

Predicting Discourse Connectives for Implicit Discourse Relation Recognition

Zhi-Min Zhou and Yu Xu
East China Normal University
51091201052@ecnu.cn

Zheng-Yu Niu
Toshiba China R&D Center
zhengyu.niu@gmail.com

Man Lan and Jian Su
Institute for Infocomm Research
sujian@i2r.a-star.edu.sg

Chew Lim Tan
National University of Singapore
tancl@comp.nus.edu.sg

Abstract

Existing works indicate that the absence of explicit discourse connectives makes it difficult to recognize implicit discourse relations. In this paper we attempt to overcome this difficulty for implicit relation recognition by automatically inserting discourse connectives between arguments with the use of a language model. Then we propose two algorithms to leverage the information of these predicted connectives. One is to use these predicted implicit connectives as additional features in a supervised model. The other is to perform implicit relation recognition based only on these predicted connectives. Results on Penn Discourse Treebank 2.0 show that predicted discourse connectives help implicit relation recognition and the first algorithm can achieve an absolute average f-score improvement of 3% over a state of the art baseline system.

1 Introduction

Discourse relation analysis is to automatically identify discourse relations (e.g., explanation relation) that hold between arbitrary spans of text. This analysis may be a part of many natural language processing systems, e.g., text summarization system, question answering system. If there are discourse connectives between textual units to explicitly mark their relations, the recognition task on these texts is defined as *explicit discourse relation recognition*. Otherwise it is defined as *implicit discourse relation recognition*.

Previous study indicates that the presence of discourse connectives between textual units can greatly help relation recognition. In Penn Discourse Treebank (PDTB) corpus (Prasad et al., 2008), the most general senses, i.e., Comparison (Comp.), Contingency (Cont.), Temporal (Temp.) and Expansion (Exp.), can be disambiguated in explicit relations with more than 90% f-scores based only on the discourse connectives explicitly used to signal the relation (Pitler and Nenkova., 2009b).

However, for implicit relations, there are no connectives to explicitly mark the relations, which makes the recognition task quite difficult. Some of existing works attempt to perform relation recognition without hand-annotated corpora (Marcu and Echihiabi, 2002), (Sporleder and Lascarides, 2008) and (Blair-Goldensohn, 2007). They use unambiguous patterns such as [Arg1, *but* Arg2] to create synthetic examples of implicit relations and then use [Arg1, Arg2] as an training example of an implicit relation. Another research line is to exploit various linguistically informed features under the framework of supervised models, (Pitler et al., 2009a) and (Lin et al., 2009), e.g., polarity features, semantic classes, tense, production rules of parse trees of arguments, etc.

Our study on PDTB test data shows that the average f-score for the most general 4 senses can reach 91.8% when we simply mapped the ground truth implicit connective of each test instance to its most frequent sense. It indicates the importance of connective information for implicit relation recognition. However, so far there is no previous study attempting to use such kind of connective information for implicit relation. One possi-

ble reason is that implicit connectives do not exist in unannotated real texts. Another evidence of the importance of connectives for implicit relations is shown in PDTB annotation. The PDTB annotation consists of inserting a connective expression that best conveys the inferred relation by the readers. Connectives inserted in this way to express inferred relations are called *implicit connectives*, which do not exist in real texts. These evidences inspire us to consider two interesting research questions:

- (1) Can we automatically predict implicit connectives between arguments?
- (2) How to use the predicted implicit connectives to build an automatic discourse relation analysis system?

In this paper we address these two questions as follows: (1) We insert appropriate discourse connectives between two textual units with the use of a language model. Here we train the language model on large amount of raw corpora without the use of any hand-annotated data. (2) Then we present two algorithms to use these predicted connectives for implicit relation recognition. One is to use these connectives as additional features in a supervised model. The other is to perform relation recognition based only on these connectives.

We performed evaluation of the two algorithms and a baseline system on PDTB 2.0 corpus. Experimental results showed that using predicted discourse connectives as additional features can significantly improve the performance of implicit discourse relation recognition. Specifically, the first algorithm achieved an absolute average f-score improvement of 3% over a state of the art baseline system.

The rest of this paper is organized as follows. Section 2 describes the two algorithms for implicit discourse relation recognition. Section 3 presents experiments and results on PDTB data. Section 4 reviews related work. Section 5 concludes this work.

2 Our Algorithms for Implicit Discourse Relation Recognition

2.1 Prediction of implicit connectives

Explicit discourse relations are easily identifiable due to the presence of discourse connectives between arguments. (Pitler and Nenkova., 2009b) showed that in PDTB corpus, the most general senses, i.e., Comparison (Comp.), Contingency (Cont.), Temporal (Temp.) and Expansion (Exp.), can be disambiguated in explicit relations with more than 90% f-scores based only on discourse connectives.

But for implicit relations, there are no connectives to explicitly mark the relations, which makes the recognition task quite difficult. PDTB data provides *implicit connectives* that are inserted between paragraph-internal adjacent sentence pairs not marked by any of explicit connectives. The availability of ground-truth implicit connectives makes it possible to evaluate the contribution of these connectives for implicit relation recognition. Our initial study on PDTB data show that the average f-score for the most general 4 senses can reach 91.8% when we obtained the sense of each test example by mapping each ground truth implicit connective to its most frequent sense. We see that connective information is an important knowledge source for implicit relation recognition. However these implicit connectives do not exist in real texts. In this paper we overcome this difficulty by inserting a connective between two arguments with the use of a language model.

Following the annotation scheme of PDTB, we assume that each implicit connective takes two arguments, denoted as $Arg1$ and $Arg2$. Typically, there are two possible positions for most of implicit connectives¹, i.e., the position before $Arg1$ and the position between $Arg1$ and $Arg2$. Given a set of possible implicit connectives $\{c_i\}$, we generate two synthetic sentences, $c_i+Arg1+Arg2$ and $Arg1+c_i+Arg2$ for each c_i , denoted as $S_{c_i,1}$ and $S_{c_i,2}$. Then we calculate the perplexity (an intrinsic score) of these sentences with the use of a language model, denoted as $PPL(S_{c_i,j})$. According

¹For parallel connectives, e.g., *if...then...*, the two connectives will take the two arguments together, so there is only one possible combination for connectives and arguments.

to the value of $PPL(S_{c_i,j})$ (the lower the better), we can rank these sentences and select the connectives in top N sentences as implicit connectives for this argument pair. The language model may be trained on large amount of unannotated corpora that can be cheaply acquired, e.g., North American News corpus.

2.2 Using predicted implicit connectives as additional features

We predict implicit connectives on both training set and test set. Then we can use the predicted implicit connectives as additional features for supervised implicit relation recognition. Previous works exploited various linguistically informed features under the framework of supervised models. In this paper, we include 9 types of features in our system due to their superior performance in previous studies, e.g., polarity features, semantic classes of verbs, contextual sense, modality, inquirer tags of words, first-last words of arguments, cross-argument word pairs, ever used in (Pitler et al., 2009a), production rules of parse trees of arguments used in (Lin et al., 2009), and intra-argument word pairs inspired by the work of (Saito et al., 2006).

Here we provide the details of the 9 features, shown as follows:

Verbs: Similar to the work in (Pitler et al., 2009a), the verb features consist of the number of pairs of verbs in Arg1 and Arg2 if they are from the same class based on their highest Levin verb class level (Dorr, 2001). In addition, the average length of verb phrase and the part of speech tags of main verb are also included as verb features.

Context: If the immediately preceding (or following) relation is an explicit, its relation and sense are used as features. Moreover, we use another feature to indicate if Arg1 leads a paragraph.

Polarity: We use the number of positive, negated positive, negative and neutral words in arguments and their cross product as features. For negated positives, we locate the negated words in text span and then define the closely behind positive word as negated positive.

Modality: We look for modal words including their various tenses or abbreviation forms in both arguments. Then we generate a feature to indicate

the presence or absence of modal words in both arguments and their cross product.

Inquirer Tags: Inquirer Tags extracted from General Inquirer lexicon (Stone et al., 1966) contains positive or negative classification of words. In fact, its fine-grained categories, such as Fall versus Rise, or Pleasure versus Pain, can indicate the relation between two words, especially for verbs. So we choose the presence or absence of 21 pair categories with complementary relation in Inquirer Tags as features. We also include their cross production as features.

FirstLastFirst3: We choose the first and last words of each argument as features, as well as the pair of first words, the pair of last words, and the first 3 words in each argument. In addition, we apply Porter’s Stemmer (Porter, 1980) to each word before preparation of these features.

Production Rule: According to (Lin et al., 2009), we extract all the possible production rules from arguments, and check whether the rules appear in Arg1, Arg2 and both arguments. We remove the rules occurring less than 5 times in training data.

Cross-argument Word Pairs: We perform the Porter’s stemming (Porter, 1980), and then group all words from Arg1 and Arg2 into two sets W_1 and W_2 respectively. Then we generate any possible word pair (w_i, w_j) ($w_i \in W_1, w_j \in W_2$). We remove the word pairs with less than 5 times.

Intra-argument Word Pairs: Let $Q_1 = (q_1, q_2, \dots, q_n)$ be the word sequence of Arg1. The intra-argument word pairs for Arg1 is defined as $WP_1 = ((q_1, q_2), (q_1, q_3), \dots, (q_1, q_n), (q_2, q_3), \dots, (q_{n-1}, q_n))$. We extract all the intra-argument word pairs from Arg1 and Arg2 and remove word pairs appearing less than 5 times in training data.

2.3 Relation recognition based only on predicted implicit connectives

After the prediction of implicit connectives, we can address the implicit relation recognition task with the methods for explicit relation recognition due to the presence of implicit connectives, e.g., sense classification based only on connectives (Pitler and Nenkova., 2009b). The work of (Pitler and Nenkova., 2009b) showed that most

of connectives are unambiguous and it is possible to obtain high performance in prediction of discourse sense due to the simple mapping relation between connectives and senses. Given two examples:

(E1) She paid less on her dress, *but* it is very nice.

(E2) We have to hurry up *because* the raining is getting heavier and heavier.

The two connectives, i.e., *but* in E1 and *because* in E2, convey Comparison and Contingency sense respectively. In most cases, we can easily recognize the relation sense by the appearance of discourse connective since it can be interpreted in only one way. That means, the ambiguity of the mapping between sense and connective is quite few.

We count the frequency of sense tags for each possible connective on PDTB training data for implicit relation. Then we build a sense recognition model by simply mapping each connective to its most frequent sense. Here we do not perform connective prediction on training data. During testing, we use the language model to insert implicit connectives into each test argument pair. Then we perform relation recognition by mapping each implicit connective to its most frequent sense.

3 Experiments and Results

3.1 Experiments

3.1.1 Data sets

In this work we used the PDTB 2.0 corpus for evaluation of our algorithms. Following the work of (Pitler et al., 2009a), we used sections 2-20 as training set, sections 21-22 as test set, and sections 0-1 as development set for parameter optimization. For comparison with the work of (Pitler et al., 2009a), we ran four binary classification tasks to identify each of the main relations (Cont., Comp., Exp., and Temp.) from the rest. For each relation, we used equal numbers of positive and negative examples as training data². The negative examples were chosen at random from sections 2-20. We used all the instances in sections 21 and 22 as test set, so the test set is representative of

²Here the numbers of training and test instances for Expansion relation are different from those in (Pitler et al., 2009a). The reason is that we do not include instances of EntRel as positive examples.

the natural distribution. The numbers of positive and negative instances for each sense in different data sets are listed in Table 1.

Table 1: Statistics of positive and negative samples in training, development and test sets for each relation.

Relation	Train	Dev	Test
	Pos/Neg	Pos/Neg	Pos/Neg
Comp.	1927/1927	191/997	146/912
Cont.	3375/3375	292/896	276/782
Exp.	6052/6052	651/537	556/502
Temp.	730/730	54/1134	67/991

In this work we used LibSVM toolkit to construct four linear SVM models for a baseline system and the system in Section 2.2.

3.1.2 A baseline system

We first built a baseline system, which used 9 types of features listed in Section 2.2.

We tuned the numbers of firstLastFirst3, cross-argument word pair, intra-argument word pair on development set. Finally we set the frequency threshold at 3, 5 and 5 respectively.

3.1.3 Prediction of implicit connectives

To predict implicit connectives, we adopt the following two steps:(1) train a language model; (2) select top N implicit connectives.

Step 1: We used SRILM toolkit to train the language models on three benchmark news corpora, i.e., New York part in the BLLIP North American News, Xin and Ltw parts of English Gigaword (4th Edition). We also tried different values for n in n -gram model. The parameters were tuned on the development set to optimize the accuracy of prediction. In this work we chose 3-gram language model trained on NY corpus.

Step 2: We combined each instance’s Arg1 and Arg2 with connectives extract from PDTB2 (100 in all). There are two types of connectives, single connective (e.g. *because* and *but*) and parallel connective (such as “*not only ... , but also*”). Since discourse connectives may appear not only ahead of the Arg1, but also between Arg1 and Arg2, we considered this case. Given a set of possible implicit connectives $\{c_i\}$, for single connective $\{c_i\}$, we constructed two synthetic sentences, $c_i+\text{Arg1}+\text{Arg2}$ and $\text{Arg1}+c_i+\text{Arg2}$. In case of

parallel connective, we constructed one synthetic sentence like $c_{i1} + \text{Arg1} + c_{i2} + \text{Arg2}$.

As a result, we can get 198 synthetic sentences for each argument pair. Then we converted all words to lower cases and used the language model trained in the above step to calculate perplexity on sentence level. The perplexity scores were ranked from low to high. For example, we got the perplexity (ppl) for two sentences as follows:

(1) *but* this is an old story, we're talking about years ago before anyone heard of asbestos having any questionable properties.

$ppl = 652.837$

(2) this is an old story, *but* we're talking about years ago before anyone heard of asbestos having any questionable properties.

$ppl = 583.514$

We considered the combination of connectives and their position as final features like *mid_but*, *first_but*, where the features are binary, that is, the presence and absence of the specific connective.

According to the value of $PPL(S_{c_i,j})$ (the lower the better), we selected the connectives in top N sentences as implicit connectives for this argument pair. In order to get the optimal N value, we tried various values of N on development set and selected the minimum value of N so that the ground-truth connectives appeared in top N connectives. The final N value is set to 60 based on the trade-off between performance and efficiency.

3.1.4 Using predicted connectives as additional features

This system combines the predicted implicit connectives as additional features and the 9 types of features in an supervised framework. The 9 types of features are listed as shown in Section 2.2 and tuned on development set.

We combined predicted connectives with the best subset features from the development data set with respect to f-score. In our experiment of selecting best subset features, single features rather than the combination of several features achieved much higher scores. So we combine single features with predicted connectives as final features.

3.1.5 Using only predicted connectives for implicit relation recognition

We built two variants for the algorithm in Section 2.3. One is to use the data for explicit relations in PDTB sections 2-20 as training data. The other is to use the data for implicit relations in PDTB sections 2-20 as training data. Given training data, we obtained the most frequent sense for each connective appearing in the training data. Then given test data, we recognized the sense of each argument pair by mapping each predicted connective to its most frequent sense. In this work we conducted another experiment to see the upper-bound performance of this algorithm. Here we performed recognition based on ground-truth implicit connectives and used the data for implicit relations as training data.

3.2 Results

3.2.1 Result of baseline system

Table 2 summarizes the best performance achieved by the baseline system in comparison with previous state-of-the-art performance achieved in (Pitler et al., 2009a). The first two lines in the table show their best results using single feature and using combined feature subset. It indicates that the performance of using combined feature subset is higher than that using single feature alone.

From this table, we can find that our baseline system has a comparable result on Continuity and Temporal. On Comparison, our system achieved a better performance around 9% f-score higher than their best result. However, for Expansion, they expanded both training and testing sets by including EntRel relation as positive examples, which makes it impossible to perform direct comparison. Generally, our baseline system is reasonable and thus the consequent experiments on it are reliable.

3.2.2 Result of algorithm 1: using predicted connectives as additional features

Table 3 summarizes the best performance achieved by the baseline system and the first algorithm (i.e., baseline + Language Model) on test set. The second and third column show the best performance achieved by the baseline system and

Table 2: Performance comparison of the baseline system with the system of (Pitler et al., 2009a) on test set.

System	Comp. vs. Not F_1 (Acc)	Cont. vs. Other F_1 (Acc)	Exp. vs. Other F_1 (Acc)	Temp. vs. Other F_1 (Acc)
Using the best single feature (Pitler et al., 2009a)	21.01(52.59)	36.75(62.44)	71.29(59.23)	15.93(61.20)
Using the best feature subset (Pitler et al., 2009a)	21.96(56.59)	47.13(67.30)	76.42(63.62)	16.76(63.49)
The baseline system	30.72(78.26)	45.38(40.17)	65.95(57.94)	16.46(29.96)

the first algorithm using predicted connectives as additional features.

Table 3: Performance comparison of the algorithm in Section 2.2 with the baseline system on test set.

Relation	Features	Baseline F_1 (Acc)	Baseline+LM F_1 (Acc)
Comp.	Production Rule	30.72 (78.26)	31.08(68.15)
	Context	24.66(42.25)	27.64(53.97)
	InquirerTags	23.31(73.25)	27.87(55.48)
	Polarity	21.11(40.64)	23.64(52.36)
	Modality	17.25(80.06)	26.17(55.20)
	Verbs	25.00(53.50)	31.79 (58.22)
Cont.	Production Rule	45.38 (40.17)	47.16 (48.96)
	Context	37.61(44.70)	34.74(48.87)
	Polarity	35.57(50.00)	43.33(33.74)
	InquirerTags	38.04(41.49)	42.22(36.11)
	Modality	32.18(66.54)	35.26(55.58)
	Verbs	40.44(54.06)	42.04(32.23)
Exp.	Context	48.34(54.54)	68.32(53.02)
	FirstLastFirst3	65.95 (57.94)	68.94(53.59)
	InquirerTags	61.29(52.84)	68.49(53.21)
	Modality	64.36(56.14)	68.9(52.55)
	Polarity	49.95(50.38)	68.62(53.40)
	Verbs	52.95(53.31)	70.11 (54.54)
Temp.	Context	13.52(64.93)	16.99(79.68)
	FirstLastFirst3	15.75(66.64)	19.70(64.56)
	InquirerTags	8.51(83.74)	19.20(56.24)
	Modality	16.46 (29.96)	19.97(54.54)
	Polarity	16.29(51.42)	20.30 (55.48)
	Verbs	13.88(54.25)	13.53(61.34)

From this table, we found that this additional feature obtained from language model showed significant improvements in almost four relations. Specifically, the top two improvements are on Expansion and Temporal relations, which improved 4.16% and 3.84% in f-score respectively. Although on Comparison relation there is only a slight improvement (+1.07%), our two best systems both got around 10% improvements of f-score over a state-of-the-art system in (Pitler et al., 2009a). As a whole, the first algorithm achieved 3% improvement of f-score over a state of the art baseline system. All these results indicate that predicted implicit connectives can help improve

the performance.

3.2.3 Result of algorithm 2: using only predicted connectives for implicit relation recognition

Table 4 summarizes the best performance achieved by the second algorithm in comparison with the baseline system on test set.

The experiment showed that the baseline system using just gold-truth implicit connectives can achieve an f-score of 91.8% for implicit relation recognition. It once again proved that implicit connectives make significant contributions for implicit relation recognition. This also encourages our future work on finding the most suitable connectives for implicit relation recognition.

From this table, we found that, using only predicted implicit connectives achieved a comparable performance to (Pitler et al., 2009a), although it was still a bit lower than our best baseline. But we should bear in mind that this algorithm only uses 4 features for implicit relation recognition and these 4 features are easy computable and fast run, which makes the system more practical in application. Furthermore, compared with other algorithms which require hand-annotated data for training, the performance of this second algorithm is acceptable if we take into account that no labeled data is used for model training.

3.3 Analysis

Experimental results on PDTB showed that using the predicted implicit connectives significantly improves the performance of implicit discourse relation recognition. Our first algorithm achieves an average f-score improvement of 3% over a state of the art baseline system. Specifically, for the relations: Comp., Cont., Exp., Temp., our first algorithm can achieve 1.07%, 1.78%, 4.16%, 3.84% f-score improvements over a state of the art baseline system. Since (Pitler et al., 2009a)

Table 4: Performance comparison of the algorithm in Section 2.3 with the baseline system on test set.

System	Comp. vs. Other F_1 (Acc)	Cont. vs. Other F_1 (Acc)	Exp. vs. Other F_1 (Acc)	Temp. vs. Other F_1 (Acc)
The baseline system	30.72(78.26)	45.38(40.17)	65.95(57.94)	16.46(29.96)
Our algorithm with training data for explicit relation	26.02(52.17)	35.72(51.70)	64.94(53.97)	13.76(41.97)
Our algorithm with training data for implicit relation	24.55(63.99)	16.26(70.79)	60.70(53.50)	14.75(70.51)
Sense recognition using gold-truth implicit connectives	94.08(98.30)	98.19(99.05)	97.79(97.64)	77.04(97.07)

used different selection of instances for Expansion sense³, we cannot make a direct comparison. However, we achieve the best f-score around 70%, which provide 5% improvements over our baseline system. On the other hand, the second proposed algorithm using only predicted connectives still achieves promising results for each relation. Specifically, the model for the Comparison relation achieves an f-score of 26.02% (5% over the previous work in (Pitler et al., 2009a)). Furthermore, the models for Contingency and Temporal relation achieve 35.72% and 13.76% f-score respectively, which are comparable to the previous work in (Pitler et al., 2009a). The model for Expansion relation obtains an f-score of 64.95%, which is only 1% less than our baseline system which consists of ten thousands of features.

4 Related Work

Existing works on automatic recognition of discourse relations can be grouped into two categories according to whether they used hand-annotated corpora.

One research line is to perform relation recognition without hand-annotated corpora.

(Marcu and Echihabi, 2002) used a pattern-based approach to extract instances of discourse relations such as Contrast and Elaboration from unlabeled corpora. Then they used word-pairs between two arguments as features for building classification models and tested their model on artificial data for implicit relations.

There are other efforts that attempt to extend the work of (Marcu and Echihabi, 2002). (Saito et al., 2006) followed the method of (Marcu and Echihabi, 2002) and conducted experiments with combination of cross-argument word pairs and phrasal

patterns as features to recognize implicit relations between adjacent sentences in a Japanese corpus. They showed that phrasal patterns extracted from a text span pair provide useful evidence in the relation classification. (Sporleder and Lascarides, 2008) discovered that Marcu and Echihabi’s models do not perform as well on implicit relations as one might expect from the test accuracies on synthetic data. (Blair-Goldensohn, 2007) extended the work of (Marcu and Echihabi, 2002) by refining the training and classification process using parameter optimization, topic segmentation and syntactic parsing.

(Lapata and Lascarides, 2004) dealt with temporal links between main and subordinate clauses by inferring the temporal markers linking them. They extracted clause pairs with explicit temporal markers from BLLIP corpus as training data.

Another research line is to use human-annotated corpora as training data, e.g., the RST Bank (Carlson et al., 2001) used by (Soricut and Marcu, 2003), adhoc annotations used by (?), (Baldrige and Lascarides, 2005), and the Graph-Bank (Wolf et al., 2005) used by (Wellner et al., 2006).

Recently the release of the Penn Discourse TreeBank (PDTB) (Prasad et al., 2008) benefits the researchers with a large discourse annotated corpora, using a comprehensive scheme for both implicit and explicit relations. (Pitler et al., 2009a) performed implicit relation classification on the second version of the PDTB. They used several linguistically informed features, such as word polarity, verb classes, and word pairs, showing performance increases over a random classification baseline. (Lin et al., 2009) presented an implicit discourse relation classifier in PDTB with the use of contextual relations, constituent Parse Features, dependency parse features and cross-argument word pairs.

³They expanded the Expansion data set by adding randomly selected EntRel instances by 50%, which is considered to significantly change data distribution.

In comparison with existing works, we investigated a new knowledge source, implicit connectives, for implicit relation recognition. Moreover, our two models can exploit both labeled and unlabeled data by training a language model on unlabeled data and then using this language model to generate implicit connectives for recognition models trained on labeled data.

5 Conclusions

In this paper we use a language model to automatically generate implicit connectives and then present two methods to use these connectives for recognition of implicit relations. One method is to use these predicted implicit connectives as additional features in a supervised model and the other is to perform implicit relation recognition based only on these predicted connectives. Results on Penn Discourse Treebank 2.0 show that predicted discourse connectives help implicit relation recognition and the first algorithm achieves an absolute average f-score improvement of 3% over a state of the art baseline system.

Acknowledgments

This work is supported by grants from National Natural Science Foundation of China (No.60903093), Shanghai Pujiang Talent Program (No.09PJ1404500) and Doctoral Fund of Ministry of Education of China (No.20090076120029).

References

- J. Baldrige and A. Lascarides. 2005. *Probabilistic head-driven parsing for discourse structure*. Proceedings of the Ninth Conference on Computational Natural Language Learning.
- L. Carlson, D. Marcu, and Ma. E. Okurowski. 2001. *Building a discourse-tagged corpus in the framework of rhetorical structure theory*. Proceedings of the Second SIG dial Workshop on Discourse and Dialogue.
- B. Dorr. LCS Verb Database. *Technical Report Online Software Database, University of Maryland, College Park, MD*, 2001.
- R. Girju. 2003. *Automatic detection of causal relations for question answering*. In ACL 2003 Workshops.
- S. Blair-Goldensohn. 2007. *Long-Answer Question Answering and Rhetorical-Semantic Relations*. Ph.D. thesis, Columbia University.
- M. Lapata and A. Lascarides. 2004. *Inferring Sentence-internal Temporal Relations*. Proceedings of the North American Chapter of the Association of Computational Linguistics.
- Z.H. Lin, M.Y. Kan and H.T. Ng. 2009. *Recognizing Implicit Discourse Relations in the Penn Discourse Treebank*. Proceedings of the 2009 Conference on EMNLP.
- D. Marcu and A. Echihiabi. 2002. *An Unsupervised Approach to Recognizing Discourse Relations*. Proceedings of the 40th ACL.
- E. Pitler, A. Louis, A. Nenkova. 2009. *Automatic sense prediction for implicit discourse relations in text*. Proceedings of the 47th ACL.
- E. Pitler and A. Nenkova. 2009. *Using Syntax to Disambiguate Explicit Discourse Connectives in Text*. Proceedings of the ACL-IJCNLP 2009 Conference Short Papers.
- M. Porter. 1980. An algorithm for suffix stripping. In *Program*, vol. 14, no. 3, pp.130-137.
- R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, B. Webber. 2008. *The Penn Discourse TreeBank 2.0*. Proceedings of LREC'08.
- M. Saito, K. Yamamoto, S. Sekine. 2006. *Using Phrasal Patterns to Identify Discourse Relations*. Proceeding of the HLTCNA Chapter of the ACL.
- R. Soricut and D. Marcu. *Sentence Level Discourse Parsing using Syntactic and Lexical Information*. Proceedings of HLT/NAACL 2003.
- C. Sporleder and A. Lascarides. 2008. *Using automatically labelled examples to classify rhetorical relations: an assessment*. Natural Language Engineering, Volume 14, Issue 03.
- P.J. Stone, J. Kirsh, and Cambridge Computer Associates. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press.
- B. Wellner, J. Pustejovsky, C. H. R. S., A. Rumshisky. 2006. *Classification of discourse coherence relations: An exploratory study using multiple knowledge sources*. Proceedings of the 7th SIGDIAL Workshop on Discourse and Dialogue.
- F. Wolf, E. Gibson, A. Fisher, M. Knight. 2005. *The Discourse GraphBank: A database of texts annotated with coherence relations*. Linguistic Data Consortium.