# SDP-JAIST: A Shallow Discourse Parsing system @ CoNLL 2016 Shared Task

**Nguyen Truong Son**
University Of Sience, VNU
Ho Chi Minh City
Viet Nam
ntson@fit.hcmus.edu.vn
nguyen.son@jaist.ac.jp

**Nguyen Le Minh**
Japan Advanced Institute of
Science and Technology
Ishikawa, 923-1292
Japan
nguyenml@jaist.ac.jp

**Abstract**

In this paper, we present an improvement of the last year architecture for identifying shallow discourse relations in texts. In the first phase, the system will detect the connective words and both of arguments by performing the Conditional Random Fields (CRFs) learning algorithm with models that are trained based on a set of features such as words, part-of-speech (POS) and pattern based features extracted from parsing trees of sentences. The second phase will classify arguments and explicit connectives into one of thirteen types of senses by using the Sequential Minimal Optimization (SMO) and Random Forest classifiers with a set of features extracted from arguments and connective along with a set of given resources. The evaluation results of the whole system on the development, test and blind data set are 29.65%, 24.67% and 20.37% in terms of F1 scores. The results are competitive with other top baseline systems in recognition of explicit discourse relations.

## 1 Introduction

The shared task of Shallow Discourse Parsing proposed by Xue et al. (2015) Xue et al. (2016) brings many opportunities for different teams in the world to solve the same task. Moreover, all built systems are evaluated objectively on the blind data sets and the TIRA evaluation platform (Potthast et al., 2014) helps us can compare and analyze the performance of different approaches . The result last year was impressive with many approaches had been implemented to solve this task (Xue et al., 2015). However, this task is still challenging task in the Natural Language Processing field because it has some difficult sub-tasks such as recognizing implicit discourse relations.

Our participating system of this year is an improvement of the last year system. It also has two main phases including recognizing arguments and connective words in the first phase then predicting the sense of discourse relations in the second phase. However, there are some changes in this year implementation. In the first phase, instead of tagging connective words and arguments at the same time as the last year one, we split this step into some sub steps. That means connective words will be identified at the first step then they are used as features for arguments tagging steps. Besides, we exploit more kinds of pattern based features based on syntactic parse trees to recognize arguments. In the phase of sense prediction, this year we also focus for both explicit and non-explicit sense classification with the exploiting of many kind of features based on resources such as MPQA Subjective lexicon, word embedding representation. These changes make a significant improvement for recognizing connective words, arguments and sense classification. The results are very competitive with top baseline systems in recognizing of explicit discourse relations.

This paper is organized as follows. Section 2 describes the details of our implemented system. Section 3 presents experimental results and some result analysis. Finally, Section 4 presents some conclusions and future works.

## 2 System Description

Our system focuses on recognizing discourse relations whose arguments are located in the same sentences (SS-type) and discourse relations whose arguments in two consecutive sentences (2CS-type) because they account for over 92% of total relations. Our system consists of two main phases in-



*Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 143–149,
Berlin, Germany, August 7-12, 2016. 

cluding *Connective and Argument detection phase* and *Sense classification phase* . In the first phase, the system will take parsed documents to identify explicit connective words and then identify arguments for both SS-type and 2CS-type discourse relations. After connective words and arguments are identified, they will be passed through the sense classification phase to identify the sense of discourse relations. The work-flow of our discourse parsing system is displayed in Figure1. We

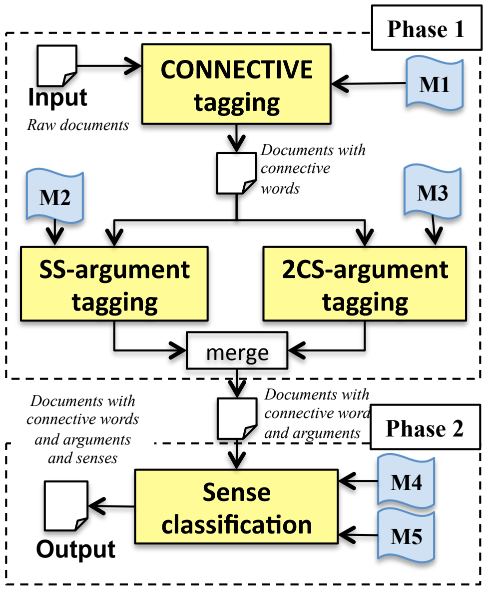



Figure 1: System work flows

have trained 5 models to recognize components of discourse relations. Models M1, M2 and M3, which are trained using CRF++ toolkit of Kudo (2005), an implementation of Conditional Random Fields proposed by Lafferty et al. (2001), are used for identifying connective words and SS-type and 2CS-type arguments. Besides, models M4 and M5, which are trained by SMO (Platt, 1998) and Random Forest (Breiman, 2001), are used for identifying the sense of explicit and non-explicit discourse relations. The details of these two phases are described in Section 2.1 and Section 2.2.

## 2.1 Phase 1: Identify connective words and arguments

We use the same approach for identifying connective words and arguments. We cast the task of recognizing these elements as a sequence labeling task. We train CRFs models to assign a specific IOB label for each token (e.g. B-C and I-C for tokens which are begin or inside of a connective word). In order to train these models, we have extracted many kind features of token. For each token, we capture features in a window size of 5 tokens including two previous tokens, the current one and two next tokens.

### 2.1.1 Features for identifying connective words

Table 1 contains a list of features (Group A) which was used to train the model for identifying explicit connective words. Beside words and their POSs (A1), we use a feature that indicates whether or not the token belongs to the list of predefined candidates extracted from the training corpus (A2). Moreover, we use two features based on syntactic parse trees of sentences including the *path-to-root* from token's POS node to the ROOT node (A3) and the *sibling-nodes-sequence* of token's POS node (A4). These features can help the machine learning algorithms to avoid some borderline cases. An example of these features are showed in Figure 2. In the case (a) of this example, *path-to-root* and *sibling-nodes-sequence* of token "*and*" are *CC-NP-...-ROOT* and *NNS-CC-NNS*. In the case (b), *path-to-root* and *sibling-nodes-sequence* of token "*and*" are *CC-S-ROOT* and *S-,-CC-S*. In this example, based on the values of these two features, it is easy to see that the token "*and*" in case (b) is likely a correct connective word more than the one in case (a). Furthermore, which parts of a verb phrase, noun phrase or a preposition phrase that the token belongs to (A5) are also a helpful information to help identifying connective words.

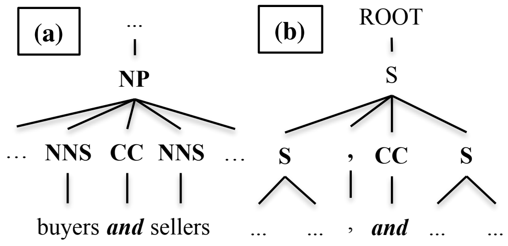
| # | **Feature description** |
|---|---|
| A1 | Word; Part of Speech |
| A2 | Does the token belong to candidate list? |
| A3 | Path to root node of the token |
| A4 | Sibling paths of POS node |
| A5 | Which parts of NP, VP, PP does the token belongs? |
| A6 | Position of token in sentence |

Table 1: Features for the connective tagging step

### 2.1.2 Features for identifying SS-type and 2CS-type arguments

All features for identifying arguments are listed in Table 2. There are three groups of features. While group B contains features that help to identify both of two argument types, group C and D contain

Figure 2: Example of path-to-root and sibling sequence feature for connective tagging




specialized features for recognizing SS-type and 2CS-type arguments. We categorize these features into two types including non-pattern-based features and pattern-based features.

The non-pattern-based features of a token consists of the token and its POS (B1), the labels received from the connective tagging step (B2), the category of Brown cluster that the token belongs to (B3), and the sentence order (1 or 2) of the token in a pair of two consecutive sentences.

Moreover, by analyzing the training corpus and linguistic features of discourse relations, we realize that there is a strong relationship between the syntactic parse trees of sentences and the boundaries of arguments and connective words. Therefore, we exploit a set of *pattern-based features* built from syntactic parse trees to capture arguments and connective of discourse relations as well as to capture some syntactic units such as phrases or clauses. If a text span matches with a pattern, their tokens will receive special values for this pattern-based feature. Below is the list of pattern-based features:

- Patterns that capture syntactic units such as subordinate clauses and phrases (B4, D6)

- Patterns that capture some useful language expressions including report statements (B5) and relative clauses (C1). For example, pattern B5 can capture some span texts such as "*he said that ...*" or "*Mr. X said ...* " or pattern C1 can capture relative clause such as "*which ...*" and "*who ...*" . If a text span matches with these patterns, their tokens rarely belong to discourse relations.

- Patterns that capture SS-type arguments: We use 4 types of pattern based features (C1, C2, C3, C4) in order to capture some popular of SS-type discourse expressions in natural language. Figure 3 shows an example of a text span with two clauses connected by a conjunction that matches the pattern S-CC-C (feature C2). In this case, it is no doubt that these two clauses and the conjunction are two arguments and the connective of a discourse relation. Another example is illustrated in Figure 4.

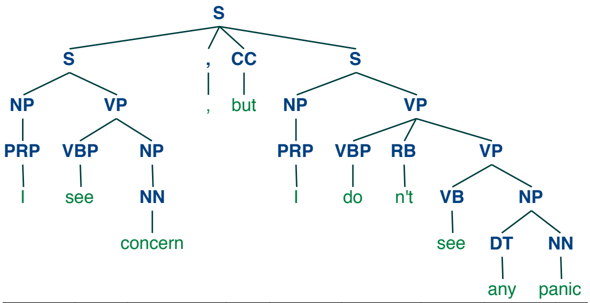
- Patterns that capture 2CS-type arguments: we used pattern based features D2, D3, D4 and D5 to capture text spans that are usually use in the second arguments of discourse relations. Figure 5 shows a sentence that matches with the pattern D5.

Table 2: List of features for the arguments tagging task

| # | **Feature description** |
|---|---|
| *Group B: common features* | |
| B1 | Word; Part of Speech |
| B2 | Connective label |
| B3 | Brown cluster |
| B4 | *Pattern* NP, VP, PP |
| B5 | *Pattern* Report statements |
| *Group C: Features for identifying SS-type Args* | |
| C1 | *Pattern* SBAR relative clause pattern |
| C2 | *Pattern* S-CC-S, SBAR-CC-SBAR |
| C3 | *Pattern* SBAR-NP-VP |
| C4 | *Pattern* SBAR begins with preposition |
| *Group D: Features for identifying 2CS-type Args* | |
| D1 | Which order of sentence does the token belong ? |
| D2 | *Pattern* SBAR begins with a conjunctive |
| D3 | *Pattern* SBAR begins with a NP follows by an adverb (e.g. also) and VP |
| D4 | *Pattern* Adverb is followed by a clause |
| D5 | *Pattern* Sentences with preposition phrases such as "for example", "by comparison", … |
| D6 | *Pattern* SBAR subordinate clause |

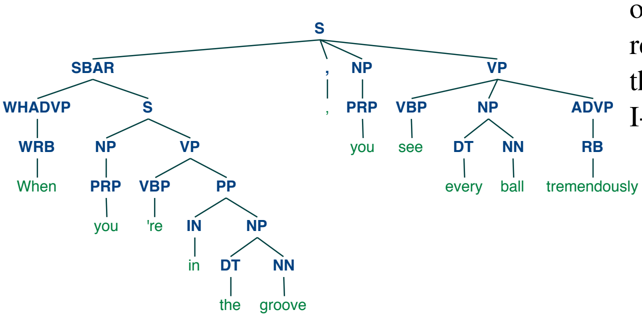
### 2.2 Phase 2: Sense classification

We use SMO and Random Forest classifier for training the models for sense classification of explicit and non-explicit discourse relations. From arguments and connectives of all discourse discourses we extract a set of features that help classifiers to build the models and classify new instances. Below are features used for non-explicit sense classification task in our system, features of




| I | see | concern | , | but | I | .. | panic | . |
|---|---|---|---|---|---|---|---|---|
| B-S1 | I-S1 | I-S1 | O | B-CC | I-S2 | .. | I-S2 | O |

Figure 3: Example of pattern S-CC-S. If a text span matches with this pattern, their tokens will receive values in{B-S1, I-S1, B-S2, I-S2, B-CC} for this feature




| When | you | … | groove | , | you | … | tremendously |
|---|---|---|---|---|---|---|---|
| B-A | B-S1 | … | I-S1 | O | B-S2 | .. | I-S2 |

Figure 4: Example of pattern SBAR-NP-VP. If a text span matches with this pattern, their tokens will receive values in {B-S1, I-S1, B-S2, I-S2, B-A} for this feature

explicit sense classification are described in the end of this section:

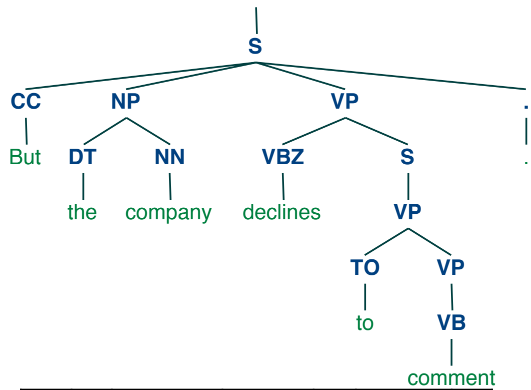
- **Similarity features**: instead of using the cosine similarity between whole text span of two arguments, we compute 5 cosine similarity scores of nouns, noun phrases, verbs, verb phrase, adjectives between two arguments to obtain similarity features.

- **MPQA Subjectivity Lexicon** (Wilson et al., 2009)- feature): We realize that the polarity (positive, negative, neural) of words may be a good indicator for machine learning algorithms to identify the sense of discourse relations, especially some kinds of discourse relations such as *Comparison.Contrast* of *Contingency.Condition*. We create these features based on the presence of words of arguments in the lexicon.

- **Word pair features**: From the training corpus, we extract frequent word pairs of arguments (frequency $>= 100$) as a feature set for sense classification. Moreover, we have used Information Gain (Sebastiani, 2002) method to reduce the size of this feature set and keep important pairs. We check the present of word pair in two arguments in these lists to obtain these features.

- **POS Pattern features**: POS patterns of sentences may indicate some sentence patterns that useful for sense classification such as patterns with modal verbs, patterns indicate the passive voice expression or patterns begin with a prepositions which express the purpose. Base on pre-defined regular expressions, we extract a list of POS patterns that have high frequency ($>= 100$) in training corpus. Table 3 shows top patterns extracted from the training corpus.

- **Word2Vec pair features**: Some pair of words have the same context relationship that




| But | the | company | declines | to | comment | . |
|---|---|---|---|---|---|---|
| B-C | B-S | I-S | I-S | I-S | I-S | O |

Figure 5: Example of pattern D2, which help recognizing second arguments of 2CS-type discourse relations. If a text span matches with this pattern, their tokens will receive values in {B-S, I-S, B-P, I-P} for this feature

may reveal the meaning of discourse relation. Such as, "*find*" and "*know*" may reveal a *Contingency.Cause.Result* discourse relations. First, for each sense, we create a word pair list from word pairs of arguments of discourse relation of that sense in the training corpus that have the cosine similarity score using word2vec higher than a given threshold (we use threshold $= 0.2$). Then, for feature extraction step, we check whether or not a pair of word from argument exists in these lists.

- **Regular expressions**: We use patterns that catch the appearance of some useful expressions for sense classification such as "could", "would", "should", etc.
- **Other features**: Beside above features, we use some extra information such as the proportion of length of argument texts over the length of sentence, number of sentences that arguments of a discourse relations covers.

Table 3: Top frequent POS patterns in arguments of discourse relations training corpus

| Pattern in ARG1 | count | Pattern in ARG2 | count |
|---|---|---|---|
| MD VB | 4094 | MD VB | 4014 |
| VBZ VBN | 1982 | VBZ VBN | 2074 |
| MD VB VBN | 926 | MD VB VBN | 969 |
| MD RB VB | 912 | MD RB VB | 932 |
| VBZ RB VBN | 413 | VBZ RB VBN | 417 |
| IN DT NN TO | 307 | IN DT NN TO | 273 |
| MD VB TO VB | 294 | MD VB TO VB | 256 |
| IN NN TO | 272 | IN NN TO | 247 |
| IN NNS TO | 173 | MD RB VB VBN | 179 |
| MD RB VB VBN | 162 | IN NNS TO | 168 |

Although all above feature types have a somehow contribute for identifying senses of non-explicit discourse relations, sometimes it does not help algorithms to predict sense of explicit discourse relations. Therefore, beside connective words, a very strong features, we just use 3 more features including POS of connective words, POS-patterns, Regular expressions for sense classification of explicit discourse relations.

## 3 Experimental results

Table 4 shows the official results of our system on three given data sets. Due to the changes in the system architecture and more kinds of features, our system this year has a significant improvement in identifying discourse relations, especially explicit discourse relations. The results of recognizing explicit discourse relations are very competitive with top-rated systems last year. That means our discovery feature sets played an important role for the task of Shallow Discourse Parsing. Moreover, the result on the development data set are higher than blind and test data sets. With the support from connective words, the results of explicit discourse relations are better than non-explicit discourse relations. The results of recognizing non-explicit discourse relations are still low because we do not have effective features for this kind of discourse relations. Table 5 and Table 6 show the

Table 4: Official result of main task on development, test and blind data sets

| | DEV dataset | | | TEST dataset | | | BLIND dataset | | |
|---|---|---|---|---|---|---|---|---|---|
| | ALL | Exp. | Non Exp. | ALL | Exp. | Non Exp. | ALL | Exp. | Non Exp. |
| **Arg1 extraction** | | | | | | | | | |
| **P** | 53.9 | 58.5 | 47.5 | 49.5 | 51.3 | 45.7 | 48.5 | 48.7 | 45.6 |
| **R** | 57.9 | 63.1 | 50.9 | 53.0 | 56.7 | 47.5 | 48.2 | 56.1 | 40.6 |
| **F1** | 55.8 | 60.7 | 49.1 | 51.2 | 53.8 | 46.6 | 48.3 | 52.2 | 43.0 |
| **Arg2 extraction** | | | | | | | | | |
| **P** | 61.7 | 69.7 | 54.5 | 58.7 | 68.4 | 49.9 | 61.7 | 65.5 | 58.5 |
| **R** | 66.3 | 75.1 | 58.4 | 62.9 | 75.6 | 52.0 | 61.3 | 75.4 | 52.0 |
| **F1** | 63.9 | 72.3 | 56.4 | 60.8 | 71.8 | 50.9 | 61.5 | 70.1 | 55.1 |
| **Arg 1 Arg2 extraction** | | | | | | | | | |
| **P** | 45.8 | 50.4 | 41.7 | 40.6 | 43.1 | 38.3 | 39.0 | 38.7 | 39.4 |
| **R** | 49.3 | 54.4 | 44.7 | 43.5 | 47.7 | 39.9 | 38.8 | 44.5 | 35.0 |
| **F1** | 47.5 | 52.3 | 43.1 | 42.0 | 45.3 | 39.1 | 38.9 | 41.4 | 37.1 |
| **Explicit connective** | | | | | | | | | |
| **P** | 85.0 | 85.0 | - | 83.4 | 83.4 | - | 79.5 | 79.5 | - |
| **R** | 91.6 | 91.6 | - | 92.2 | 92.2 | - | 91.5 | 91.5 | - |
| **F1** | 88.2 | 88.2 | - | 87.6 | 87.6 | - | 85.1 | 85.1 | - |
| **Parser** | | | | | | | | | |
| **P** | 30.6 | 48.1 | 15.1 | 25.5 | 41.4 | 11.9 | 20.3 | 33.2 | 11.9 |
| **R** | 28.8 | 45.4 | 14.2 | 23.9 | 37.5 | 11.5 | 20.4 | 28.8 | 13.3 |
| **F1** | 29.7 | 46.7 | 14.6 | 24.7 | 39.4 | 11.7 | 20.4 | 30.8 | 12.6 |

comparison of our system and 4-top-rated last year systems. On both of two these data sets, our results are not good at recognizing non-explicit discourse relations.

Moreover, there may have more than one explicit discourse relations in a pair of consecutive sentence but our current implementation just keeps only one and remove the others. Therefore, this may affects the performance of recognizing explicit connective words.

The result of supplement task are showed in Table 7. We have chosen Random Forest classifier for non-explicit discourse relations and SMO for explicit discourse relations because they achieved best results in the development data set. Table 8 shows the contribution of exploited feature sets. In non-explicit sense classification the result would improve significantly if we use these features.

Table 5: Result on test data set of our system and top-4 last year systems including **lan**: (Wang and Lan, 2015), **ste.** (Stepanov et al., 2015), **yo.** (Yoshida et al., 2015)

| | **System** | lan | step. | yo. | xue | **Our system** |
|---|---|---|---|---|---|---|
| ALL | Arg 1 Arg2 | 49.4 | 40.7 | 43.8 | 30.2 | 42.0 |
| | Arg1 | 60.1 | 47.8 | 52.5 | 37.8 | 51.2 |
| | Arg2 | 72.5 | 60.7 | 64.4 | 46.5 | 60.8 |
| | Connective | 94.2 | 92.7 | 89.1 | 89.4 | 87.6 |
| | **Parser** | 29.7 | 25.4 | 25.0 | 21.8 | 24.7 |
| Exp. | Arg 1 Arg2 | 45.2 | 44.6 | 38.8 | 41.6 | 45.3 |
| | Arg1 | 50.7 | 50.1 | 46.1 | 49.8 | 53.8 |
| | Arg2 | 77.3 | 76.2 | 68.3 | 68.6 | 71.8 |
| | Connective | 94.2 | 92.7 | 89.1 | 89.4 | 87.6 |
| | **Parser** | 40.0 | 39.6 | 34.5 | 37.6 | 39.4 |
| Non-Expl | Arg 1 Arg2 | 53.0 | 37.3 | 48.8 | 19.4 | 39.1 |
| | Arg1 | 67.1 | 44.4 | 57.9 | 24.7 | 46.6 |
| | Arg2 | 68.3 | 47.4 | 60.1 | 25.3 | 50.9 |
| | **Parser** | 20.8 | 13.3 | 15.1 | 6.6 | 11.7 |

Table 6: Result on blind data set of our system and top-4 last year systems including **lan** , **ste.** (Stepanov et al., 2015), **li** (Kong et al., 2015), **minh** (Nguyen et al., 2015)

| | **System** | lan | ste. | li | minh | **Our system** |
|---|---|---|---|---|---|---|
| ALL | Arg 1 Arg2 | 46.4 | 38.9 | 33.2 | 32.1 | 38.9 |
| | Arg1 | 55.8 | 46.5 | 46.3 | 41.0 | 48.3 |
| | Arg2 | 74.5 | 62.6 | 61.7 | 48.5 | 61.5 |
| | Connective | 91.9 | 89.9 | 91.6 | 61.7 | 85.1 |
| | **Parser** | 24.0 | 21.8 | 18.5 | 18.3 | 20.4 |
| Exp. | Arg 1 Arg2 | 41.4 | 39.6 | 30.4 | 34.2 | 41.4 |
| | Arg1 | 48.3 | 49.0 | 36.4 | 44.1 | 52.2 |
| | Arg2 | 74.3 | 70.7 | 73.0 | 51.4 | 70.1 |
| | Connective | 91.9 | 89.9 | 91.6 | 61.7 | 85.1 |
| | **Parser** | 30.4 | 30.0 | 23.0 | 27.2 | 30.8 |
| Non-Expl | Arg 1 Arg2 | 50.4 | 38.3 | 35.9 | 30.4 | 37.1 |
| | Arg1 | 60.9 | 43.3 | 49.9 | 36.9 | 43.0 |
| | Arg2 | 74.6 | 56.6 | 51.1 | 46.1 | 55.1 |
| | **Parser** | 18.9 | 15.8 | 14.4 | 11.3 | 12.6 |

Table 7: Result of sense classification task

| | **DEV dataset** | | | **TEST dataset** | | | **BLIND dataset** | | |
|---|---|---|---|---|---|---|---|---|---|
| | ALL | Exp. | Non-Exp. | ALL | Exp. | Non Ex. | ALL | Exp. | Non-Exp. |
| **P** | 60.5 | 90.3 | 34.3 | 57.4 | 88.7 | 28.8 | 51.4 | 74.9 | 31.4 |
| **R** | 60.5 | 90.3 | 34.3 | 57.4 | 88.7 | 28.8 | 51.3 | 74.6 | 31.4 |
| **F1** | 60.5 | 90.3 | 34.3 | 57.4 | 88.7 | 28.8 | 51.3 | 74.8 | 31.4 |

Table 8: Comparison between feature sets in sense classification task

| | **Features** | **Random Forest** | **SMO** |
|---|---|---|---|
| Non-Exp. | Similarity features | 28.0 | 29.9 |
| | All features mentioned above | 36.5 | 30.3 |
| Exp. | Connective words | 89.4 | 89.4 |
| | Connective words and their POS POS pattern of arguments, Regular expression and Others | 87.1 | 90.3 |

## 4 Conclusion

Our approach has some positive points. It achieved a better result in comparison with our system last year. Moreover, compare to top-rated systems, the result of explicit discourse parsing is very competitive. This year we concentrated on solving both of explicit and non-explicit sense classification tasks. In non-explicit sense classification, it achieved some initial results.

There are a few things that can be improved in our system such as solving the problem that there may be more than one explicit discourse relations in pairs of consecutive sentences or finding effective features for implicit sense classifications.

Recognizing non-explicit discourse relations and explicit discourse relations whose arguments are not located in two adjacent sentences is still difficult for both identification of arguments and sense classification task. They are still a challenge for us at the moment. In the future, deep learning techniques may be promising approaches to achieve the better results.


## Acknowledgment

This research is supported by JAIST Student Research Grants program.

## References


Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Fang Kong, Sheng Li, and Guodong Zhou. 2015. The sonlp-dp system in the conll-2015 shared task. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 32–36, Beijing, China, July. Association for Computational Linguistics.

Taku Kudo. 2005. Crf++: Yet another crf toolkit. *Software available at http://crfpp. sourceforge. net*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Son Nguyen, Quoc Ho, and Minh Nguyen. 2015. Jaist: A two-phase machine learning approach for identifying discourse relations in newswire texts. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 66–70, Beijing, China, July. Association for Computational Linguistics.

John Platt. 1998. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research, April.

Martin Potthast, Tim Gollub, Francisco Rangel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. 2014. Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In Evangelos Kanoulas, Mihai Lupu, Paul Clough, Mark Sanderson, Mark Hall, Allan Hanbury, and Elaine Toms, editors, *Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14)*, pages 268–299, Berlin Heidelberg New York, September. Springer.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Evgeny Stepanov, Giuseppe Riccardi, and Ali Orkan Bayer. 2015. The unitn discourse parser in conll 2015 shared task: Token-level sequence labeling with argument-specific models. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 25–31, Beijing, China, July. Association for Computational Linguistics.

Jianxiang Wang and Man Lan. 2015. A refined end-to-end discourse parser. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 17–24, Beijing, China, July. Association for Computational Linguistics.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational linguistics*, 35(3):399–433.

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*, pages 1–16, Beijing, China, July. Association for Computational Linguistics.

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.

Yasuhisa Yoshida, Katsuhiko Hayashi, Tsutomu Hirao, and Masaaki Nagata. 2015. Hybrid approach to pdtb-styled discourse parsing for conll-2015. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 95–99, Beijing, China, July. Association for Computational Linguistics.