temporal	0.90	17.94
sticks vs. sticks-nolex		
Comparison	0.94	14.61
Conjunction	0.92	16.63
Contingency	0.97	10.16
Expansion	0.91	17.35
Instantiation	0.97	9.51
Restatement	0.97	11.26
Temporal	0.98	8.42

Table 9: Q statistic and disagreement of different classes of representations

9 Conclusion

In this work we study implicit discourse relation classification from the perspective of the interplay between lexical and syntactic feature representation. We are particularly interested in the trade-off between reducing sparsity and preserving lexical features. We first emphasize the important role of sparsity for traditional word-pair representations and how a less sparse representation could improve performance. Then we proposed a less sparse feature representation for production rules, the best feature category so far, that further improves classification. We study the role of lexical features and show the contrast between the sparsity problem they brought along and their dominant presence in the highly ranked features. Also, lexical features included in syntactic features that are most informative to the classifiers are found to be style or domain specific in certain relations. Finally, we compare the representations in terms of classifier disagreement and showed that within the same feature category different feature representation can also be complementary with each other.

References

Or Biran and Kathleen McKeown. 2013. Aggregated word pair features for implicit discourse relation disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL): Short Papers*, pages 69–73.

Sasha Blair-Goldensohn, Kathleen McKeown, and Owen Rambow. 2007. Building and refining rhetorical-semantic relation models. In *Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 428–435.

Hugo Hernault, Danushka Bollegala, and Mitsuru Ishizuka. 2010. A semi-supervised approach to improve classification of infrequent discourse relations using feature vector extension. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 399–409.

Thorsten Joachims. 1999. Making large-scale support vector machine learning practical. In *Advances in kernel methods*, pages 169–184.

Ludmila I. Kuncheva and Christopher J. Whitaker. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, May.

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 (EMNLP)*, pages 343–351.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, 20:151–184, 4.

Daniel Marcu and Abdessamad Echihabi. 2002. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, pages 368–375.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics - Special issue on using large corpora*, 19(2):313–330.

Joonsuk Park and Claire Cardie. 2012. Improving implicit discourse relation recognition through feature set optimization. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 108–112.

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 (ACL-IJCNLP)*, pages 683–691.

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 Sixth International Conference on Language Resources and Evaluation (LREC)*.

Caroline Sporleder and Alex Lascarides. 2008. Using automatically labelled examples to classify rhetorical relations: An assessment. *Natural Language Engineering*, 14(3):369–416, July.

Zhi-Min Zhou, Yu Xu, Zheng-Yu Niu, Man Lan, Jian Su, and Chew Lim Tan. 2010. Predicting discourse connectives for implicit discourse relation recognition. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 1507–1514.