The DCU Discourse Parser for Connective, Argument Identification and Explicit Sense Classification

Longyue Wang, Chris Hokamp, Tsuyoshi Okita, Xiaojun Zhang, Qun Liu
ADAPT Centre
School of Computing, Dublin City University
Glasnevin, Dublin 9, Ireland
{lwang, chokamp, tokita, xzhang, qliu}@computing.dcu.ie

Abstract

This paper describes our submission to the CoNLL-2015 shared task on discourse parsing. We factor the pipeline into sub-components which are then used to form the final sequential architecture. Focusing on achieving good performance when inferring explicit discourse relations, we apply maximum entropy and recurrent neural networks to different sub-tasks such as connective identification, argument extraction, and sense classification. The our final system achieves 16.51%, 12.73% and 11.15% overall F1 scores on the dev, WSJ and blind test sets, respectively.

1 Introduction

The task of discourse parsing is generally conceived as a pipeline of steps, corresponding to: i) locating explicit discourse connectives, ii) identifying the spans of text that serve as the two arguments for each discourse connective, and iii) predicting the sense for both explicit and implicit relations. Understanding such discourse information is clearly an important component of natural language understanding that impacts a wide range of downstream natural language applications.

Since Penn Discourse Treebank was released, a number of data driven approaches have been proposed to deal with different challenging subtasks of discourse parsing. As explicit arguments may be intra-sentential or inter-sentential, Lin et al. (2012), Xu et al. (2012), Stepanov and Riccardi (2012) propose to employ argument position classification as heuristic and then apply separated models for argument extraction. Ghosh et al. (2011) regarded argument extraction as a token-level sequence labeling task, applying conditional random fields (CRFs) to label each token in a sentence. Following on this work, Ghosh et al. (2012) designed many global features to help distinguish Argument1 and Argument2 within the same sentence. Lin et al. (2014) formulated the task as finding the nodes in the constituent parse that are Argument1 or Argument2. However, the performance of this approach is heavily dependent upon the quality of the input parse trees. The different characteristic of implicit and explicit discourse relations are another important consideration. Lin et al. (2009) apply three feature classes: the constituent parse, the dependency parse and word-pair features for implicit relation classification. Rutherford and Xue (2014) exploit Brown cluster pairs to represent discourse relations in naturally occurring text. Considering the whole task, Lin et al. (2014) introduce a pipeline framework including several sub-tasks (connective classifier, argument labeler, explicit classifier and non-explicit classifier) to handle both explicit and non-explicit relations based on the PDTB corpus using maximum entropy.

In our work, we design the framework of our system based on Lin et al. (2014). The task is split the into seven components: connective classifier, argument positions classifier, three argument extractors, explicit sense classifier and implicit sense classifier. We approach argument extraction as a sequence labelling task, employing recurrent neural network (RNN) to classify each candidate token. We use distributional representations via word embeddings to decrease the out-of-vocabulary words (OOVs) problem which result from the scarcity of training data. After a post-precessing step which resolves label conflicts, we extract the spans of arguments. For other components, we use a classification via maximum entropy, and explore diverse features. In this system, we mainly focus on explicit relations, thus we only apply a simple majority function for the non-explicit component.

The remainder of this paper is organized as fol-

Proceedings of the Nineteenth Conference on Computational Natural Language Learning: Shared Task, pages 89–94,
Beijing, China, July 26-31, 2015.

lows: Section 2 describes the framework and each component of our proposed system. Then we discuss the results, including the official results and post-task results, in Section 3. Finally, we summarize our conclusions in Section 4.

2 Proposed System

The framework of our system is shown in Figure 1. In the first step, the connective classifier is used to identify connectives according to the occurrences of the predefined connectives. Once a candidate is labelled as a connective, an explicit relation is created. The next step is then to find the argument positions (*arg1* and *arg2*) for each explicit relation. Here we use a classifier to label two cases: 1, *arg1* and *arg2* are in the same sentence (SS), or 2, *arg1* and *arg2* are not in the same sentence (OT). Then we train and apply different argument extraction models for these two cases. After labelling the argument span, we use a sense classification component to classify them to predefined sense types.

After processing the explicit relations, the non-explicit part extracts all the adjacent sentence pairs which are not explicit relations and then infers implicit relations. As we mainly focus on explicit relations, in this part, we only apply a simple majority function to give all candidate pairs the same results.

2.1 Connective Classifier

As words which can be discourse connectives do not always function as discourse connectives, we need to identify if an instance of a connective candidate is a functional connective each time it occurs. Pilter and Nenkova (2009) showed that syntactic features extracted from constituent parse trees are very useful in disambiguating discourse connectives from other functions. Lin et al. (2014) tackled this problem by first using the connective list to identify the candidates and then using a combination of simple POS-based features and tree-based features, an approach which also achieved good performance. To model the syntactic relation, they also propose a path feature, which is the combined tags of sub-tree nodes from connective to the root. Compressed path means the adjacent identical tags are combined (e.g., -NP-NP- is combined into -NP-).

Based on above work, we extract the 99 types of connectives defined in the PDTB training corpus. As shown in Table 1, we use three feature classes: lexical, syntactic and others. Especially, we employ the position of connection as a new feature (i.e., beginning or not), because we observe that the candidates occurring at the beginning are always the connectives. Then a ME model is applied to classify each connective candidate as a connective or not. After exploring 14 features and combinations, we finally found that the feature set {2-10,13, 14} which yields the best performance on dev set. The final score is shown in Section 3.

2.2 Argument Position Classification

arg2 is the argument with which the connective is syntactically associated, and its position is fixed once we have located the connective from the previous component (Section 2.1). Thus, the challenging step for this task is to identify the location of *arg1*.

Prasad et al. (2008) show that *arg1* may be located in various positions to the connective, such as within the same sentence (SS), before (PS), or after (FS) the sentence containing the connective. Furthermore, *arg1* may be adjacent or non-adjacent with connective sentence. *arg1* may also contain one or more sentences. Table 2 shows the statistics of each of above scenarios.

Relative Position	1 Sent	n Sents
SS	60.38%	-
FS	0.01%	0.03%
PS	27.93%	1.89%
Other Scenarios	9.79%	

Table 2: Statistics of *arg1*'s Positions. (Percentage (%) is computed as the number of the scenario divided by the total relations; n>1)

As SS and PS constitute 90.20% of all explicit relations, our system mainly focus on these two cases. Therefore, we use a argument position classifier to classify a relation as SS or PS. In our experiment, we compared 17 features and their combinations, which are shown in Table 3. Finally, we use the feature set {1-3, 5, 7, 9, 11-14, 17} since it achieves the highest accuracy (97.78%) on dev set.

2.3 Argument Extraction

One of the key problems in discourse parsing is the task of extraction of argument spans of discourse relation. In the light of the recent success

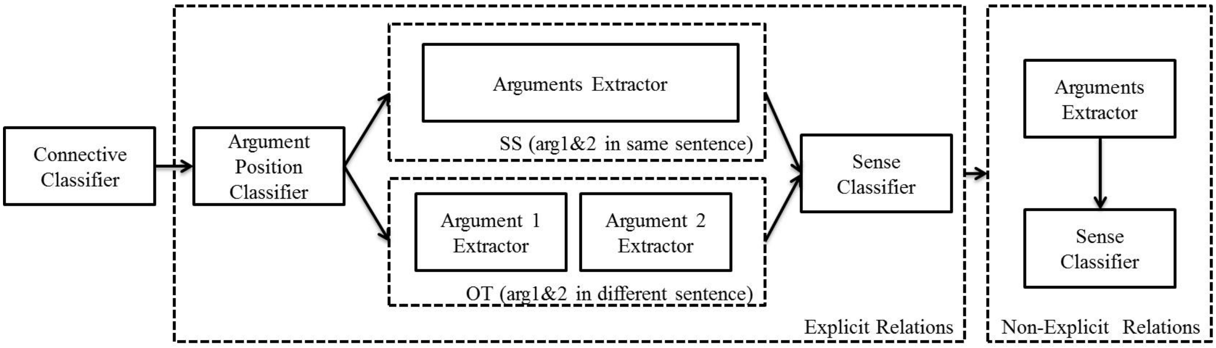

Figure 1: Framework of Our System

Type	ID	Features
Lexical Features	1	Connective Word
	2	Connective POS
	3	1st Previous Word of Connective
	4	1st Next Word of Connective
	5	1st Previous Word + Connective Word
	6	Connective Word + 1st Next Word
	7	1st Previous POS + Connective POS
	8	Connective POS + 1st Next POS
	9	1st Previous Word + Connective Word + 1st Next Word
	10	1st Previous POS + Connective POS + 1st Next POS
Syntactic Features	11	Path of Connective to the Root
	12	Path of Connective's Parent to the Root
	13	Compressed Path of Connective's Parent to the Root
Others	14	Low-Cased Connective Word

Table 1: Features for Connective Classification

Type	ID	Features
Lexical Features	1	Connective Word
	2	Connective POS
	3	1st Previous Word of Connective
	4	1st Next Word of Connective
	5	1st Previous POS of Connective
	6	1st Next POS of Connective
	7	1st Previous Word + Connective Word
	8	Connective Word + 1st Next Word
	9	1st Previous POS + Connective POS
	10	Connective POS + 1st Next POS
	11	2nd Previous POS of Connective
	12	2nd Previous Word of Connective
	13	2nd Previous POS + Connective POS
	14	2nd Previous Word + Connective Word
	15	1st Previous Word + Connective Word + 1st Next Word
	16	1st Previous POS + Connective POS + 1st Next POS
Others	17	Position of Connective

Table 3: Features for Argument Position Classification

of applying deep neural network technologies in natural language processing, we carried out an investigation of the use of recurrent neural network (RNN) for this difficult task (Mesnil et al., 2013; Raymond and Riccardi, 2007).

After determining the likely position of *arg1*, we split the explicit relations into two sets: SS and OT. We apply token-level sequence labeling approach with the separate models for arguments of intra-sentential and inter-sentential explicit discourse relations (Ghosh et al. 2011; Stepanov and Riccardi, 2012). As shown in Figure 1, we apply two components to deal with these two cases. Besides, in OT, we also train separated models to deal with Arg1 and Arg2 extraction.

Since for sequence labeling we use IOBE (Inside, Out, Begin, End) notation as the labels for both Arg1 and Arg2. For example, the set of classes for the SS case is {arg1-B, arg1-I, arg1-E, arg2-B, arg2-I, arg2-E and None}. The sets of classes for OT are {arg1-B, arg1-I, arg1-E and None} and {arg2-B, arg2-I, arg2-E and None}.

As input features, we use the word embeddings for Arg1 and Arg2 in order to infer the argument labels. We use RNNs to learn a word embedding on the part of training data. As the official scorer will give points only when the whole argument span is right, we employ this scorer to calculate the performance in each iteration of training. Furthermore, we compare the performance with different parameters: number of context windows, hidden layers, iterations and word embeddings. Finally, we set number of context windows as 5, hidden layers as 300, iterations as 10 and word embeddings as 100 to achieve the highest performance.

Besides, we only extract the relations in the corresponding scenario as the training data, thus OOVs may harm the models. We use distributional representations via word embeddings to alleviate the problem, which results from the scarcity of training data.

2.4 Explicit Sense Classification

One method that has previously been employed to resolve the ambiguity in discourse connectives is to build a classifier with some very simple features. They are the connective (one or more words), the connectives POS, and the connective + its previous word (Lin et al., 2014). This approach achieves an F1 score of 86.77, which is quite impressive compared the human agreement score of 84%.

Therefore, for this component, we still employ the similar feature set, which is shown in Table 4. Finally, we apply the feature set {1-3, 5-6} to obtain the best scores on dev set.

2.5 Non-Explicit Relations

The non-explicit relation includes Implicit , AltLex, EntRel and NoRel relations.

The non-explicit relations are annotated for all adjacent sentence pairs within paragraphs. If there is already an explicit relation from the previous step between two adjacent sentences, they are exempt from this step. In our system, we just apply a majority classifier, labeling all non-explicit relation candidates as EntRel.

3 Experiments and Results

3.1 System Setup

All available training data, development set, test sets from CoNLL 2015 (LDC2015E21)[1] are used in this study. Besides, we use the Skip-gram Neural Word Embeddings[2] for RNNs. All the used syntactic information are automatically predicted by the Berkeley Parser[3].

We use Maxent toolkit[4] for the ME method. And we apply Theano[5] (Bastien et al., 2012; Bergstra et al., 2010) for the RNNs. We use the Python programming language to develop all the compontents and divided each component into two parts: one is training which is processed in our CPU and GPU servers and the other is decoding which is run on TIRA server[6].

3.2 Official Results

The official results are shown in Table 5. The performance of connective classifier is around 80%, which is not good enough. There are two reasons: 1, we skip some separated connectives such as either or, neither nor etc. and 2, the current feature set missed some syntactic information. For argument extraction, the reasonable scores show our proposed method can really work for this part. However, it does not work well for OT case, because the span is always located the whole sentence. It may be helpful by adding structure fea-

[1] Available at https://www.ldc.upenn.edu

[2] Available at https://code.google.com/p/word2vec

[3] Description at http://www.cs.brandeis.edu/ clp/conll15st/rules.html

[4] Available at https://github.com/lzhang10/maxent

[5] Available at http://deeplearning.net/tutorial/rnnslu.html

[6] Available at http://www.tira.io

Type of Feature	ID	Features
Lexical Features	1	Connective Word
	2	Connective POS
	3	Connective + 1st Previous Word
	4	Connective + 2st Previous Word
	5	Connective + 1st Previous POS
Others	6	Low-Cased Connective

Table 4: Features for Explicit Sense Classification.

tures into RNNs. The sense classifier is the worst component, which only obtained about 8% F1 scores. It is because 1, the errors from previous components are propagated, which is also the limitation of the pipeline architecture; 2, we apply a simple non-explicit component and miss a lot implicit relations, which result in the low recall. On the whole, our system can still be improved in many ways.

4 Conclusions and Further Work

This paper describes the discourse parsing system we implemented for the CoNLL-2015 shared task. We build a pipeline system which focuses on achieving good performance when inferring explicit discourse relations. We apply maximum entropy and recurrent neural networks to different sub-tasks.

This is our ongoing work, and we will keep on improving the system by employing novel neural network methods.

Acknowledgments

This work is supported by the Science Foundation of Ireland (SFI) CNGL project (Grant No.: 12/CE/I2267), and partly supported by the DCU-Huawei Joint Project (Grant No.:201504032) and the Open Projects Program of National Laboratory of Patter Recognition (Grant No.: 201407353), and also by the European Commission FP7 EXPERT project.

References

Ziheng Lin, Hwee Tou Ng and Min-Yen Kan. 2014. *A PDTB-styled End-to-End Discourse Parser*, volume 1-34. Natural Language Engineering.

Sucheta Ghosh, Richard Johansson and Sara Tonelli. 2011. *Shallow Discourse Parsing with Conditional Random Fields*. In Proceedings of the 5th International Joint Conference on Natural Language Processing.

Sucheta Ghosh, Giuseppe Riccardi and Richard Johansson. 2012. *Global Features for Shallow Discourse Parsing*, 150-159. In Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue. Association for Computational Linguistics

Emily Pitler and Ani Nenkova. 2009. *Global Features for Shallow Discourse Parsing*, 150-159. In Proceedings of the ACL-IJCNLP 2009 Conference Short Papers. Association for Computational Linguistics.

Attapol T. Rutherford and Nianwen Xue. 2014. *Discovering Implicit Discourse Relations Through Brown Cluster Pair Representation and Coreference Patterns*, 645. In Proceedings of EACL 2014.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, Bonnie Webber. 2008. *The Penn Discourse TreeBank 2.0*. In Proceedings of the LREC.

Grgoire Mesnil, Xiaodong He, Li Deng and Yoshua Bengio. 2013. *Investigation of Recurrent-Neural-Network Architectures and Learning Methods for Spoken Language Understanding*. In Proceedings of the Interspeech 2013.

Christian Raymond and Giuseppe Riccardi. 2007. *Generative and discriminative algorithms for spoken language understanding*. In Proceedings of the Interspeech 2007.

Frdric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, Yoshua Bengio. 2012. *Theano: new features and speed improvements*. In Proceedings of the Python for Scientific Computing Conference (SciPy).

James Bergstra, Olivier Breuleux, Frdric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, Yoshua Bengio. 2010. *Theano: a CPU and GPU math expression compiler*. In Proceedings of the Python for Scientific Computing Conference (SciPy).

Evgeny A. Stepanov and Giuseppe Riccardi. 2013. *Comparative evaluation of argument extraction algorithms in discourse relation parsing*. In Proceedings of 13th International Conference on Parsing Technologies (IWPT 2013).

	Dev Set			Test Set			Blind Set		
Components	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Connectives	0.9010	0.8162	0.8565	0.9040	0.8570	0.8799	0.8487	0.7464	0.7943
Arg1	0.3437	0.4770	0.3995	0.3100	0.4384	0.3632	0.2794	0.3755	0.3204
Arg2	0.3778	0.5244	0.4392	0.3559	0.5034	0.4170	0.3489	0.4690	0.4001
Arg1 & Arg2	0.2559	0.3552	0.2975	0.2174	0.3074	0.2546	0.1926	0.2589	0.2209
Sense	0.3194	0.1080	0.0938	0.2257	0.1124	0.0849	0.0905	0.0701	0.0481
Overall	0.1420	0.1971	0.1651	0.1087	0.1537	0.1273	0.0972	0.1307	0.1115

Table 5: Official Results.

Xu Ming, Zhu Qiao and Zhou Guo Dong. 2012. *A Unified Framework for Discourse Argument Identification via Shallow Semantic Parsing*. In Proceedings of 24th International Conference on Computational Linguistics.