Shallow Discourse Parsing Using Convolutional Neural Network

Lianhui Qin[1,2], **Zhisong Zhang**[1,2], **Hai Zhao**[1,2,*]
[1]Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, 200240, China
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China
{qinlianhui, zzs2011}@sjtu.edu.cn,zhaohai@cs.sjtu.edu.cn

Abstract

This paper describes a discourse parsing system for our participation in the CoNLL 2016 Shared Task. We focus on the supplementary task: Sense Classification, especially the Non-Explicit one which is the bottleneck of discourse parsing system. To improve Non-Explicit sense classification, we propose a Convolutional Neural Network (CNN) model to determine the senses for both English and Chinese tasks. We also explore a traditional linear model with novel dependency features for Explicit sense classification. Compared with the best system in CoNLL-2015, our system achieves competitive performances. Moreover, as shown in the results, our system has higher F1 score on Non-Explicit sense classification.

1 Introduction

This paper presents the Shanghai Jiao Tong University discourse parsing system for the CoNLL 2016 Shared Task (Xue et al., 2016) on Shallow Discourse Parsing and the supplementary tasks of sense classification for English and Chinese.

As shown by the results of the same task in CoNLL 2015 (Xue et al., 2015), sense classification has been found more difficult than other subtasks, especially determining Non-Explicit senses which is the bottleneck of the end-to-end discourse parsing system. Without the discourse connectives which provide strong indications, the Non-Explicit relations between adjacent sentences are difficult to figure out. Therefore, our primary work is to improve sense classification components, especially on Non-Explicit relations. For other components such as connectives detection and arguments extraction, we just follow the top ranked system (Wang and Lan, 2015) in CoNLL-2015, which is as the baseline system in this paper.

In CoNLL-2015, various approaches were explored to conquer the sense classification problem, which is a straightforward multi-category classification task (Okita et al., 2015; Wang and Lan, 2015; Chiarcos and Schenk, 2015; Song et al., 2015; Stepanov et al., 2015; Yoshida et al., 2015; Sun et al., 2015; Nguyen et al., 2015; Laali et al., 2015). Typical data-driven machine learning methods, like Maximum Entropy and Support Vector Machine, were adopted. Some of them selected lexical and syntactic features over the arguments, including linguistically motivated word groupings such as Levin verb classes and polarity tags. Brown cluster features, surface features and entity semantics were also effective to enhance sense classification. Additionally, paragraph embeddings were also used to determine the senses (Okita et al., 2015). In other previous work of implicit sense classification, Chen et al (2015) used word-pair features for predicting missing connectives, Zhou et al. (2010) attempted to insert discourse connectives between arguments with the use of a language model, Lin et al. (2009) applied various feature selection methods. Although traditional methods have performed well on semantic tasks through feature engineering (Zhao et al., 2009a; Zhao et al., 2009b; Zhao et al., 2013), they still suffer from data sparsity problems.

Recently, Neural Network (NN) methods have shown competitive or even better performance

[*] Corresponding author. This paper was partially supported by Cai Yuanpei Program (CSC No. 201304490199 and No. 201304490171), National Natural Science Foundation of China (No. 61170114 and No. 61272248), National Basic Research Program of China (No. 2013CB329401), Major Basic Research Program of Shanghai Science and Technology Committee (No. 15JC1400103), Art and Science Interdisciplinary Funds of Shanghai Jiao Tong University (No. 14JCRZ04), and Key Project of National Society Science Foundation of China (No. 15-ZDA041).

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 70–77,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics

than traditional linear models with hand-crafted sparse features for some Nature Language Process (NLP) tasks (Wang et al., 2013; Wang et al., 2014; Cai and Zhao, 2016; Wang et al., 2016; Zhang and Zhao, 2016), such as sentence modeling (Kalchbrenner et al., 2014; Kim, 2014). In Non-Explicit sense classification, due to the absence of discourse connectives, the task is exactly to classify a sentence pair, where CNN could be utilized.

For Explicit sense classification which has strong discourse relation information provided by the connectives, we will use traditional linear methods with novel dependency features.

The rest of the paper is organized as follows: Section 2 briefly describes our system, Section 3 introduces the CNN model for modeling sentence pairs, Section 4 discusses our main works including Explicit sense classification and Non-Explicit sense classification, Section 5 shows our experiments on sense classification and Section 6 reports our results on the final official evaluation. Section 7 concludes this paper.

2 System Overview

Our parsing system uses the sequential pipeline following by (Lin et al., 2014; Wang and Lan, 2015). Figure 1 shows the system pipeline. The system can be roughly split into two parts: the Explicit parser and the Non-Explicit parser. We will give a brief introduction for every components. The overall parser starts from detecting discourse connectives for the Explicit Parser. Then the types of relative location of Argument1 (Arg1) and Argument2 (Arg2) are identified: Arg1 located in the exact previous sentence of Arg2 (noted as PS) or both arguments are within the same sentence (noted as SS). For the last part of Explicit parser, the tuples (Arg1, Connective, Arg2) are classified into one of the Explicit relation senses. For the Non-Explicit parser, it classifies the senses of Non-Explicit with original arguments and then extracts the arguments of the argument pairs. Finally, the senses of Non-Explicit argument pairs are again decided with refined arguments. Among all subtasks, we will focus on sense classification the other parts have been done relatively well in previous work.

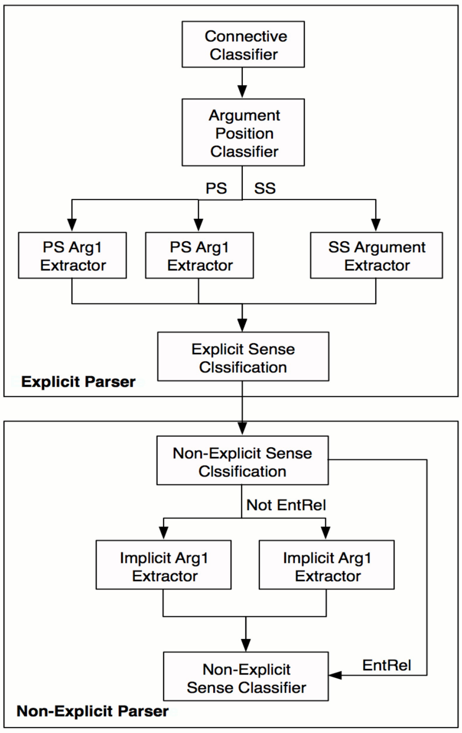

Figure 1: System pipeline for the discourse parser

3 Convolutional Neural Network

Each sentence could obtain a sentence vector through CNN and the final classification is based on the transformations of the sentence vectors. Although both Explicit and Non-Explicit tasks could utilize the neural model, CNN might be more apposite for the Non-Explicit one because of lacking indicating connectives.

The architecture of our CNN model, is illustrated in Figure 2. Firstly, a look-up table is utilized to fetch the embeddings of words and part-of-speech (POS) tags, forming two sentence embeddings which will be the input of the convolutional layer. Through the convolution and max pooling operations, two sentence vectors are obtained. Finally, these vectors will be sent to the final softmax layer after concatenated.

Embedding For a sentence $\mathbf{S} = \mathbf{w}_1\mathbf{w}_2 \dots \mathbf{w}_n$ and POS sequence $\mathbf{P} = \mathbf{p}_1\mathbf{p}_2 \dots \mathbf{p}_n$, the sentence embedding M is formed through projection and concatenating. Following the jargons in the task, the input sentences will be called "Arguments" and the two arguments are represented as follows:

$$\mathbf{M}^1 = [\mathbf{w}_1^1 \oplus \mathbf{p}_1^1; \mathbf{w}_2^1 \oplus \mathbf{p}_2^1; \dots; \mathbf{w}_n^1 \oplus \mathbf{p}_n^1]$$
$$\mathbf{M}^2 = [\mathbf{w}_1^2 \oplus \mathbf{p}_1^2; \mathbf{w}_2^2 \oplus \mathbf{p}_2^2; \dots; \mathbf{w}_n^2 \oplus \mathbf{p}_n^2]$$

Here $\mathbf{w}_i^j \in \mathbb{R}^{d_w}$ is the word vector corresponding to the i-th word in the j-th argument, and $\mathbf{p}_i^j \in \mathbb{R}^{d_p}$ is the POS vector for $\mathbf{w}_i^j$, where d_w and d_p respectively stand for the dimensions of word and POS vectors. $\oplus$ and ; are the concatenation operators on different dimensions. Considering the efficiency, we specialize a max sentence length for both arguments, and apply truncating or zero-padding when needed.

Convolutional layer Filter matrices $[\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_k]$ with several variable sizes $[l_1, l_2, \ldots, l_k]$ are utilized to perform the convolution operations for the sentence embeddings. Via parameter sharing, this feature extraction procedure become same for both arguments. For the sake of simplicity, ignoring the superscripts, we will explain the procedure for only one argument. The sentence embedding will be transformed to sequences $\mathbf{C}_j (j \in [1, k])$:

$$\mathbf{C}_j = [\ldots; \tanh(\mathbf{W}_j \cdot \mathbf{M}_{[i:i+l_j-1]} + \mathbf{b}_j); \ldots]$$

Here, $[i : i + l_j - 1]$ indexes the convolution window. Additionally, We apply wide convolution operation between embedding layer and filter matrices, because it ensures that all weights in the filters reach the entire sentence, including the words at the margins.

Max Pooling A one-max-pooling operation is adopted after convolution and the sentence vector $\mathbf{s}$ is obtained through concatenating all the mappings for those $\mathbf{k}$ filters.

$$\mathbf{s} = [\mathbf{s}_1 \oplus \cdots \oplus \mathbf{s}_j \oplus \cdots \oplus \mathbf{s}_k]$$
$$\mathbf{s}_j = \mathbf{max}(\mathbf{C}_j)$$

In this way, the model can capture the most important features in the sentence with different filters.

Concatenating and Softmax Now adding the superscripts and considering the two arguments ($\mathbf{s}^1$, $\mathbf{s}^2$), they are concatenated to form the argument-pair representation vector $\mathbf{v}$ as below:

$$\mathbf{v} = \mathbf{s}^1 \oplus \mathbf{s}^2$$

For the final labeling decision, a softmax layer will be applied using the argument-pair vector $\mathbf{v}$.

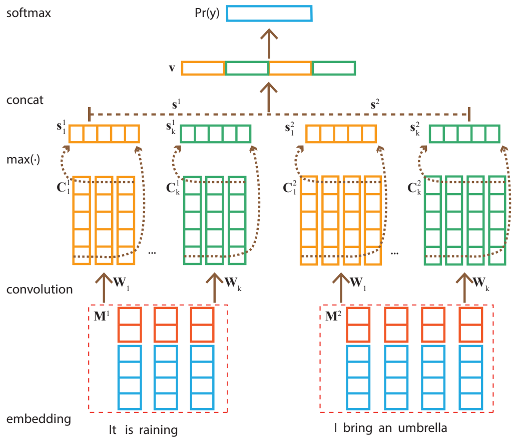

Figure 2: Our neural model for sentence classification.

Training The training object *J* will be the cross-entropy error *E* with *L2* regularization:

$$E(\hat{y}, y) = -\sum_{j}^{l} y_j \times \log(Pr(\hat{y}_j))$$

$$J(\theta) = \frac{1}{m} \sum_{k}^{m} E(\hat{y}^{(k)}, y^{(k)}) + \frac{\lambda}{2} \|\theta\|^2$$

where y_j is the gold label and $\hat{y}_j$ is the predicted one. For the optimization process, we apply the diagonal variant of AdaGrad (Duchi et al., 2011) with mini-batches.

4 Sense Classification

Now we will discuss about the sense classification task. Both the Explicit and Non-Explicit labeling are typical classification tasks with the argument-pair as the input and the CNN model could be applied to both of them. However, the Explicit task provides the connectives which are the crucial indicators and we find that CNN performs slightly poorly on this task even if embeddings for indicators are concatenated. Thus, for the Explicit task, we will adopt the traditional linear model considering only the features related with the indicators and CNN model will be applied to the more difficult Non-Explicit task.

4.1 Explicit Sense Classification

For the Explicit classification task, connectives provide the crucial and decisive information. The connective itself has been found to be a very good

data set	baseline	C+C POS	add C-HP
English	90.14	91.35	92.11
Chinese	-	96.15	97.43

Table 1: Explicit Sense Classification on English and Chinese development sets without error propagation.

data set	baseline	CNN model
English	42.92	45.50
Chinese	-	71.57

Table 2: Non-Explicit Sense Classification on English and Chinese development sets without error propagation.

feature, as connectives are ambiguous as pointed out in Pitler et al. (2008), and the majority of the ambiguous connectives is highly skewed toward certain senses (Lin et al., 2014). Thus, the task is in fact to disambiguate the connective under different contexts.

Although the provided context contains the two whole arguments, the most crucial indicators are still the words that near the connectives or the ones that have close syntactic dependency relations with the connectives. This might explain why plain CNN model performs poorly on this task without these key features.

Thus, for the Explicit task, we will adopt the traditional method, using Support Vector Machines (SVM) with linear kernel and manually selected features. We consider only three features which are all related to Connective C: (1) C string (2) C POS (3) C string combined with POS of C's parent node in dependency tree (noted as C-HP).

We will use an example in the Chinese task to explain the influence of the third feature which utilizes the dependency tree.

(1) 男选手的成绩是近１０年来最差的一次，说明水平在下降[*Arg1*] 而 [*Connective*] **罗莉、乔娅和莫惠兰３名女选手都是第一次参加世界大赛，均表现不错。**[*Arg2*]
(Contrast - CHTB_0310)

In Chinese, '而' is a connective with ambiguity relations of *'Contrast'* and *'Conjunction'*. Because *'Conjunction'* accounts for a large part of these instances, the classifier will tend to predict '而' as *'Conjunction'* if just using connective features. Like in this example, the sense of the instance is *'Contrast'* but is predicted as *'Conjunction'* if considering only the connective itself. But if we add the third feature, which means the combination feature '而-VC' will be added (C is '而' and POS of C's parent node is 'VC'), the classifier will correctly decide the right sense.

filter-size	on original Args
(2,3,3)	38.45
(2,4,5)	38.86
(2,6,12)	38.45
(3,3,3)	39.40
(4,8,12)	40.08
(6,8,18)	38.99

Table 3: F_1 scores (%) with different CNN filter sizes for Non-Explicit on **original** arguments on development set.

filter-size	on refined Args
(1,2,3)	45.11
(2,3,4)	44.18
(2,5,10)	44.97
(2,8,16)	43.25
(3,3,3)	45.50
(3,5,9)	43.92

Table 4: F_1 scores (%) with different CNN filter sizes for Explicit on **refined** arguments on development set.

4.2 Non-Explicit Sense Classification

The situations for the Non-Explicit task are quite different. Without the information of connectives, we have to extract the discourse relations through the two arguments, which might need semantic comprehensions sometimes. This might be hard for traditional methods because it is not easy to extract hand-craft features. The neural models which can automatically extract features may be another solution.

We apply the CNN model described in Section 3 for this task. To simplify model building and parameter tuning, and also due to the similar architectures, the model structures for sense classification components in English and Chinese are identical.

5 Experiments

Our system is trained on the PDTB 2.0 corpus. Sections 02-21 are used as training set, and Section 22 as the development set. There are two tests

Components	WSJ Test					
	baseline			our parser		
	P	R	F	P	R	F
ALL Explicit connective	94.83	93.49	94.16	92.42	94.88	93.63
Explicit Arg1 extraction	51.05	50.33	50.68	49.73	51.06	50.38
Explicit Arg2 extraction	77.89	76.79	77.33	75.73	77.75	76.73
Explicit Both extraction	45.54	44.90	45.22	44.31	45.49	44.90
Explicit only Parser	-	-	39.96	41.05	40.02	40.53
Non-Explicit Arg1 extraction	64.83	69.50	67.08	67.42	63.08	65.18
Non-Explicit Arg2 extraction	66.02	70.78	68.32	70.18	65.65	67.84
Non-Explicit Both extraction	51.20	54.89	52.98	53.44	50.00	51.67
Non-Explicit only Parser	-	-	20.74	20.66	22.11	21.36
All Arg1 extraction	59.20	61.03	60.10	59.67	58.29	58.97
All Arg2 extraction	71.43	73.64	72.52	72.82	71.13	71.97
All Both extration	48.62	50.13	49.36	49.10	47.96	48.52
All Parser	29.27	30.08	29.72	29.90	30.65	30.27

Table 5: Results of the Shallow Discourse Parsing task on English WSJ test set.

Components	Blind Test					
	baseline			our parser		
	P	R	F	P	R	F
ALL Explicit connective	93.48	90.29	91.86	88.67	93.73	91.13
Explicit Arg1 extraction	49.16	47.48	48.31	47.12	49.81	48.43
Explicit Arg2 extraction	75.61	73.02	74.29	71.58	75.56	73.57
Explicit Both extraction	42.09	40.65	41.35	40.29	42.59	41.40
Explicit only Parser	-	-	30.38	32.57	30.76	31.64
Non-Explicit Arg1 extraction	58.66	63.25	60.87	64.01	59.38	61.61
Non-Explicit Arg2 extraction	71.88	77.49	74.58	80.86	75.00	77.82
Non-Explicit Both extraction	48.58	52.37	50.41	55.44	51.42	53.35
Non-Explicit only Parser	-	-	18.87	18.32	19.75	19.01
All Arg1 extraction	55.12	56.58	55.84	56.91	55.93	56.42
All Arg2 extraction	73.49	75.43	74.45	76.59	75.28	75.93
All Both extration	45.77	46.98	46.37	48.47	47.64	48.05
All Parser	23.69	24.32	24.00	24.41	24.81	24.61

Table 6: Results of the Shallow Discourse Parsing task on English Blind test set.

sets for the shared task: Section 23 of the PDTB, and a blind test prepared especially for this task. We participate in the closed track, so only two resources (Brown Clusters and MPQA Subjectivity Lexicon) are used. test platform of CoNLL-2016 still adopts still the TIRA evaluation platform (Potthast et al., 2014).

Non-Explicit relations contains three types: *Implicit*, *EntRel* and *AltLex*. Originally *EntRel* is not treated as discourse relation in Penn Discourse TreeBank (PDTB) (Prasad et al., 2008), but this category has been included in this task and we also count it as one sense. Some instances are annotated with two senses, so the predicted sense for a relation must match one of the two senses if there is more than one sense. We compare with the best system in the competition of CoNLL 2015 (Wang and Lan, 2015), which is regarded as the baseline.

5.1 Explicit Sense Classification

Table 1 reports our results of the Explicit sense classifier on both English and Chinese development sets. Compared with the baseline, our methods obtain progress and the overall F1 score of Explicit Sense classification increases by 1.97% for English task.

For both English and Chinese sense classification, the C string and C POS features can classify most of the relations correctly. Moreover, the new combination feature based on dependency relations helps effectively disambiguate senses.

5.2 Non-Explicit Sense Classification

For the Non-Explicit task, we utilize the CNN model to model the argument pairs. Following (Wang and Lan, 2015), in the final discourse parsing pipeline, we utilize the sense classifier twice, once for original arguments (adjacent sentence pairs) and once for redefined arguments (after argument extraction). Because the two classifiers expect different inputs, we train different CNN models for these two tasks and also with slightly different hyper-parameters.

	WSJ Test			En Blind Test			CTB Test			CH Blind Test		
components	P	R	F	P	R	F	P	R	F	P	R	F
Explicit Sense Classification	89.59	89.59	89.59	75.95	75.54	75.74	93.68	92.71	93.19	75.82	73.67	74.73
Non-Explicit Sense Classification	38.20	38.20	38.20	35.38	35.38	35.38	67.41	67.41	67.41	56.35	56.35	56.35
All Parser	62.69	62.69	62.69	53.94	53.85	53.89	72.91	72.75	72.83	61.02	61.02	61.02

Table 7: Results of the supplementary task on English and Chinese.

On Original Arguments The input for this classifier will be two adjacent sentences without Explicit discourse relations. The maximum input length for both sentences is set to 80, the dimensions for word embeddings and POS embeddings are 300 and 50 respectively. The word embeddings are initialized with pre-trained word vectors using *word2vec*[1] (Mikolov et al., 2013) and other parameters are randomly initialized including POS embeddings. We employ three categories of CNN filters, and choose 512 as the number of feature maps. About the filter region sizes, Zhang and Wallace (2015) have concluded that each dataset has its own optimal range. We set the three filter sizes to 4,8,12 separately according to the empirical results in Table 3.

On Refined Arguments This module is similar to the above one but with some differences. The input will be the refined arguments and correspondingly, golden argument pairs are utilized for training. Thus, we adopt slightly different hyperparameters. The number of feature maps for each filter categories is set to 1024, and the final filter region sizes are 3,3,3 accordingly to the empirical results in Table 4. For the choice of filter region sizes, we have attempted a lot of combinations, but only the best ones are shown.

Results of classification The trained model on refined arguments could be directly utilized for part of Non-Explicit sense classification in the supplementary task and Table 2 reports the results on English and Chinese development sets. Compared to the Explicit task, the Non-Explicit task is indeed much more difficult. Using CNN, we achieve an improvement of 2.58% compared to the baseline. This result fully illustrates that CNN model is suitable to determine the Non-Explicit relations.

6 Results

We report our official results and comparisons on Shallow Discourse Parsing task on English and the supplementary tasks of sense classification on English and Chinese.

Table 5 and 6 show the performance on two test sets for English: i) (Official) Blind test set; ii) Standard WSJ test set. Our parsers give higher F1 scores than baselines: 0.55% higher on WSJ test set and 0.61% on Blind Test set, though our Explicit connective detection F1 is less than theirs at the beginning of the pipeline, which might introduce more error propagations. This might suggest that our sense classifiers play key roles in the system.

To see the performances of the sense classifiers, Table 7 shows the results for English and Chinese supplementary tasks (sense classifications on golden argument pairs without errors propagation). For Explicit sense classification, the features we proposed are proved to be effective. For Non-Explicit sense classification, our CNN model also works well on the test sets. Compared to the performance of discourse parsing sense classification components (with error propagation), the subtask results are higher. The reasons include: i) Connective detection serves as the first component of the pipeline and plays an important role, because it has a major influence on Explicit sense classification which relies heavily on discourse connectives. ii) Arguments extraction also have important effects on the classifications for both Explicit and Non-Explicit relations.

7 Conclusions

This paper describes our discourse parsing system for the CoNLL 2016 shared Task and reports our results on test data and blind test data. Despite of the errors propagation in the beginning of discourse parsing pipeline, we still obtain improvements against baseline, and perform well on the supplementary tasks. Especially, the CNN model for Non-Explicit sense classification gives competitive performances. Actually, Non-Explicit sense classification performance can be furthermore improved in the future.

[1] http://www.code.google.com/p/word2vec

References

Deng Cai and Hai Zhao. 2016. Neural word segmentation learning for Chinese. In *Proceedings of ACL*, Berlin, Germany, August.

Changge Chen, Peilu Wang, and Hai Zhao. 2015. Shallow discourse parsing using constituent parsing tree. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 37–41, Beijing, China, July. Association for Computational Linguistics.

Christian Chiarcos and Niko Schenk. 2015. A minimalist approach to shallow discourse parsing and implicit relation recognition. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 42–49, Beijing, China, July. Association for Computational Linguistics.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Majid Laali, Elnaz Davoodi, and Leila Kosseim. 2015. The clac discourse parser at CoNLL-2015. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 56–60, Beijing, China, July. Association for Computational Linguistics.

Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing implicit discourse relations in the penn discourse treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 343–351. Association for Computational Linguistics.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A pdtb-styled end-to-end discourse parser. *Natural Language Engineering*, 20(02):151–184.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

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.

Tsuyoshi Okita, Longyue Wang, and Qun Liu. 2015. The dcu discourse parser: A sense classification task. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 71–77, Beijing, China, July. Association for Computational Linguistics.

Emily Pitler, Mridhula Raghupathy, Hena Mehta, Ani Nenkova, Alan Lee, and Aravind K Joshi. 2008. Easily identifiable discourse relations.

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.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The penn discourse treebank 2.0. In *LREC*. Citeseer.

Yangqiu Song, Haoruo Peng, Parisa Kordjamshidi, Mark Sammons, and Dan Roth. 2015. Improving a pipeline architecture for shallow discourse parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 78–83, Beijing, China, July. Association for Computational Linguistics.

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.

Jia Sun, Peijia Li, Weiqun Xu, and Yonghong Yan. 2015. A shallow discourse parsing system based on maximum entropy model. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 84–88, 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.

Rui Wang, Masao Utiyama, Isao Goto, Eiichro Sumita, Hai Zhao, and Bao-Liang Lu. 2013. Converting continuous-space language models into n-gram language models for statistical machine translation.

In *Proceedings of EMNLP*, pages 845–850, Seattle, Washington, USA, October.

Rui Wang, Hai Zhao, Bao-Liang Lu, Masao Utiyama, and Eiichiro Sumita. 2014. Neural network based bilingual language model growing for statistical machine translation. In *Proceedings of EMNLP*, pages 189–195, Doha, Qatar, October.

Peilu Wang, Yao Qian, Frank Soong, Lei He, and Hai Zhao. 2016. Learning distributed word representations for bidirectional lstm recurrent neural network. In *Proceedings of NAACL*, June.

Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Rashmi PrasadO Christopher Bryant, and Attapol T Rutherford. 2015. The CoNLL-2015 shared task on shallow discourse parsing. In *Proceedings of CoNLL*, page 2.

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.

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

Zhisong Zhang and Hai Zhao. 2016. Probabilistic graph-based dependency parsing with convolutional neural network. In *Proceedings of ACL*, Berlin, Germany, August.

Hai Zhao, Wenliang Chen, and Chunyu Kit. 2009a. Semantic dependency parsing of NomBank and PropBank: An efficient integrated approach via a large-scale feature selection. In *Proceedings of EMNLP*, pages 30–39, Singapore, August.

Hai Zhao, Wenliang Chen, Chunyu Kity, and Guodong Zhou. 2009b. Multilingual dependency learning: A huge feature engineering method to semantic dependency parsing. In *Proceedings of CoNLL*, pages 55–60, Boulder, Colorado, June.

Hai Zhao, Xiaotian Zhang, and Chunyu Kit. 2013. Integrative semantic dependency parsing via efficient large-scale feature selection. *Journal of Artificial Intelligence Research*, 46:203–233.

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