Kernel Based Discourse Relation Recognition with Temporal Ordering Information

WenTing Wang[1] **Jian Su**[1] **Chew Lim Tan**[2]
[1]Institute for Infocomm Research
1 Fusionopolis Way, #21-01 Connexis
Singapore 138632
{wwang,sujian}@i2r.a-star.edu.sg

[2]Department of Computer Science
University of Singapore
Singapore 117417
tacl@comp.nus.edu.sg

Abstract

Syntactic knowledge is important for discourse relation recognition. Yet only heuristically selected flat paths and 2-level production rules have been used to incorporate such information so far. In this paper we propose using tree kernel based approach to automatically mine the syntactic information from the parse trees for discourse analysis, applying kernel function to the tree structures directly. These structural syntactic features, together with other normal flat features are incorporated into our composite kernel to capture diverse knowledge for simultaneous discourse identification and classification for both explicit and implicit relations. The experiment shows tree kernel approach is able to give statistical significant improvements over flat syntactic path feature. We also illustrate that tree kernel approach covers more structure information than the production rules, which allows tree kernel to further incorporate information from a higher dimension space for possible better discrimination. Besides, we further propose to leverage on temporal ordering information to constrain the interpretation of discourse relation, which also demonstrate statistical significant improvements for discourse relation recognition on PDTB 2.0 for both explicit and implicit as well.

1 Introduction

Discourse relations capture the internal structure and logical relationship of coherent text, including *Temporal*, *Causal* and *Contrastive* relations etc. The ability of recognizing such relations between text units including identifying and classifying provides important information to other natural language processing systems, such as language generation, document summarization, and question answering. For example, *Causal* relation can be used to answer more sophisticated, non-factoid *'Why'* questions.

Lee et al. (2006) demonstrates that modeling discourse structure requires prior linguistic analysis on syntax. This shows the importance of syntactic knowledge to discourse analysis. However, most of previous work only deploys lexical and semantic features (Marcu and Echihabi, 2002; Pettibone and PonBarry, 2003; Saito et al., 2006; Ben and James, 2007; Lin et al., 2009; Pitler et al., 2009) with only two exceptions (Ben and James, 2007; Lin et al., 2009). Nevertheless, Ben and James (2007) only uses flat syntactic path connecting connective and arguments in the parse tree. The hierarchical structured information in the trees is not well preserved in their flat syntactic path features. Besides, such a syntactic feature selected and defined according to linguistic intuition has its limitation, as it remains unclear what kinds of syntactic heuristics are effective for discourse analysis.

The more recent work from Lin et al. (2009) uses 2-level production rules to represent parse tree information. Yet it doesn't cover all the other sub-trees structural information which can be also useful for the recognition.

In this paper we propose using tree kernel based method to automatically mine the syntactic

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 710–719,
Uppsala, Sweden, 11-16 July 2010.

information from the parse trees for discourse analysis, applying kernel function to the parse tree structures directly. These structural syntactic features, together with other flat features are then incorporated into our composite kernel to capture diverse knowledge for simultaneous discourse identification and classification. The experiment shows that tree kernel is able to effectively incorporate syntactic structural information and produce statistical significant improvements over flat syntactic path feature for the recognition of both explicit and implicit relation in Penn Discourse Treebank (PDTB; Prasad et al., 2008). We also illustrate that tree kernel approach covers more structure information than the production rules, which allows tree kernel to further work on a higher dimensional space for possible better discrimination.

Besides, inspired by the linguistic study on tense and discourse anaphor (Webber, 1988), we further propose to incorporate temporal ordering information to constrain the interpretation of discourse relation, which also demonstrates statistical significant improvements for discourse relation recognition on PDTB v2.0 for both explicit and implicit relations.

The organization of the rest of the paper is as follows. We briefly introduce PDTB in Section 2. Section 3 gives the related work on tree kernel approach in NLP and its difference with production rules, and also linguistic study on tense and discourse anaphor. Section 4 introduces the frame work for discourse recognition, as well as the baseline feature space and the SVM classifier. We present our kernel-based method in Section 5, and the usage of temporal ordering feature in Section 6. Section 7 shows the experiments and discussions. We conclude our works in Section 8.

2 Penn Discourse Tree Bank

The Penn Discourse Treebank (PDTB) is the largest available annotated corpora of discourse relations (Prasad et al., 2008) over 2,312 Wall Street Journal articles. The PDTB models discourse relation in the predicate-argument view, where a discourse connective (e.g., *but*) is treated as a predicate taking two text spans as its arguments. The argument that the discourse connective syntactically bounds to is called Arg2, and the other argument is called Arg1.

The PDTB provides annotations for both explicit and implicit discourse relations. An *explicit* relation is triggered by an explicit connective. Example (1) shows an explicit *Contrast* relation signaled by the discourse connective *'but'*.

(1). Arg1. *Yesterday, the retailing and financial services giant reported a 16% drop in third-quarter earnings to $257.5 million, or 75 cents a share, from a restated $305 million, or 80 cents a share, a year earlier.*
Arg2. <u>But</u> the news was even worse for Sears's core U.S. retailing operation, the largest in the nation.

In the PDTB, *local* implicit relations are also annotated. The annotators insert a connective expression that best conveys the inferred *implicit* relation between adjacent sentences within the same paragraph. In Example (2), the annotators select *'because'* as the most appropriate connective to express the inferred *Causal* relation between the sentences. There is one special label *AltLex* pre-defined for cases where the insertion of an *Implicit* connective to express an inferred relation led to a *redundancy* in the expression of the relation. In Example (3), the *Causal* relation derived between sentences is *alternatively lexicalized* by some *non-connective expression* shown in square brackets, so no *implicit* connective is inserted. In our experiments, we treat *AltLex* Relations the same way as normal *Implicit* relations.

(2). Arg1. *Some have raised their cash positions to record levels.*
Arg2. <u>Implicit = Because</u> High cash positions help buffer a fund when the market falls.

(3). Arg1. *Ms. Bartlett's previous work, which earned her an international reputation in the non-horticultural art world, often took gardens as its nominal subject.*
Arg2. [Mayhap this metaphorical connection made] the BPC Fine Arts Committee think she had a literal green thumb.

The PDTB also captures two non-implicit cases: (a) *Entity* relation where the relation between adjacent sentences is based on entity coherence (Knott et al., 2001) as in Example (4); and (b) *No* relation where no discourse or entity-based coherence relation can be inferred between adjacent sentences.

(4). But for South Garden, the grid was to be a 3-D network of masonry or hedge walls with real plants inside them.

In a Letter to the BPCA, kelly/varnell called *this* "arbitrary and amateurish."

Each *Explicit*, *Implicit* and *AltLex* relation is annotated with a sense. The *senses* in PDTB are arranged in a three-level hierarchy. The top level has four tags representing four major semantic classes: *Temporal*, *Contingency*, *Comparison* and *Expansion*. For each class, a second level of *types* is defined to further refine the semantic of the class levels. For example, *Contingency* has two types *Cause* and *Condition*. A third level of *subtype* specifies the semantic contribution of each argument. In our experiments, we use only the top level of the sense annotations.

3 Related Work

Tree Kernel based Approach in NLP. While the feature based approach may not be able to fully utilize the syntactic information in a parse tree, an alternative to the feature-based methods, tree kernel methods (Haussler, 1999) have been proposed to implicitly explore features in a high dimensional space by employing a kernel function to calculate the similarity between two objects directly. In particular, the kernel methods could be very effective at reducing the burden of feature engineering for structured objects in NLP research (Culotta and Sorensen, 2004). This is because a kernel can measure the similarity between two discrete structured objects by directly using the original representation of the objects instead of explicitly enumerating their features.

Indeed, using kernel methods to mine structural knowledge has shown success in some NLP applications like parsing (Collins and Duffy, 2001; Moschitti, 2004) and relation extraction (Zelenko et al., 2003; Zhang et al., 2006). However, to our knowledge, the application of such a technique to discourse relation recognition still remains unexplored.

Lin et al. (2009) has explored the 2-level production rules for discourse analysis. However, Figure 1 shows that only 2-level sub-tree structures (e.g. T_a - T_e) are covered in production rules. Other sub-trees beyond 2-level (e.g. T_f - T_j) are only captured in the tree kernel, which allows tree kernel to further leverage on information from higher dimension space for possible better discrimination. Especially, when there are enough training data, this is similar to the study

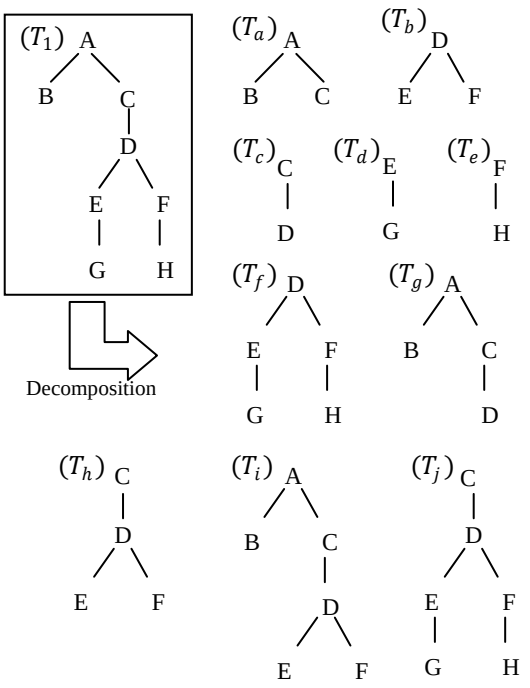

Figure 1. Different sub-tree sets for T_1 used by 2-level production rules and convolution tree kernel approaches. T_a-T_j and T_1 itself are covered by tree kernel, while only T_a-T_e are covered by production rules.

on language modeling that N-gram beyond unigram and bigram further improves the performance in large corpus.

Tense and Temporal Ordering Information. Linguistic studies (Webber, 1988) show that a tensed clause C_b provides two pieces of semantic information: (a) a description of an event (or situation) E_b; and (b) a particular configuration of the *point of event* (ET), the *point of reference* (RT) and the *point of speech* (ST). Both the characteristics of E_b and the configuration of ET, RT and ST are critical to interpret the relationship of event E_b with other events in the discourse model. Our observation on temporal ordering information is in line with the above, which is also incorporated in our discourse analyzer.

4 The Recognition Framework

In the learning framework, a training or testing instance is formed by a non-overlapping clause(s)/sentence(s) pair. Specifically, since *implicit* relations in PDTB are defined to be *local*, only clauses from adjacent sentences are paired for implicit cases. During training, for each discourse relation encountered, a positive instance is created by pairing the two arguments. Also a

set of negative instances is formed by paring each argument with neighboring non-argument clauses or sentences. Based on the training instances, a binary classifier is generated for each type using a particular learning algorithm. During resolution, (a) clauses within same sentence and sentences within three-sentence spans are paired to form an explicit testing instance; and (b) neighboring sentences within three-sentence spans are paired to form an implicit testing instance. The instance is presented to each explicit or implicit relation classifier which then returns a class label with a confidence value indicating the likelihood that the candidate pair holds a particular discourse relation. The relation with the highest confidence value will be assigned to the pair.

4.1 Base Features

In our system, the base features adopted include lexical pair, distance and attribution etc. as listed in Table 1. All these base features have been proved effective for discourse analysis in previous work.

Feature Names	Description
(F1)	cue phrase
(F2)	neighboring punctuation
(F3)	position of connective if presents
(F4)	extents of arguments
(F5)	relative order of arguments
(F6)	distance between arguments
(F7)	grammatical role of arguments
(F8)	lexical pairs
(F9)	attribution

Table 1. Base Feature Set

4.2 Support Vector Machine

In theory, any discriminative learning algorithm is applicable to learn the classifier for discourse analysis. In our study, we use Support Vector Machine (Vapnik, 1995) to allow the use of kernels to incorporate the structure feature.

Suppose the training set S consists of labeled vectors $\{(x_i, y_i)\}$, where x_i is the feature vector of a training instance and y_i is its class label. The classifier learned by SVM is:

$$f(x) = sgn\left(\sum_{i=1} y_i a_i x * x_i + b\right)$$

where a_i is the learned parameter for a feature vector x_i, and b is another parameter which can be derived from a_i . A testing instance x is classified as positive if $f(x) > 0$[1].

One advantage of SVM is that we can use tree kernel approach to capture syntactic parse tree information in a particular high-dimension space.

In the next section, we will discuss how to use kernel to incorporate the more complex structure feature.

5 Incorporating Structural Syntactic Information

A parse tree that covers both discourse arguments could provide us much syntactic information related to the pair. Both the syntactic flat path connecting connective and arguments and the 2-level production rules in the parse tree used in previous study can be directly described by the tree structure. Other syntactic knowledge that may be helpful for discourse resolution could also be implicitly represented in the tree. Therefore, by comparing the common sub-structures between two trees we can find out to which level two trees contain similar syntactic information, which can be done using a convolution tree kernel.

The value returned from the tree kernel reflects the similarity between two instances in syntax. Such syntactic similarity can be further combined with other flat linguistic features to compute the overall similarity between two instances through a composite kernel. And thus an SVM classifier can be learned and then used for recognition.

5.1 Structural Syntactic Feature

Parsing is a sentence level processing. However, in many cases two discourse arguments do not occur in the same sentence. To present their syntactic properties and relations in a single tree structure, we construct a syntax tree for each paragraph by attaching the parsing trees of all its sentences to an upper paragraph node. In this paper, we only consider discourse relations within 3 sentences, which only occur within each pa-

[1] In our task, the result of $f(x)$ is used as the confidence value of the candidate argument pair x to hold a particular discourse relation.

ragraph, thus paragraph parse trees are sufficient. Our 3-sentence spans cover 95% discourse relation cases in PDTB v2.0.

Having obtained the parse tree of a paragraph, we shall consider how to select the appropriate portion of the tree as the structured feature for a given instance. As each instance is related to two arguments, the structured feature at least should be able to cover both of these two arguments. Generally, the more substructure of the tree is included, the more syntactic information would be provided, but at the same time the more noisy information would likely be introduced. In our study, we examine three structured features that contain different substructures of the paragraph parse tree:

Min-Expansion This feature records the minimal structure covering both arguments and connective word in the parse tree. It only includes the nodes occurring in the shortest path connecting Arg1, Arg2 and connective, via the nearest commonly commanding node. For example, considering Example (5), Figure 2 illustrates the representation of the structured feature for this relation instance. Note that the two clauses underlined with dashed lines are *attributions* which are not part of the relation.

(5). Arg1. *Suppression of the book,* Judge Oakes observed, *would operate as a prior restraint and thus involve the First Amendment.*

Arg2. Moreover, and here Judge Oakes went to the heart of the question, "Responsible biographers and historians constantly use primary sources, letters, diaries and memoranda."

Simple-Expansion *Min-Expansion* could, to some degree, describe the syntactic relationships between the connective and arguments. However, the syntactic properties of the argument pair might not be captured, because the tree structure surrounding the argument is not taken into consideration. To incorporate such information, *Simple-Expansion* not only contains all the nodes in *Min-Expansion*, but also includes the first-level children of these nodes[2]. Figure 3 illustrates such a feature for Example (5). We can see that the nodes "PRN" in both sentences are included in the feature.

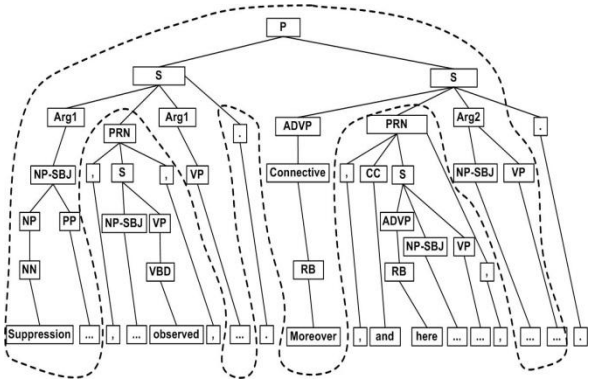

Figure 2. Min-Expansion tree built from golden standard parse tree for the explicit discourse relation in Example (5). Note that to distinguish from other words, we explicitly mark up in the structured feature the arguments and connective, by appending a string tag "Arg1", "Arg2" and "Connective" respectively.

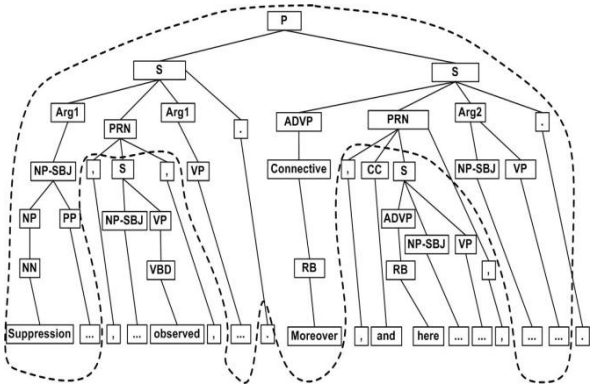

Figure 3. Simple-Expansion tree for the explicit discourse relation in Example (5).

Full-Expansion This feature focuses on the tree structure between two arguments. It not only includes all the nodes in *Simple-Expansion*, but also the nodes (beneath the nearest commanding parent) that cover the words between the two arguments. Such a feature keeps the most information related to the argument pair. Figure 4

[2] We will not expand the nodes denoting the sentences other than where the arguments occur.

shows the structure for feature *Full-Expansion* of Example (5). As illustrated, different from in *Simple-Expansion*, each sub-tree of "PRN" in each sentence is fully expanded and all its children nodes are included in *Full-Expansion*.

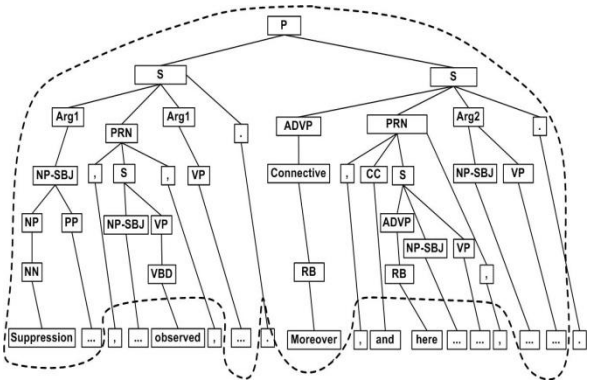

Figure 4. Full-Expansion tree for the explicit discourse relation in Example (5).

5.2 Convolution Parse Tree Kernel

Given the parse tree defined above, we use the same convolution tree kernel as described in (Collins and Duffy, 2002) and (Moschitti, 2004). In general, we can represent a parse tree T by a vector of integer counts of each sub-tree type (regardless of its ancestors):

$$\emptyset(T) = (\#of\ subtrees\ of\ type\ 1, \dots, \#\ of\ subtrees\ of type\ I, \dots, \#\ of\ subtrees\ of\ type\ n).$$

This results in a very high dimensionality since the number of different sub-trees is exponential in its size. Thus, it is computational infeasible to directly use the feature vector $\emptyset(T)$. To solve the computational issue, a tree kernel function is introduced to calculate the dot product between the above high dimensional vectors efficiently.

Given two tree segments T_1 and T_2, the tree kernel function is defined:

$$\begin{aligned} K(T_1, T_2) &= < \emptyset(T_1), \emptyset(T_2) > \\ &= \sum_i \emptyset(T_1)[i], \emptyset(T_2)[i] \\ &= \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} \sum_i I_i(n_1) * I_i(n_2) \end{aligned}$$

where N_1 and N_2 are the sets of all nodes in trees T_1 and T_2, respectively; and $I_i(n)$ is the indicator function that is 1 iff a subtree of type i occurs with root at node n or zero otherwise. (Collins and Duffy, 2002) shows that $K(T_1, T_2)$ is an instance of convolution kernels over tree structures, and can be computed in $O(|N_1|, |N_2|)$ by the following recursive definitions:

$$\Delta(n_1, n_2) = \sum_i I_i(n_1) * I_i(n_2)$$

(1) $\Delta(n_1, n_2) = 0$ if n_1 and n_2 do not have the same syntactic tag or their children are different;

(2) else if both n_1 and n_2 are pre-terminals (i.e. POS tags), $\Delta(n_1, n_2) = 1 \times \lambda$;

(3) else, $\Delta(n_1, n_2) =$

$$\lambda \prod_{j=1}^{nc(n_1)} (1 + \Delta(ch(n_1, j), ch(n_2, j))),$$

where $nc(n_1)$ is the number of the children of n_1, $ch(n, j)$ is the j^{th} child of node n and λ ($0 < \lambda < 1$) is the decay factor in order to make the kernel value less variable with respect to the sub-tree sizes. In addition, the recursive rule (3) holds because given two nodes with the same children, one can construct common sub-trees using these children and common sub-trees of further offspring.

The parse tree kernel counts the number of common sub-trees as the syntactic similarity measure between two instances. The time complexity for computing this kernel is $O(|N_1| \cdot |N_2|)$.

5.3 Composite Tree Kernel

Besides the above convolution parse tree kernel $\widehat{K}_{tree}(x_1, x_2) = K(T_1, T_2)$ defined to capture the syntactic information between two instances x_1 and x_2, we also use another kernel $\widehat{K}_{flat}$ to capture other flat features, such as base features (described in Table 1) and temporal ordering information (described in Section 6). In our study, the composite kernel is defined in the following way:

$$\begin{aligned} \widehat{K}_1(x_1, x_2) = \alpha \cdot \widehat{K}_{flat}(x_1, x_2) + \\ (1 - \alpha) \cdot \widehat{K}_{tree}(x_1, x_2). \end{aligned}$$

Here, $\widehat{K}(\cdot, \cdot)$ can be normalized by $\widehat{K}(y, z) = K(y, z)/\sqrt{K(y, y) \cdot K(z, z)}$ and α is the coefficient.

6 Using Temporal Ordering Information

In our discourse analyzer, we also add in temporal information to be used as features to predict discourse relations. This is because both our observations and some linguistic studies (Webber, 1988) show that temporal ordering information including tense, aspectual and event orders between two arguments may constrain the discourse relation type. For example, the connective

word is the same in both Example (6) and (7), but the tense shift from progressive form in clause 6.a to simple past form in clause 6.b, indicating that the *twisting* occurred during the state of *running the marathon*, usually signals a *temporal* discourse relation; while in Example (7), both clauses are in past tense and it is marked as a *Causal* relation.

(6). a. Yesterday Holly was running a marathon
b. ***when*** she twisted her ankle.

(7). a. Use of dispersants was approved
b. ***when*** a test on the third day showed some positive results.

Inspired by the linguistic model from Webber (1988) as described in Section 3, we explore the temporal order of events in two adjacent sentences for discourse relation interpretation. Here event is represented by the *head of verb*, and the temporal order refers to the *logical* occurrence (i.e. before/at/after) between events. For instance, the event ordering in Example (8) can be interpreted as:

$$Event(broken) \prec_{before} Event(went) .$$

8. a. John went to the hospital.
b. He had broken his ankle on a patch of ice.

We notice that the feasible temporal order of events differs for different discourse relations. For example, in *causal* relations, *cause* event usually happens before *effect* event, i.e.

$$Event(cause) \prec_{before} Event(effect).$$

So it is possible to infer a *causal* relation in Example (8) if and only if 8.b is taken to be the *cause* event and 8.a is taken to be the effect event. That is, 8.b is taken as happening prior to *his going into hospital*.

In our experiments, we use the TARSQI[3] system to identify event, analyze tense and aspectual information, and label the temporal order of events. Then the tense and temporal ordering information is extracted as features for discourse relation recognition.

[3] http://www.isi.edu/tarsqi/

7 Experiments and Results

In this section we provide the results of a set of experiments focused on the task of simultaneous discourse identification and classification.

7.1 Experimental Settings

We experiment on PDTB v2.0 corpus. Besides four top-level discourse relations, we also consider *Entity* and *No* relations described in Section 2. We directly use the golden standard parse trees in Penn TreeBank. We employ an SVM coreference resolver trained and tested on ACE 2005 with 79.5% Precision, 66.7% Recall and 72.5% F_1 to label coreference mentions of the same named entity in an article. For learning, we use the binary SVMLight developed by (Joachims, 1998) and Tree Kernel Toolkits developed by (Moschitti, 2004). All classifiers are trained with default learning parameters.

The performance is evaluated using *Accuracy* which is calculated as follow:

$$Accuracy = \frac{TruePositive + TrueNegative}{All}$$

Sections 2-22 are used for training and Sections 23-24 for testing. In this paper, we only consider any non-overlapping clauses/sentences pair in 3-sentence spans. For training, there were 14812, 12843 and 4410 instances for *Explicit*, *Implicit* and *Entity+No* relations respectively; while for testing, the number was 1489, 1167 and 380.

7.2 System with Structural Kernel

Table 2 lists the performance of simultaneous identification and classification on level-1 discourse senses. In the first row, only base features described in Section 4 are used. In the second row, we test Ben and James (2007)'s algorithm which uses heuristically defined syntactic paths and acts as a good baseline to compare with our learned-based approach using the structured information. The last three rows of Table 2 reports the results combining base features with three syntactic structured features (i.e. *Min-Expansion*, *Simple-Expansion* and *Full-Expansion*) described in Section 5.

We can see that all our tree kernels outperform the manually constructed flat path feature in all three groups including *Explicit* only, *Implicit* only and *All* relations, with the accuracy increasing by 1.8%, 6.7% and 3.1% respectively. Especially, it shows that structural syntactic information is more helpful for *Implicit* cases which is generally much harder than *Explicit* cases. We

Features	Accuracy		
	Explicit	Implicit	All
Base Features	67.1	29	48.6
Base + Manually selected flat path features	70.3	32	52.6
Base + Tree kernel (Min-Expansion)	71.9	38.6	55.6
Base + Tree kernel (Simple-Expansion)	**72.1**	**38.7**	**55.7**
Base + Tree kernel (Full-Expansion)	71.8	38.4	55.4

Table 2. Results of the syntactic structured kernels on level-1 discourse relation recognition.

conduct chi square statistical significance test on *All* relations between flat path approach and *Simple-Expansion* approach, which shows the performance improvements are statistical significant ($\rho < 0.05$) through incorporating tree kernel. This proves that structural syntactic information has good predication power for discourse analysis in both explicit and implicit relations. We also observe that among the three syntactic structured features, *Min-Expansion* and *Simple-Expansion* achieve similar performances which are better than the result for *Full-Expansion*. This may be due to that most significant information is with the arguments and the shortest path connecting connectives and arguments. However, *Full-Expansion* that includes more information in other branches may introduce too many details which are rather tangential to discourse recognition. Our subsequent reports will focus on *Simple-Expansion*, unless otherwise specified.

As described in Section 5, to compute the structural information, parse trees for different sentences are connected to form a large tree for a paragraph. It would be interesting to find how the structured information works for discourse relations whose arguments reside in different sentences. For this purpose, we test the accuracy for discourse relations with the two arguments occurring in the same sentence, one-sentence apart, and two-sentence apart. Table 3 compares the learning systems with/without the structured feature present. From the table, for all three cases, the accuracies drop with the increase of the distances between the two arguments. However, adding the structured information would bring consistent improvement against the baselines regardless of the number of sentence distance. This observation suggests that the structured syntactic information is more helpful for inter-sentential discourse analysis.

We also concern about how the structured information works for identification and classification respectively. Table 4 lists the results for the two sub-tasks. As shown, with the structured information incorporated, the system (Base + Tree Kernel) can boost the performance of the two baselines (Base Features in the first row andBase + Manually selected paths in the second row), for both identification and classification respectively. We also observe that the structural syntactic information is more helpful for classification task which is generally harder than identification. This is in line with the intuition that classification is generally a much harder task. We find that due to the weak modeling of *Entity* relations, many *Entity* relations which are non-discourse relation instances are mis-identified as implicit *Expansion* relations. Nevertheless, it clearly directs our future work.

Sentence Distance	0 (959)	1 (1746)	2 (331)
Base Features	52	49.2	35.5
Base + Manually selected flat path features	56.7	52	43.8
Base + Tree Kernel	58.3	55.6	49.7

Table 3. Results of the syntactic structured kernel for discourse relations recognition with arguments in different sentences apart.

Tasks	Identification	Classification
Base Features	58.6	50.5
Base + Manually selected flat path features	59.7	52.6
Base + Tree Kernel	63.3	59.3

Table 4. Results of the syntactic structured kernel for simultaneous discourse identification and classification subtasks.

7.3 System with Temporal Ordering Information

To examine the effectiveness of our temporal ordering information, we perform experiments

on simultaneous identification and classification of level-1 discourse relations to compare with using only base feature set as baseline. The results are shown in Table 5. We observe that the use of temporal ordering information increases the accuracy by 3%, 3.6% and 3.2% for *Explicit*, *Implicit* and *All* groups respectively. We conduct chi square statistical significant test on *All* relations, which shows the performance improvement is statistical significant ($\rho < 0.05$). It indicates that temporal ordering information can constrain the discourse relation types inferred within a clause(s)/sentence(s) pair for both explicit and implicit relations.

Features	Accuracy		
	Explicit	Implicit	All
Base Features	67.1	29	48.6
Base + Temporal Ordering Information	70.1	32.6	51.8

Table 5. Results of tense and temporal order information on level-1 discourse relations.

We observe that although temporal ordering information is useful in both explicit and implicit relation recognition, the contributions of the specific information are quite different for the two cases. In our experiments, we use tense and aspectual information for explicit relations, while event ordering information is used for implicit relations. The reason is explicit connective itself provides a strong hint for explicit relation, so tense and aspectual analysis which yields a reliable result can provide additional constraints, thus can help explicit relation recognition. However, event ordering which would inevitably involve more noises will adversely affect the explicit relation recognition performance. On the other hand, for implicit relations with no explicit connective words, tense and aspectual information alone is not enough for discourse analysis. Event ordering can provide more necessary information to further constrain the inferred relations.

7.4 Overall Results

We also evaluate our model which combines base features, tree kernel and tense/temporal ordering information together on *Explicit*, *Implicit* and *All* Relations respectively. The overall results are shown in Table 6.

Relations	Accuracy
Explicit	74.2
Implicit	40.0
All	57.3

Table 6. Overall results for combined model (Base + Tree Kernel + Tense/Temporal).

8 Conclusions and Future Works

The purpose of this paper is to explore how to make use of the structural syntactic knowledge to do discourse relation recognition. In previous work, syntactic information from parse trees is represented as a set of heuristically selected flat paths or 2-level production rules. However, the features defined this way may not necessarily capture all useful syntactic information provided by the parse trees for discourse analysis. In the paper, we propose a kernel-based method to incorporate the structural information embedded in parse trees. Specifically, we directly utilize the syntactic parse tree as a structure feature, and then apply kernels to such a feature, together with other normal features. The experimental results on PDTB v2.0 show that our kernel-based approach is able to give statistical significant improvement over flat syntactic path method. In addition, we also propose to incorporate temporal ordering information to constrain the interpretation of discourse relations, which also demonstrate statistical significant improvements for discourse relation recognition, both explicit and implicit.

In future, we plan to model *Entity* relations which constitute 24% of *Implicit+Entity+No* relation cases, thus to improve the accuracy of relation detection.

Reference

Ben W. and James P. 2007. *Automatically Identifying the Arguments of Discourse Connectives*. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 92-101.

Culotta A. and Sorensen J. 2004. *Dependency Tree Kernel for Relation Extraction*. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2004), pages 423-429.

Collins M. and Duffy N. 2001. *New Ranking Algorithms for Parsing and Tagging: Kernels over Dis-*

crete Structures and the Voted Perceptron. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002), pages 263-270.

Collins M. and Duffy N. 2002. *Convolution Kernels for Natural Language.* NIPS-2001.

Haussler D. 1999. *Convolution Kernels on Discrete Structures*. Technical Report UCS-CRL-99-10, University of California, Santa Cruz.

Joachims T. 1999. *Making Large-scale SVM Learning Practical*. In Advances in Kernel Methods – Support Vector Learning. MIT Press.

Knott, A., Oberlander, J., O'Donnel, M., and Mellish, C. 2001. *Beyond elaboration: the interaction of relations and focus in coherent text.* In T. Sanders, J. Schilperoord, and W. Spooren, editors, Text Representation: Linguistic and Psycholinguistics Aspects, pages 181-196. Benjamins, Amsterdam.

Lee A., Prasad R., Joshi A., Dinesh N. and Webber B. 2006. *Complexity of dependencies in discourse: are dependencies in discourse more complex than in syntax?* In Proceedings of the 5th International Workshop on Treebanks and Linguistic Theories. Prague, Czech Republic, December.

Lin Z., Kan M. and Ng H. 2009. *Recognizing Implicit Discourse Relations in the Penn Discourse Treebank*. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009), Singapore, August.

Marcu D. and Echihabi A. 2002. *An Unsupervised Approach to Recognizing Discourse Relations*. In Proceedings of the 40th Annual Meeting of ACL, pages 368-375.

Moschitti A. 2004. *A Study on Convolution Kernels for Shallow Semantic Parsing*. In Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics (ACL 2004), pages 335-342.

Pettibone J. and Pon-Barry H. 2003. *A Maximum Entropy Approach to Recognizing Discourse Relations in Spoken Language*. Working Paper. The Stanford Natural Language Processing Group, June 6.

Pitler E., Louis A. and Nenkova A. 2009. *Automatic Sense Predication for Implicit Discourse Relations in Text*. In Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009).

Prasad R., Dinesh N., Lee A., Miltsakaki E., Robaldo L., Joshi A. and Webber B. 2008. T*he Penn Discourse TreeBank 2.0*. In Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008).

Saito M., Yamamoto K. and Sekine S. 2006. *Using phrasal patterns to identify discourse relations*. In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2006), pages 133–136, New York, USA.

Vapnik V. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

Webber Bonnie. 1988. *Tense as Discourse Anaphor*. Computational Linguistics, 14:61–73.

Zelenko D., Aone C. and Richardella A. 2003. *Kernel Methods for Relation Extraction*. Journal of Machine Learning Research, 3(6):1083-1106.

Zhang M., Zhang J. and Su J. *Exploring Syntactic Features for Relation Extraction using a Convolution Tree Kernel*. In Proceedings of the Human Language Technology conference - North American chapter of the Association for Computational Linguistics annual meeting (HLT-NAACL 2006), New York, USA.