Detection, Disambiguation and Argument Identification of Discourse Connectives in Chinese Discourse Parsing

Yong-Siang Shih and Hsin-Hsi Chen
Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan
{r02922036,hhchen}@ntu.edu.tw

Abstract

In this paper, we investigate four important issues together for explicit discourse relation labelling in Chinese texts: (1) discourse connective extraction, (2) linking ambiguity resolution, (3) relation type disambiguation, and (4) argument boundary identification. In a pipelined Chinese discourse parser, we identify potential connective candidates by string matching, eliminate non-discourse usages from them with a binary classifier, resolve linking ambiguities among connective components by ranking, disambiguate relation types by a multiway classifier, and determine the argument boundaries by conditional random fields. The experiments on Chinese Discourse Treebank show that the F1 scores of 0.7506, 0.7693, 0.7458, and 0.3134 are achieved for discourse usage disambiguation, linking disambiguation, relation type disambiguation, and argument boundary identification, respectively, in a pipelined Chinese discourse parser.

1 Introduction

Discourse relations represent how discourse units logically connect with each other. A discourse connective explicitly signals the presence of a discourse relation, and therefore it is an important clue for discourse analysis. There are several challenges in Chinese discourse parsing.

Firstly, there are more discourse connectives in Chinese than in English and their parts of speech have more varieties (Huang et al., 2014). Therefore, it is likely to encounter words that have the same surface forms as real connectives but do not function as discourse connectives.

Secondly, many Chinese connectives are parallel connectives that have multiple discontinuous components (Zhou and Xue, 2012). Each connective can be composed of one or more *connective components*. For example, "雖然-但" (although-but) consists of two connective components: "雖然" (although) and "但" (but). When multiple connectives are present in a paragraph, their components often link with each other in multiple possible ways. (S1) is an example. There are five possible connective candidates, including (1) "除了...還" (in addition to ... also), (2) "還...也" (also ... also), (3) "除了" (in addition to), (4) "還" (also), and (5) "也" (also). Figure 1 illustrates the ambiguous linking. Only candidates (1) and (5) are correct. Moreover, when spurious component candidates exist in a paragraph, they form many more spurious connective candidates by linking together in different ways. Finding the correct linking between correct components is useful for discourse analysis because they provide clues to determine the positions of the relations.

(S1) 除了投資環境優越，還在於這些企業所具有的產品優勢。(...) 對上海的產業優化也有很大的帶動作用。(In addition to superior investment environment, it's also due to the product advantages possessed by the enterprise. (...) It also has great effect in promoting the optimization of industries in Shanghai.)

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 1891–1902, Osaka, Japan, December 11-17 2016.

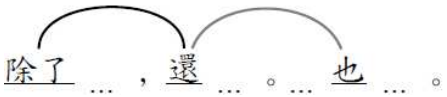

Figure 1: Ambiguous linking between connective components.

Thirdly, the sentence structures in Chinese texts are not clearly defined. Thus it is more challenging to detect the arguments for a given relation. Here, arguments of a discourse relation are the discourse units it involves. Since the relations form a hierarchical discourse structure, the arguments for relations higher in the hierarchy or lower in the hierarchy could span over ranges of various lengths.

In this paper, we aim at investigating these unique challenges at the same time. The goal of this research is to build an end-to-end system to analyse the explicit discourse relations in Chinese texts. In particular, we deal with four tasks together: (1) extraction of explicit discourse connectives, (2) linking resolution between the component candidates, (3) classification of the relation type for each discourse connective, and (4) extraction of the discourse arguments of a connective.

This paper is organized as follows. Section 2 surveys the related work. Section 3 describes the datasets used in this study. Section 4 presents our Chinese discourse parser. Section 5 shows and discusses the experimental results. Section 6 concludes the remarks.

2 Related Work

Rhetorical Structure Theory Discourse Treebank (RST-DT) (Carlson et al., 2001) and the Penn Discourse Treebank 2.0 (PDTB2) (Prasad et al., 2008) are two popular English discourse corpora for discourse analysis. Many groups have investigated different subtasks of English discourse parsing on PDTB2, including discourse connective identification, relation type disambiguation (Pitler and Nenkova, 2009; Wellner, 2009; Faiz and Mercer, 2013), and argument extraction (Wellner and Pustejovsky, 2007; Elwell and Baldridge, 2008; Ghosh et al., 2011; Ghosh et al., 2012; Kong et al., 2014). Lin et al. (2014) build an end-to-end discourse parser. As RST-DT provides hierarchical discourse structure annotations, there are also many attempts to construct the discourse structures automatically for sentences (Sporleder and Lapata, 2005; Fisher and Roark, 2007; Joty et al., 2012) and documents (Hernault et al., 2010; Feng and Hirst, 2012; Joty et al., 2013; Li, Li et al., 2014; Ji and Eisenstein, 2014).

Comparatively, there have been few large-scale Chinese discourse corpora until recently (Zhou and Xue, 2012; Zhou and Xue, 2015; Zhang et al., 2014; Li, Feng et al., 2014). Due to limited resource, early studies often used self-constructed corpora that made it difficult to compare between different works. T'sou et al. (1999), T'sou et al. (2000) and Chan et al. (2000) investigated connective detection in Chinese texts as a part of a tagging system. Hu et al. (2009) developed an automatic system to extract connective components from sentences. They used a rule-based method and found that the performance was sensitive to the connective lexicon. They improved their performance by removing words commonly used in non-discourse contexts. Zhou et al. (2012) and Li, Carpuat et al. (2014) employed cross-lingual information to deal with discourse usage ambiguity. Li, Carpuat et al. (2014) used 5-way classification to classify a connective between four relation types and non-discourse usage. Li et al. (2015) used CDTB to investigate detection and classification of connective components. They used maximum entropy and decision tree algorithms with various syntactic features. Chen et al. (2016) investigated fine-grained Chinese discourse relation labelling. Hu et al. (2011) dealt with linking ambiguity. However, they only focused on intra-sentential relations in sentences that have multiple clauses and assumed all connective components in the sentence have already been correctly identified.

As researchers start to focus on higher level problems for linguistic analysis, interest in discourse parsing also grows. The CoNLL-2015 Shared Task (Xue et al., 2015) and the CoNLL-2016 Shared Task (Xue et al., 2016) both focus on shallow discourse parsing. In particular, the CoNLL-2016 Shared Task features Chinese discourse parsing with Chinese Discourse Treebank 0.5 (Zhou and Xue, 2015), a PDTB-style annotated corpus.

3 Datasets

In this section, we briefly introduce the datasets used in this study and provide some statistics.

3.1 Chinese Discourse Treebank (CDTB)

In Chinese Discourse Treebank (CDTB) (Li, Feng et al., 2014), 500 documents selected from the Chinese Treebank (CTB) (Xue et al., 2005) were annotated. Totally, CDTB contains 2,342 paragraphs. Each paragraph was segmented into elementary discourse units (EDUs), and each paragraph is represented as a discourse tree as shown in Figure 2. Each relation is represented by an internal node, while each EDU is represented by a leaf node. The explicit and implicit relations between different spans of EDUs were annotated. In addition, discourse connectives for each relation were annotated. The exact positions of the arguments for a relation is heavily influenced by the complete discourse tree, and can range over multiple sentences in a paragraph.

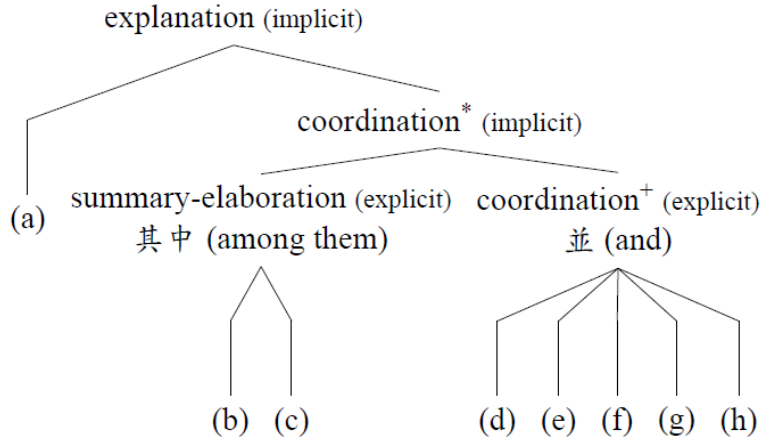

Figure 2: A discourse tree.

In CDTB, total 7,310 relations are annotated, and 1,814 of them are explicit. The set of discourse relation types is organized in a three-level hierarchy. In this paper, we only focus on the four top-level relation types, i.e., causality, coordination, transition, and explanation.

Some errors including duplicate annotations and erroneous positions are found in the corpus. After manual correction for explicit relation annotations, there are 1,813 explicit relations, each of which consists of exactly one connective instance. These 1,813 connectives are composed of 2,131 connective component instances.

The length distribution for the annotated connectives is shown in Table 1. The distribution is imbalance. There are totally 274 classes of connectives, but 147 of them appear only once.[1] Since some connectives share the same components, there are only 227 classes of connective components. Most of the connective classes only have one unique top-level relation type.

#Components	1	2	3	4	6	7
#Connective Classes	143	108	15	6	1	1
#Instances	1,544	235	24	8	1	1

Table 1: Lengths of connectives.

Table 2 shows the number of arguments for explicit relations. Most relations only have two arguments, but there are relations that have as many as 7 arguments. The number of arguments does not always match with the number of connective components. For example, single connective "并" (and) can have as many as 5 arguments because it connects multiple parallel segments together.

#Arguments	2	3	4	5	6	7
#Instances	1,688	85	33	4	2	1

Table 2: Number of arguments for each explicit relation.

Totally there are 3,802 arguments for explicit relations. Depending on the position the relation resides in the discourse tree, the length of an argument could vary greatly, but most of the arguments for

[1] These numbers are computed by their surface forms, i.e., instances of the same connective class may have different relation types.

explicit relation are composed of only one EDU. On average, each argument is composed of 1.6 EDUs, and the longest argument is composed of 20 EDUs.

3.2 NTU PN-Gram Corpus

Recently, efficient methods to learn word embeddings have been developed. In this paper, we investigate whether such word vectors are useful for dealing with discourse issues. NTU PN-Gram Corpus released by Yu et al., (2012), which was constructed by POS-tagging the Chinese texts extracted from the ClueWeb09 dataset (Callan et al., 2009). It has 21,217,147 unique sentences, containing 326,996,602 tokens. We used this corpus to create 400-dimensional embeddings using GloVe tool (Pennington et al., 2014) and word2vec tool (Mikolov et al., 2013a; Mikolov et al., 2013b) with skip-gram and continuous bag-of-words models.

3.3 Connective Component Dictionary

Discourse connectives serve as linking elements that connect discourse units. There are three kinds of linking directions (Li and Thompson, 1989): (1) forward-linking, (2) backward-linking, and (3) couple-linking. Such linking directions could be useful for identifying the positions of arguments for a given connective. We used the connective linking dictionary collected by Huang et al. (2014) as features for argument boundary identification. Totally, it contains 301 distinct connective components that are annotated with linking directions.

4 Chinese Discourse Parser

There are five modules in the proposed pipelined system. Each paragraph in CDTB is processed by the following modules: (1) identify connective candidates, (2) eliminate non-discourse usages from connective candidates, (3) resolve linking ambiguities, (4) disambiguate relation types, and (5) extract arguments.

4.1 Connective Candidate Extraction

We use string matching with the connective lexicon collected from CDTB to extract all possible instances. Directly matching with raw text would yield 12,498 candidate components[2] because many characters used for connectives appear in other unrelated words. Therefore, Stanford Chinese segmenter (Chang et al., 2008) is employed to segment paragraphs into tokens. Only the components composed of complete tokens are extracted.

Total 7,649 component candidates which recover 2,068 of 2,131 annotated components are extracted. These candidates form 7,976 connective candidates which recover 1,755 of 1,813 annotated connectives. While some correct instances are not extracted due to segmentation errors, it reduces the number of spurious candidates substantially while maintaining high recall.

4.2 Discourse Usage Disambiguation

A logistic regression classifier is trained to eliminate spurious connective candidates. The features are listed below:

P&N: We used a subset of the features selected from Pitler and Nenkova (2009). It includes four binary features for each connective component: (1) the highest category that dominates exactly the component itself, which is called self-category, (2) the parent, (3) the left-sibling, and (4) the right-sibling of the self-category. Null features are set when no such nodes exist. For example, in Figure 3, there is no node that dominates exactly the component "卻是" (but). For multi-component connectives, the union of the features is used. We have also experimented with the full feature set, but the performance does not increase.

CONNECTIVE: The string of the connective.

[2] Candidate components that could not form complete connectives are already eliminated. If we simply match with component lexicon without checking whether they could form connectives, 24,539 candidate components would be detected.

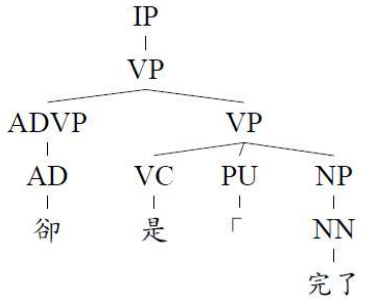

Figure 3: A sub-parsing tree for 卻是.

POS: The feature set contains: (1) POS tags for all tokens that constitute the connective component, (2) POS tag of the token to the left of the component, and (3) POS tag of the token to the right of the component. For multi-component connectives, the union of the features is used.

NUM: The feature set contains a one-hot encoded feature–the number of components. In addition, there are seven numerical features: (1) the number of overlapped connective candidates, (2) the number of connective candidates that enclose any components of the current connective, (3) the distance between the leftmost and the rightmost tokens of the connective measured by tokens, (4) the geometric mean of distances between all neighbouring connective components for the current connective candidate, (5) the distances from the leftmost component to a separating element including "!?:;, ◦ " or the paragraph boundary on the left, (6) the distance from the rightmost component to a separating element on the right, and (7) the minimum distance from any separating element to any connective component. We normalize the numerical features by scaling each to zero mean and unit variance.

VECTOR: This feature set is built using word embeddings. The vectors are used to construct three features for each connective component: (1) the mean of the vectors representing each token that constitutes the connective component, (2) the vector for the token to the left, and (3) the vector for the token to the right. Zero-valued vector is used when the vector does not exist. In total, it is a 1,200-dimensional vector for each component when the 400-dimensional embeddings are used. For multi-component connectives, we averaged the vectors.

4.3 Connective Linking Disambiguation

If we can classify discourse usage perfectly, we would have already solved the linking ambiguities because only the correct connectives remain. Due to imperfect classification, there may still exist some overlapped candidates. Here, we propose a greedy algorithm to resolve linking ambiguities among a set of connective candidates as outlined in Algorithm 1. The algorithm filters the candidate set C and produces a result set A that contains only non-overlapped connective candidates. All connective candidates are ranked under some criteria and the one with the highest priority is greedily accepted. We will use different ranking criteria to evaluate our models, including (1) Score: the probability obtained by logistic regression as described in Section 4.2, (2) Length: the number of components each connective candidate has, the larger the better, and (3) Position. Position is used mainly as a tiebreaker. In particular, we accept the left-most candidate first.

Algorithm 1. Linking Resolution Algorithm

Input C: A set of connective candidates
Output A: A set of accepted connectives

1. $A \leftarrow \{\}$
2. Rank all connective candidates in C
3. **while** C is not empty **do**
4. let c_i be the connective candidate that has the highest priority
5. $C \leftarrow C - \{c_i\}$
6. $A \leftarrow A \cup \{c_i\}$
7. Remove all connective candidates $c_j \in C$ that overlap with c_i
8. **end while**
9. **return** A

4.4 Discourse Relation Type Disambiguation

We use a logistic regression classifier to investigate whether the features we discussed in Section 4.2 are useful for relation type disambiguation.

4.5 Connective Argument Extraction

We formulate argument extraction as a sequence labelling problem. As we know the arguments span over a continuous interval, we use four labels for the EDUs: *before*, *start*, *inside*, and *after*. Each argument is represented by a *start* followed by zero or more *inside*s. Figure 2 is a discourse tree, where the explanation relation has two arguments (a) and (b-h), while the coordination$^+$ relation has five arguments (d), (e), (f), (g), and (h). Figure 4 shows the arguments and the corresponding labels for summary elaboration relation shown in Figure 2.

As our goal is to extract the arguments, only the argument boundaries must be determined. Although correct EDU segmentation is unavailable to our system because we attempt to extract arguments from raw texts, the EDU boundaries in CDTB only occur with certain punctuation symbols that separate phrases and sentences. Thus, we segment a paragraph by these symbols, and solve the sequence labelling problem on these segments instead of EDUs.

We use Conditional Random Fields (CRFs) to deal with the sequence labelling problem. CRFsuite (Okazaki, 2007) is adopted along with its default parameters. When training, each explicit relation with its corresponding labelling is used as a training case. When testing, CRFs are used to label the segments for each connective we extracted. Therefore, the same segments for a paragraph are labelled independently for each explicit relation inside the paragraph. The resulting argument boundaries are used to extract the connective's arguments.

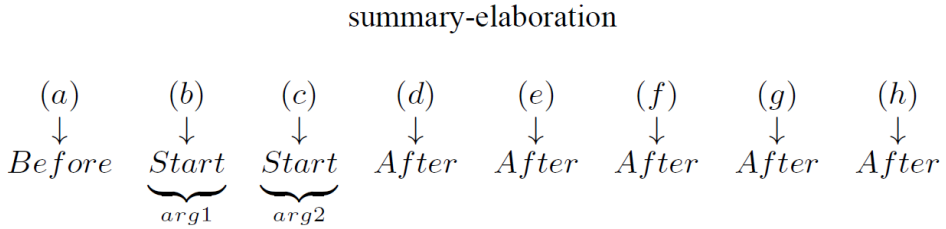

Figure 4: Sequence labelling for relation arguments identification.

The features, which are determined by the current connective being considered and the segment, are shown as follows.

CONTEXT: The concatenation of the self-category and the categories of the parent, the left-sibling, and the right-sibling is used as a binary feature as done by Kong et al. (2014).

PATH: The feature set is similar to the CON-NT-Path features of Kong et al. (2014). We use the path from the self-category of each connective component to the self-category of the segment as the feature.

POS: The POS tags for all tokens in the segment.

SUBJ: The SUBJ feature is set if a segment contains a subject.

ENDCHAR: The last token in the segment, i.e., the symbol that separates the current segment from the next one.

COMPONENT: The feature set contains the information about connective components: (1) whether the segment has a connective component, (2) the string of the component if it exists, (3) whether there exists a component at the beginning of the segment, (4) whether there exists a component at the end of the segment, (5) whether the segment contains only a component and the separating symbol, (6) whether the segment is before all connective components, (7) the distance to the first segment that contains a component as a binary feature, (8) whether the segment is after all connective components, and (9) the distance to the last segment that contains a connective component as a binary feature.

LINK: The feature set contains the linking directions a connective component could be used if it exists in the given segment. A connective component dictionary is consulted.

CONNECTIVE: This feature set contains connective related information, including the string of the connective and the number of connective components it has.

5 Results and Discussions

In the experimental setups, we first evaluate the performance of each individual disambiguation task and then examine the propagation effects in the pipelined system.

5.1 Discourse Usage Disambiguation

We evaluate our models using 10-fold cross-validation. The 2,342 paragraphs are divided into 10 splits while keeping the distribution for the number of explicit relations in each paragraph roughly equal. The averaged precision, recall, and F1 scores for the positive connective instances are reported. As there are many spurious connectives, we balance the training set by oversampling the correct instances for three times, but keep the original distribution when evaluating on test set.

For statistical significance, we use Wilcoxon signed-ranks test (Wilcoxon, 1945) as suggested by Demšar (2006) at confidence level 95%. For each experiment, we select the best model (denoted by bold), and * is used to denote the scores that are significantly different.

We firstly investigate the performance between different word embeddings as shown in Table 3. The vectors constructed by skip-gram model are the most useful. We will use them in the remaining experiments. Table 4 shows the results for all features. The best results are obtained with ALL-SKIP-GRAPM in discourse usage disambiguation. We also experiment with different learning models including Naive Bayes, SVM, decision trees, and random forest. Logistic Regression performs the best. For all models, we use default parameters provided by scikit-learn without tuning, i.e., C=1.0, penalty=l2 for Logistic Regression.

Features	Precision	Recall	F1 Score
CBOW	0.5808	0.7625	0.6593*
SKIP-GRAM	0.6013	0.8068	**0.6887**
GLOVE	0.5980	0.7996	0.6840
ALL	0.5837	0.7439	0.6539*

Table 3: Performance of discourse usage disambiguation using different word embeddings.

Features	Precision	Recall	F1 Score
P&N	0.4239	0.8409	0.5634*
CONNECTIVE	0.5205	0.8620	0.6487*
POS	0.5426	0.7805	0.6399*
NUM	0.4298	0.8456	0.5696*
SKIP-GRAM	0.6013	0.8068	0.6887*
ALL-P&N	0.6547	0.8186	0.7273*
ALL-POS	0.6576	0.8222	0.7305*
ALL-NUM	0.6357	0.8160	0.7144*
ALL-SKIP-GRAM	0.6503	0.8882	**0.7506**
ALL	0.6682	0.8203	0.7363*

Table 4: Performance of discourse usage disambiguation using different features.

5.2 Connective Linking Disambiguation

To evaluate linking disambiguation individually, we first assume all correct connective components are already known. We use the 10-fold for paragraphs specified in Section 5.1 to evaluate our model using each connective as an instance. The results are reported in Table 5. We evaluate different ranking criteria and the combination. A baseline model that simply ranks the candidates by their positions is also reported. We find that the ambiguity among the components is low. The baseline model already achieves an F1 score of 0.8797. In fact, only 472 out of 2,131 components are involved in more than one connective candidate. Length is relatively weaker than Score reported by the logistic regression model. Moreover, we also evaluate linking disambiguation within the pipelined system. The results are shown in Table 6. Integrating both Score and Length criteria performs the best. The comparison shows that most of the linking ambiguity is caused by spurious linking with spurious connective component candidates.

Ranking Criteria	Precision	Recall	F1 Score
Baseline	0.8528	0.9084	0.8797*
Score	0.9770	0.9796	**0.9783**
Length	0.9760	0.9604	0.9681*
Length+Score	0.9793	0.9636	0.9714*

Table 5: Performance of linking disambiguation with known connective components.

Ranking Criteria	Precision	Recall	F1 Score
Baseline	0.6696	0.7919	0.7254*
Score	0.7024	0.8222	0.7573*
Length	0.7099	0.8238	0.7624*
Length+Score	0.7165	0.8310	**0.7693**

Table 6: Performance of linking disambiguation in the pipelined system.

Methods	Precision	Recall	F1 Score
w/o Linking resolution	0.7399	0.8680	0.7985
Length+Score	0.7493	0.8585	**0.7999**
Li et al. (2015) ME	0.7880	0.6180	0.6920
Li et al. (2015) DT	0.5680	0.4960	0.5230

Table 7: Performance of discourse connective disambiguation on the component level.

In the above evaluation, we consider a connective instance as an evaluation unit. To compare with the related work we also take a connective component as an evaluation unit. Table 7 summarizes the results of discourse connective disambiguation on this level. The first model is for discourse usage disambiguation without linking disambiguation. The second model eliminates additional candidates by resolving linking ambiguity. The experimental results of Li et al. (2015) are listed for reference because they use the same dataset as us. The best results for Maximum Entropy (ME) and Decision Tree (DT) classifiers with automatic parsing tree features are selected. Although linking disambiguation has small effect for identifying individual components, it effectively improves the performance of connective extraction as shown in Table 6. The result is also important for argument extraction, because the positions of connective components provide clues for the positions of the arguments of an explicit relation.

5.3 Discourse Relation Type Disambiguation

At first, we evaluate the relation type disambiguation by assuming connectives are known. We use 10-fold cross-validation with the 1,813 explicit connectives to evaluate our model. We keep the distribution for the relation types roughly equal for each fold. While the NUM features have some discriminative power for discourse usage disambiguation, it does not help for relation disambiguation. When used independently, the performance is the same as always predicting the major category. On the other hand, the string of the connective provides strong clues for the relation type. When used individually, it already achieves a micro average F1 of 0.9308. In addition, the SKIP-GRAM feature is also useful for this task, achieving a micro average F1 of 0.9473. In Table 8, we show the performance for different relation types using ALL-NUM as features. We can find that the number of instances affects the performance of the learning model. The lesser the instances, the worse the performance. To compare with Li et al. (2015), we also evaluate the results on component level. Table 9 shows that we also achieve better performance on relation type classification.

Relation Type	Precision	Recall	F1 Score	#instances
causality	0.9634	0.9504	0.9561	465
coordination	0.9575	0.9723	0.9645	974
transition	0.9372	0.9131	0.9234	173
explanation	0.9754	0.9450	0.9588	201
macro average	0.9584	0.9452	0.9507	1,813

Table 8: Performance of relation type disambiguation when connectives are known.

	Our Model			Li et al. (2015)		
Relation Type	P	R	F1	P	R	F1
causality	0.9584	0.9420	0.9490	0.8380	0.6840	0.7510
coordination	0.9566	0.9734	0.9645	0.8250	0.9360	0.8770
transition	0.9504	0.9178	0.9318	0.7850	0.5960	0.6700
explanation	0.9754	0.9407	0.9563	0.8970	0.8280	0.8590

Table 9: Performance of relation type disambiguation on component level.

We further evaluate the relation type classification on the pipelined system. We predict the relation type with features ALL-NUM for each connective candidate extracted by using the Length+Score approach. The 10-fold for paragraphs specified in Section 5.1 is used. We obtain micro-averaged F1 score of 0.7458. Compared with the F1 score of 0.7693 shown in Table 6, the performance only decreases a little when one more module is integrated into the pipelined system. That shows the effectiveness of our model for relation type disambiguation.

5.4 Connective Argument Extraction

To evaluate argument extraction individually, we first assume all connectives are already correctly identified. The 10-fold for paragraphs specified in Section 5.1 is used for cross-validation. The precision, recall, and F1 scores for the argument boundaries are computed. In addition, accuracy scores evaluated on 1,813 connective instances are computed. Each instance is counted as correct only when all boundaries are all correctly identified. The averaged results over all folds are reported in Table 10. While the best F1 for argument boundaries is 0.7848, the accuracy for the connectives as evaluation units is only 0.4074. It means that for each connective, we are able to recover most of its argument boundaries, but it is challenging to recover all of them at the same time.

An analysis on the errors reveals that our model can handle the relations that have exact two arguments. For more arguments, the error rates are almost close to 1. In addition, out of the 1,074 error cases, there are only 164 cases that both sides of the interval the arguments span over are incorrect. The reason behind this is probably due to the fact that the existence of a connective often gives strong hint on at least one side of the interval. On the contrary, there is often no explicit indication on the boundary of the other side of the interval.

Finally, we evaluate the performance of the pipelined Chinese discourse parser. Here, the Length+Score approach is used for connective extraction, the ALL-NUM features are used to disambiguate the 4 top-level relation types, and the CRF models with ALL features are used for argument boundary detection. Each explicit relation is counted as true positive only when the three tasks are all correctly done; otherwise, it is counted as false positive. Under the rigorous evaluation, precision, recall, and F1 score are 0.2917, 0.3389 and 0.3134, respectively. We also evaluate on a relaxed partial match for argument extraction. An F1 score is computed for each argument tokenwise, and an instance is treated as correct when the number of arguments is correct and the averaged tokenwise F1 score is above a threshold.[3] The F1 scores computed with 0.3, 0.5, 0.7, and 0.9 as thresholds are 0.6013, 0.5782, 0.5063 and 0.3455, respectively.

Features	Precision	Recall	F1 Score	Accuracy
CONTEXT	0.5225	0.3030	0.3835*	0.0189
PATH	0.7660	0.5645	0.6497*	0.1471
POS	0.5576	0.3856	0.4556*	0.0734
SUBJ	0.8800	0.0788	0.1440*	0.0000
ENDCHAR	0.4606	0.2974	0.3614*	0.0000
LINK	0.7690	0.4001	0.5261*	0.0183
CONNECTIVE	0.4695	0.3046	0.3695*	0.0049
COMPONENT	0.6884	0.6698	0.6789*	0.2190
ALL	0.8024	0.7680	**0.7848**	**0.4074**

Table 10: Performance of argument boundary detection using different features.

[3] The tokenwise averaged F1 is adopted from partial scoring for CoNLL 2016 Shared Task: http://conll16st.blogspot.tw/2016/04/partial-scoring-and-other-evaluation.html.

6 Conclusion

In this paper, we investigate four issues regarding Chinese discourse analysis at the same time. We propose four types of features for discourse usage disambiguation. A greedy algorithm is also developed to resolve linking ambiguities. Besides, we also investigate relation type disambiguation and argument extraction. These modules are integrated into a pipelined system that extracts explicit discourse relations and their arguments from the raw text. The pipelined system achieves an overall F1 score of 0.3134.

There still exist some issues that need to be further investigated. Firstly, a closer integration between discourse usage disambiguation and linking disambiguation may be valuable. In our work, these two stages are pipelined. Although some linking information is used as features, the greedy algorithm still ranks each candidate individually. We expect that utilizing global relationship between conflicting candidates may improve the performance for both tasks. Secondly, the arguments for a relation may be useful for relation type recognition. However, the accuracy for argument extraction must be improved before the extracted arguments are used as features. Finally, implicit relations must be dealt with to construct the full discourse structure. Resolving these issues will be helpful to construct a complete Chinese discourse parser.

7 Acknowledgments

This research was partially supported by Ministry of Science and Technology, Taiwan, under grants MOST-102-2221-E-002-103-MY3 and MOST-105-2221-E-002-154-MY3, and National Taiwan University under grant NTU-ERP-104R890858. We are also very thankful to Professor Zhou Guodong for providing us Chinese Discourse TreeBank, and the three anonymous reviewers for their helpful comments to revise this paper.

References

Jamie Callan, Mark Hoy, Changkuk Yoo, and Le Zhao. 2009. Clueweb09 data set.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2001. Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*, pages 1--10, Stroudsburg, PA, USA.

Samuel W. K. Chan, Tom B. Y. Lai, Weijun Gao, and Benjamin K. T'sou. 2000. Mining Discourse Markers for Chinese Textual Summarization. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization*, pages 11--20, Stroudsburg, PA, USA.

Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing Chinese Word Segmentation for Machine Translation Performance. In P*roceedings of the Third Workshop on Statistical Machine Translation*, pages 224--232, Stroudsburg, PA, USA.

Huan-Yuan Chen, Wan-Shan Liao, Hen-Hsen Huang and Hsin-Hsi Chen. 2016. Fine-Grained Chinese Discourse Relation Labelling. In *Proceedings of 10th Language Resources and Evaluation Conference*, pages 1034--1038, Portorož, Slovenia.

Janez Demšar. 2006. Statistical Comparisons of Classifiers Over Multiple Data Sets. *The Journal of Machine Learning Research*, 7:1--30.

Robert Elwell and Jason Baldridge. 2008. Discourse Connective Argument Identification with Connective Specific Rankers. In *Proceedings of 2008 IEEE International Conference on Semantic Computing*, pages 198--205.

Syeed Ibn Faiz and Robert E. Mercer. 2013. Identifying Explicit Discourse Connectives in Text. *Lecture Notes in Computer Science*, volume 7884, pages 64--76. Springer Berlin Heidelberg.

Vanessa Wei Feng and Graeme Hirst. 2012. Text Level Discourse Parsing with Rich Linguistic Features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 60--68.

Seeger Fisher and Brian Roark. 2007. The Utility of Parse-derived Features for Automatic Discourse Segmentation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 488--495, Prague, Czech Republic.

Sucheta Ghosh, Richard Johansson, Giuseppe Riccardi, and Sara Tonelli. 2011. Shallow Discourse Parsing with Conditional Random Fields. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1071—1079, Chiang Mai, Thailand.

Sucheta Ghosh, Giuseppe Riccardi, and Richard Johansson. 2012. Global Features for Shallow Discourse Parsing. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '12, pages 150--159, Stroudsburg, PA, USA.

Hugo Hernault, Helmut Prendinger, David A duVerle, Mitsuru Ishizuka, et al. 2010. HILDA: A Discourse Parser Using Support Vector Machine Classification. *Dialogue and Discourse*, 1(3):1--33.

Jin-zhu Hu, Jiang-bo Shu, Shuang-yun Yao, Xing Zhou, Feng-wen Wu, and Sheng Xiao. 2009. Research on the Extraction of Relation Markers in Compound Sentences Oriented to Chinese Information Processing. *Computer Engineering and Science*, 31(10):90--93.

Jin-zhu Hu, Li-li Lei, Jin-cai Yang, Jiang-bo Shu, and Chen Jiang-man. 2011. Research on a Solving Model of the Collocations between the Relation Markers in Multiple Compound Sentences. *Computer Engineering and Science*, 33(11):177--182.

Hen-Hsen Huang, Tai-Wei Chang, Huan-Yuan Chen, and Hsin-Hsi Chen. 2014. Interpretation of Chinese Discourse Connectives for Explicit Discourse Relation Recognition. In *Proceedings of 25th International Conference on Computational Linguistics*, pages 632—643, Dublin, Ireland.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation Learning for Text-level Discourse Parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Shafiq Joty, Giuseppe Carenini, and Raymond T. Ng. 2012. A Novel Discriminative Framework for Sentence-level Discourse Analysis. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 904--915, Stroudsburg, PA.

Shafiq R Joty, Giuseppe Carenini, Raymond T Ng, and Yashar Mehdad. 2013. Combining Intra-and Multi- Sentential Rhetorical Parsing for Document-level Discourse Analysis. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 486--496.

Fang Kong, Hwee Tou Ng, and Guodong Zhou. 2014. A Constituent-based Approach to Argument Labelling with Joint Inference in Discourse Parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 68--77, Doha, Qatar.

Charles N Li and Sandra A Thompson. 1989. Mandarin Chinese: A Functional Reference Grammar. University of California Press.

Yancui Li, Jing Sun, and Guodong Zhou. 2015. Automatic Recognition and Classification on Chinese Discourse Connective. Acta Scientiarum Naturalium Universitatis Pekinensis, 2:016.

Jiwei Li, Rumeng Li, and Eduard Hovy. 2014. Recursive Deep Models for Discourse Parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 2061--2069, Doha, Qatar.

Junyi Jessy Li, Marine Carpuat, and Ani Nenkova. 2014. Cross-lingual Discourse Relation Analysis: A Corpus Study and A Semi-supervised Classification System. In *Proceedings of the 25th International Conference on Computational Linguistics*: Technical Papers, pages 577--587.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A PDTB-styled End-to-End Discourse Parser. *Natural Language Engineering*, 20:151--184.

Yancui Li, Wenhe Feng, Jing Sun, Fang Kong, and Guodong Zhou. 2014. Building Chinese Discourse Corpus with Connective-Driven Dependency Tree Structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 2105--2114.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of ICLR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed Representations of Words and Phrases and Their Compositionality. In *Advances in Neural Information Processing Systems* 26, pages 3111--3119.

Naoaki Okazaki. 2007. CRFsuite: a Fast Implementation of Conditional Random Fields (CRFs).

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532--1543.

Emily Pitler and Ani Nenkova. 2009. Using Syntax to Disambiguate Explicit Discourse Connectives in Text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 13--16, Stroudsburg, PA, USA.

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 LREC*.

Caroline Sporleder and Mirella Lapata. 2005. Discourse Chunking and Its application to Sentence Compression. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 257--264, Stroudsburg, PA, USA.

Benjamin K. T'sou, Weijun Gao, Tom B. Y. Lai, and Samuel W. K. Chan. 1999. Applying Machine Learning to identify Chinese Discourse Markers. In *Proceedings of International Conference on Information Intelligence and Systems*, pages 548--553.

Benjamin K. T'sou, Tom B. Y. Lai, Samuel W. K. Chan, Weijun Gao, and Xuegang Zhan. 2000. Enhancement of a Chinese Discourse Marker Tagger with C4.5. In *Proceedings of the Second Workshop on Chinese Language Processing*, pages 38--45, Stroudsburg, PA, USA.

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

Ben Wellner. 2009. Sequence Models and Ranking Methods for Discourse Parsing. Ph.D. thesis, Waltham, MA, USA..

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. Biometrics bulletin, pages 80--83.

Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Rashmi Prasad, Christopher Bryant, and Attapol Rutherford. 2015. The CoNLL-2015 Shared Task on Shallow Discourse Parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning: Shared Task*, Beijing, China.

Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Bonnie Webber, Attapol Rutherford, Chuan Wang, and Hongmin Wang. 2016. The CoNLL-2016 Shared Task on Multilingual Shallow Discourse Parsing. In *Proceedings of the Twentieth Conference on Computational Natural Language Learning - Shared Task*, Berlin, Germany, August. Association for Computational Linguistics.

Naiwen Xue, Fei Xia, Fu-dong Chiou, and Marta Palmer. 2005. The Penn Chinese Treebank: Phrase Structure Annotation of a Large Corpus. Nat. Lang. Eng., 11(2):207--238.

Chi-Hsin Yu, Yi-jie Tang, and Hsin-Hsi Chen. 2012. Development of a Web-scale Chinese Word N-gram Corpus with Parts of Speech Information. In *Proceedings of LREC*, pages 320--324.

Muyu Zhang, Bing Qin, and Ting Liu. 2014. Chinese Discourse Relation Semantic Taxonomy and Annotation. *Journal of Chinese Information Processing*, 28(2):28.

Yuping Zhou and Nianwen Xue. 2012. PDTB-style Discourse Annotation of Chinese Text. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 69--77, Stroudsburg, PA, USA.

Yuping Zhou and Nianwen Xue. 2015. The Chinese Discourse Treebank: a Chinese Corpus Annotated with Discourse Relations. *Language Resources and Evaluation*, 49(2):397--431.

Lanjun Zhou, Wei Gao, Binyang Li, Zhongyu Wei, and Kam-Fai Wong. 2012. Cross-lingual Identification of Ambiguous Discourse Connectives for Resource Poor Language. In Proceedings of COLING 2012: Posters, pages 1409--1418.