Implicit Discourse Relation Recognition with Context-aware Character-enhanced Embeddings

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

For the task of implicit discourse relation recognition, traditional models utilizing manual features can suffer from data sparsity problem. Neural models provide a solution with distributed representations, which could encode the latent semantic information, and are suitable for recognizing semantic relations between argument pairs. However, conventional vector representations usually adopt embeddings at the word level and cannot well handle the rare word problem without carefully considering morphological information at character level. Moreover, embeddings are assigned to individual words independently, which lacks of the crucial contextual information. This paper proposes a neural model utilizing context-aware character-enhanced embeddings to alleviate the drawbacks of the current word level representation. Our experiments show that the enhanced embeddings work well and the proposed model obtains state-of-the-art results.

1 Introduction

It is widely agreed that in a formal text, units including clauses and sentences are not isolated but instead connected logically, semantically, and syntactically. Discourse parsing is a fundamental task in natural language processing (NLP) that analyzes the latent relation structure and discovers those connections across text units. It could benefit various downstream NLP applications such as question answering (Chai and Jin, 2004; Verberne et al., 2007), machine translation (Hardmeier, 2012; Guzmán et al., 2014), sentiment analysis (Bhatia et al., 2015; Hu et al., 2016b), and automatic summarization (Maskey and Hirschberg, 2005; Murray et al., 2006).

For discourse parsing, Penn Discourse Treebank (PDTB) (Prasad et al., 2008) provides the lexically-grounded annotations of discourse relations. Each discourse relation consists of two abstract object arguments and the corresponding sense annotations, which can be roughly characterized according to whether explicit connectives could be drawn from the texts. In *Explicit* relations, explicit connectives can be found in the texts; when such indicators are not given directly, an inferred connective expression could be inserted, forming *Implicit* relations. The following two examples describes these two kinds of discourse relations: the former has an explicit connective *"so"* which reveals the *Explicit* relation, while in the latter case, an inferred *Implicit* connective *"that is"* has to be inserted to express the relation.

(1) **Arg1:** We're standing in gasoline.
Arg2: So don't smoke. (Contingency.Cause.Result - wsj_0596)

(2) **Arg1:** The ploy worked.
Arg2: Implicit=that is The defense won. (Contingency.Cause - wsj_1267)

It has been shown that discourse connective is crucial for high-accuracy relation recognition (Pitler et al., 2009; Lin et al., 2014). Compared to explicit discourse relations in which senses between adjacent

*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, No. 61672343 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 COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers,
pages 1914–1924, Osaka, Japan, December 11-17 2016.

clauses are effectively indicated by explicit connectives like *"but"* and *"so"*, implicit discourse relation recognition is much more difficult. Without effective indicators, the relations could only be inferred from indirect plain texts, which makes implicit discourse relation recognition the bottleneck of the entire discourse parsing system (Qin et al., 2016a; Li et al., 2016; Chen et al., 2015). This paper attempts to deal with this challenging task.

The challenge stems from the fact that, without connective cues, recognizing implicit relation has to rely solely on two textual arguments, and it must capture latent semantic and logical relationship between two arguments in discourse-level. First, given limited amount of annotated corpus, both the traditional indicator feature based methods and the recent embedding based neural methods (Wang et al., 2015) can suffer from insufficient data. The training is especially difficult for rare words, which appear rarely in the corpus but generally take up a large share of the dictionary, making it hard to effectively learn their representations, resulting in high perplexities for discourse relation recognition. Moreover, implicit relation recognition calls for semantic understanding, which needs to encode the word meaning in the context and the sentence-level understanding for the argument pairs. Considering the complexity of natural language, the task is quite nontrivial and requires more effective encoding of the arguments.

Conventional methods for implicit discourse relation recognition are based on manually specified indicator features, such as bag-of-words, production rules, and other linguistically-informed features (Zhou et al., 2010; Park and Cardie, 2012; Biran and McKeown, 2013; Rutherford and Xue, 2014). Recently, embedding based neural models have been proved effective to address the data sparsity problem that is not well solved in traditional methods. The key techniques include real-valued dense embeddings for feature representations and non-linear neural models for feature combinations and transformations. However, most of the neural models take words as the smallest processing units, which can suffer a lot from insufficient training on rare words. Discourse parsing, as the highest level language processing at present, covers word and sentence levels for feature representation. This work extends the current word-level representation onto more fine-grained character-level which is helpful for encoding morphology information and alleviating the rare word problem.

In summary, this paper presents a neural model with context-aware character-enhanced embeddings to address implicit discourse relation recognition task. Recently, character-aware models have been popular for English and other morphologically rich languages (Kim et al., 2016; Zhang et al., 2015b; Ling et al., 2015). The proposed model enhanced the word embeddings with character-aware representations learned from stacked convolutional and recurrent neural models. Utilizing these enhanced embeddings, the model covers information of three levels from character, word, to sentence. Through extensive experiments on standard discourse corpus, we analyze several models and show the superiority of the proposed method.

The remaining of the paper is organized as follows: Section 2 discusses related work; Section 3 describes the proposed model; Section 4 provides the details of experiments and model analysis; and Section 5 concludes the paper.

2 Related work

Implicit discourse relation recognition is the subcomponent of the end-to-end discourse parsing system, which is also used as the share-task in CoNLL 2015 and CoNLL 2016 (Xue et al., 2015; Xue et al., 2016). In the share-task, the classification task concerns other Non-Explicit types including *EntRel* and *AltLex*, in addition to the *Implicit* relations.

Early work for implicit discourse relation recognition focuses on typical machine learning solutions with sparse indicator features and linear models. Pitler et al. (2009) use several linguistically informed features, including polarity tags, Levin verb classes and length of verb phrases. Zhou et al (2010) improve the performance through predicting connective words as extra features. Park and Cardie (2012) propose a method using a locally-optimal feature set. Biran and McKeown (2013) collect word pairs from arguments of explicit examples to help the learning. Rutherford and Xue (2014) employ Brown cluster pairs to represent discourse relation and incorporate coreference patterns to identify the meaning in text. Li and Nenkova (2014) introduce a syntactic representation to reduce sparsity. Rutherford and

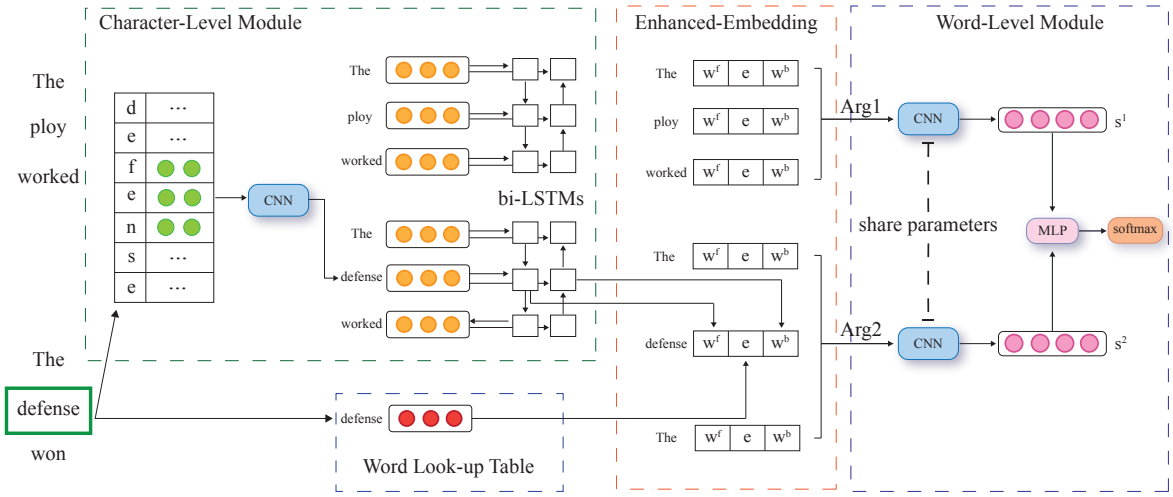

Figure 1: Architecture of the proposed model.

Xue (2015) and Ji et al. (2015) add automatically-labeled instances to expand data. Fisher and Simmons (2015) incorporate a mixture of labeled and unlabeled data to reduce the need for annotated data.

More recently, neural network models have been proved effective for NLP tasks (Wang et al., 2016; Zhang et al., 2016; Cai and Zhao, 2016; Hu et al., 2016a) and also utilized for implicit discourse relation recognition. Ji and Eisenstein (2015) adopt recursive neural network and incorporated with entity-augmented distributed semantics. Zhang et al. (2015a) propose a simplified neural network which contains only one hidden layer and use three different pooling operations (max, min, average). Chen et al. (2016) adopt a deep gated neural model to capture the semantic interactions between argument pairs. Ji et al. (2016) propose a latent variable recurrent neural network architecture for jointly modeling sequences of words. (Qin et al., 2016b) propose a stacking neural network model to solve the classification problem. In their model, convolutional neural network is utilized for sentence modeling and a collaborative gated neural network is proposed for feature transformation.

3 Model

3.1 Architecture

The architecture of our model is shown in Figure 1. The model is a hybrid neural network including a character-level module and a word-level module. The character-level module receives inputs of character-level embeddings followed by stacked CNN and bidirectional LSTMs layers. First, the convolutional and max-pooling operations perform local information encoding and feature selection, obtaining a fixed-dimensional representation of the character-based word representation sequence. Then via bidirectional LSTMs, the sequence is transformed to a new sequence which encodes the rich contextual information. This new sequence is the output of the character-level module which models context-aware character-level information, and will be utilized in later layers. In the word-level module, the character-based word representations will be concatenated to ordinary word embeddings, forming enhanced embeddings which integrate both character-level and word-level information. Later CNN will be utilized again to obtain sentence-level representations for the two arguments, followed by conventional hidden layers and a softmax layer for the final classification.

3.2 Character-Level Module

This module aims to get the most out of the character sequence and obtain the character-based word representations, utilizing an architecture of stacked CNN and LSTM. Modeling from character level could alleviate rare words problem and useful capture morphological information, like the prefixes and suffixes of words.

Character Embedding The character representation will still be in the form of embeddings. In this task, we define an alphabet of characters which contains uppercase and lowercase letter as well as numbers and punctuation. The input word will be decomposed into a character sequence. Through a character look-up table, a word will be projected to a sequence of character vectors: $[\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n]$, where $\mathbf{c}_i \in \mathbb{R}^{d_c}$ is the vector for the i-th character in the word with dimension d_c and n is word length. For the convenience of notation, the character vectors for a word can be regarded as a character matrix $\mathbf{C}$:

$$\mathbf{C} = [\mathbf{c}_1; \mathbf{c}_2; \ldots; \mathbf{c}_n]$$

Convolutional Neural Network A convolutional operation followed by a max-pooling operation will be applied to the character matrix $\mathbf{C}$ of each word. The convolutional layer is used to extract and combine local features from adjacent characters and the following max-pooling layer forms the representations for the current word. For the convolutional operation, k groups of filter matrices [$\mathbf{F}_1$, $\mathbf{F}_2$, ..., $\mathbf{F}_k$] with variable sizes $[l_1, l_2, \ldots, l_k]$ and biases [$\mathbf{b}_1$, $\mathbf{b}_2$, ..., $\mathbf{b}_k$] are utilized. Each of them transforms the character matrix $\mathbf{C}$ to another sequence. The transformed sequences $\mathbf{C}'_j (j \in [1, k])$ will be obtained as follows:

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

Here, i indexes the convolutional window. Next, a one-max-pooling operation is adopted and the representation $\mathbf{w}$ for a word is obtained through concatenating all the mappings after pooling as follows:

$$\mathbf{w}'_j = \mathbf{max}(\mathbf{C}'_j)$$
$$\mathbf{w} = [\mathbf{w}'_1 \oplus \mathbf{w}'_2 \oplus \ldots \mathbf{w}'_k]$$

Bidirectional LSTM The character-based word representation obtained through CNN can be directly utilized in the word-level module, however, each word vector from the CNN is individually obtained and lacks of the encoding of contextual information. In a sentence, word can never be understood independently without context. Nearby words can offer important cues to the current word as suggested by N-gram language model and context-aware sentence modeling. Motivated by this, we propose to utilize bidirectional LSTMs to encode the character-based word vectors.

Given the character-based word representations $[\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n]$ as the input sequence, an LSTM computes the state sequence $[\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]$ by applying the following formulation for each time step:

$$\mathbf{i}_t = \sigma(\mathbf{W}^i_w \mathbf{w}_t + \mathbf{W}^i_h \mathbf{h}_{t-1} + \mathbf{W}^i_c \mathbf{w}_{t-1} + \mathbf{b}_i)$$
$$\mathbf{f}_t = \sigma(\mathbf{W}^f_w \mathbf{w}_t + \mathbf{W}^f_h \mathbf{h}_{t-1} + \mathbf{W}^f_c \mathbf{w}_{t-1} + \mathbf{b}_f)$$
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot tanh(\mathbf{W}^c_w \mathbf{w}_t + \mathbf{W}^c_h \mathbf{h}_{t-1} + \mathbf{b}_c)$$
$$\mathbf{o}_t = \sigma(\mathbf{W}^o_w \mathbf{w}_t + \mathbf{W}^o_h \mathbf{h}_{t-1} + \mathbf{b}_o)$$
$$\mathbf{h}_t = \tanh(\mathbf{c}_t) \odot \mathbf{o}_t$$

Here the σ denotes the sigmoid function and the $\odot$ denotes element-wise multiplication. $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{c}_t$, $\mathbf{o}_t$ and $\mathbf{h}_t$ stand for input gate, forget gate, memory cells, output gate and the current state, respectively. Finally, the state sequence $[\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]$ will be utilized as the context-aware word representations. Recent works (Graves et al., 2013; Graves et al., 2005) show that backward LSTM can also effectively encode the context by modeling the word sequence backward, combined with the ordinary forward LSTM, the so-called Bidirectional LSTM could effectively capture the information from both past and future words, and we will utilize it in this module. We will denote the output state sequence (originally noted as $\mathbf{h}$) of forward LSTM as $[\mathbf{w}^f_1, \mathbf{w}^f_2, \ldots, \mathbf{w}^f_n]$ and the one of the backward LSTM as $[\mathbf{w}^b_1, \mathbf{w}^b_2, \ldots, \mathbf{w}^b_n]$.

3.3 Word-Level Module

In recent neural models, words are represented as real-valued dense vectors. With the prevalence of deep learning methods in NLP, continuous space word vectors have been found an effective means for word

representations. Unlike the traditional model in which words usually represent as one-hot vectors and are independent with each other, vector space models reveal the relationship and capture the intuition among words which different or similar to others along a variety of dimensions (Mikolov et al., 2013). However, embeddings considering only at word level is usually not good for rare words as discussed above and we introduce character-level embedding to enhance the current word embedding.

Enhanced Word Embedding For obtaining word representations, we enhance ordinary word vectors with the character-based vectors obtained from the character-level module by concatenating all the representations on words. Thus a sequence of enhanced word embeddings could be obtained, which could cover contextual information from character-level to word-level, and the character-based embedding could alleviate rare word problems in some way. Formally speaking, an argument could be represented as a sequence as follows:

$$\mathbf{M} = [\mathbf{w}_1^f \oplus \mathbf{e}_1 \oplus \mathbf{w}_1^b; \mathbf{w}_2^f \oplus \mathbf{e}_2 \oplus \mathbf{w}_2^b; \ldots; \mathbf{w}_n^f \oplus \mathbf{e}_n \oplus \mathbf{w}_n^b]$$

where $\oplus$ is the concatenation operator. $\mathbf{w}^f, \mathbf{e}, \mathbf{w}^b$ stand for the state of forward LSTM, word embedding and the state of backward LSTM, respectively.

Convolutional Neural Network In the word-level module, CNN is utilized again to extract local context features. Like the convolutional layer in character-level module, several groups of filter matrices with various filter window sizes are utilized to extract features from different ranges. This procedure is quite similar to the one in character-level module and we will leave out the formulas. Unlike the character-level CNN, here the convolutional operation is applied on the arguments and the following max-pooling layer will produce the sentence vectors. Via parameter sharing, this feature extraction procedure become same for both arguments. We will note the sentence vectors for the two arguments as $\mathbf{s}^1$ and $\mathbf{s}^2$.

Softmax Getting the sentence-level representations, we can concatenate the sentence vectors and feed them to the conventional softmax layer for the final classification.

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

$$\Pr(y_i) = \frac{\exp \mathbf{w}^i \times \mathbf{v}}{\sum_j^l \exp \mathbf{w}^j \times \mathbf{v}}$$

Here, $\Pr(y_i)$ means the probability of assigning the instance to label i, $\mathbf{w}$ indicates the parameters in the final softmax layer. Additionally, multilayer perceptron (MLP) hidden layers could be added between sentence vectors and the final softmax layer, and we will leave out the descriptions for brevity.

3.4 Training

For training, the object is the cross-entropy error 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 $y^{(k)}$ is the gold labels and $\hat{y}^{(k)}$ is the predicted ones. For the optimization process, we apply the diagonal variant of AdaGrad (Duchi et al., 2011) with mini-batches.

4 Experiment

PDTB 2.0[1], which is one of the largest manually annotated corpus of discourse relation, is utilized for the experiments. Annotated on Wall Street Journal corpus with one million words, the data contain

[1] http://www.seas.upenn.edu/ pdtb/

16,224 implicit relations. It provides three hierarchies of relations: Level 1 *Class*, Level 2 *Type*, and Level 3 *Subtypes*. The first level consists of four major relation *Class*: COMPARISON, CONTINGENCY, EXPANSION and TEMPORAL. There are 16 Level 2 relation types of implicit relations. The third level of *Subtypes* is types that are only available for specific types.

For the evaluation of implicit relation classification, there are two settings in previous works: one is multi-class classification for second-level discourse relations (Lin et al., 2009); the other is the "One-Versus-Others" setting which employs binary classification only for Level 1 $Class$, which is first used by Pitler et at. (2009). Note that the results for the latter setting can be also derived from the specific statistics over the results of the former setting. In this paper, we will focus on the more practical multi-class classification, which is a necessary component for building a complete discourse parser such as that for the shared tasks of CoNLL-2015 and 2016 (Xue et al., 2015; Xue et al., 2016). For the model analysis, we perform the experiments with the multi-classification setting. In order to compare with previous results, we will also evaluate our system on the binary relation classification task.

4.1 Multi-class classification

Following (Lin et al., 2009), we adopt the standard PDTB splittings as follows: Sections 2-21 as training set, Section 22 as development set and Section 23 as test set. We will denote this dataset as the PDTB Standard setting *PDTB-STD*. In order to be in consistence with previous setting, we also remove 5 *Types* which are too few in the corpus: CONDITION, PRAGMATIC CONCESSION, PRAGMATIC CONCESSION, PRAGMATIC CONTRAST and EXCEPTION. Thus, we use the remaining 11 Level 2 *Types* in our experiments. In addition, for nearly 2% of the implicit relations have more than one type during annotating in PDTB, we consider these relations as two relation types with the same argument pairs when training. During testing, the predictions which match one of the gold types will be considered as correct. To compare with the state-of-the-art system (Ji and Eisenstein, 2015), which uses a slightly different setting: Sections 2-20 as training set, 0-1 as development set, and 21-22 as testing set. We also run experiments on this setting (noted as the PDTB Alternative setting *PDTB-ALT*) and show the comparisons.

4.1.1 Hyper-Parameters

For the hyper-parameters of the model and training process, we fix the lengths of both arguments (number of words) to be 80 and the lengths of the words (number of characters) to be 20, and apply truncating or zero-padding when necessary. The dimensions for character embeddings and word embeddings are 30 and 300 respectively. The word embeddings are initialized with pre-trained word vectors using *word2vec*[2] (Mikolov et al., 2013) and other parameters are randomly initialized by sampling from uniform distribution in [-0.5, 0.5] including character embeddings. The learning rate is set as 0.002.

In the character-level module, the CNN part uses three groups of 128 filters, with filter window sizes of (2, 3, 4); while the output dimensions of bidirectional LSTMs is set to 50. In the word-level module, the CNN part also contains three groups of filters. For we need more parameters to accurately model the sentence level information, each group has 1024 filters and their filter window sizes are (2, 4, 8). We also add another hidden layer above the concatenated sentence vectors and its dimension is set to 100.

4.1.2 Models

In this sub-section, we will describe the models in the comparisons of our main experiments, which show the effectiveness of the proposed neural model with enhanced embeddings. Our experiments mainly concerns four group of models: Baseline Models, Word-level Only Neural Models, Character-level Only Neural Models and Combined Models. The proposed model, namely **Char+Word-Enhanced**, falls into the last group and many other models can be considered as partial models of it.

Baseline Models These include simplified models or previous traditional model.

- **Majority Baseline** The most commom *Type* class is CAUSE, which accounts for 26.1% of the implicit relations in the PDTB test set.

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

Model	Accuracy
Majority Baseline	26.10
Word Representation	34.07
Lin et al. (2009)	40.20
Word-BiLSTMs	33.42
Word-CNN	41.12
Char-CNN	30.15
Char-[CNN+BiLSTMs]	34.86
Char+Word-Concat	42.55
Char+Word-Enhanced	**43.81**

Table 1: Comparisons on test set of *PDTB-STD* for multi-class classification.

Model	Accuracy
Majority Baseline	26.03
Word Representation	36.86
Lin et al. (2009)	-
+Brown clusters	40.66
Ji and Eisenstein (2015)	36.98
+Entity semantics	37.63
+Surface features	43.75
+both	44.59
Char+Word-Enhanced	**45.04**

Table 2: Comparisons on test set of *PDTB-ALT* for multi-class classification.

- **Word Representation** This model just utilizes sum of word vector as sentence vectors, for showing how the model with only word vector embeddings can work.
- **Lin et al. (2009)** Traditional linear model with manually specified features, including production rules, dependency rules, word pairs and context features.

Word-level Models These models only utilize conventional word vectors through a word-level embedding table looking-up process and does not use the character-level module.

- **Word-level CNN** This model adopts CNN with conventional word embeddings, which does not utilize the character-level module.
- **Word-level BiLSTMs** This model replaces CNN to Bidirectional LSTMs, also with conventional word embeddings. The sentence vectors will be the last state vector of the LSTMs.

Character-level Models These models only make use of the word representations learned from the Character-level module. For the word-level module (from word-level representations to sentence-level ones), CNN will be used.

- **Char-level CNN** This model utilizes only the embeddings from character-level module (without concatenating the word-level embeddings), and in the character-level part BiLSTMs are not utilized and the word representations are directly from CNN.
- **Char-level CNN+BiLSTMs** This model integrates BiLSTMs in the character-level module, which could encode the context information in the character-level embeddings.

Combined Models These combine the Char-level and Word-level modules (**Char-level CNN+BiLSTMs** and **Word-level CNN**) through concatenation on different levels.

- **Char+Word-Concat** This model combines the two modules at the sentence representation level, by concatenating the sentence vectors.
- **Char+Word-Enhanced** This is the proposed model, which combines the modules at the word embedding level, forming enhanced embeddings.

4.1.3 Model Analysis

The analysis of the models will be based on the results of Table 1 and we will discuss them in groups. First, the traditional linear model performs well, but it needs manually specified features. Simply adding word vectors is not a very good idea, because it ignores the crucial information of word order. In the second group, we could see that CNN performs well, for it provides the capacity of modeling local word sequences (via convolutional operations) and capturing sentence-level features (via max-pooling operations). Somewhat surprisingly, the model of BiLSTMs seems not good for this sentence-pair modeling task, the reason might be that using the last states of LSTMs ignores too much information of the

Competitive System	COMP.	CONT.	EXP.+	TEMP.	AVG.
Pitler et al. (2009)	21.96	47.13	76.42	16.76	40.57
Zhou et al. (2010)	31.79	47.16	70.11	20.30	40.32
Park and Cardie (2012)	31.32	49.82	79.22	26.57	46.73
McKeown and Biran (2013)	25.40	46.94	75.87	20.23	42.11
R&Xue (2014)	39.70	54.42	80.44	28.69	50.81
Ji and Eisenstein (2015)	35.93	52.78	80.02	27.63	49.09
Braud (2015)	36.36	**55.76**	61.76	29.30	45.80
Zhang et al.(2015a)	33.22	52.04	-	30.54	-
Chen et al. (2016)	**40.17**	54.76	80.62	31.32	51.72
Char+Word-Enhanced	38.67	54.91	**80.66**	**32.76**	**51.75**

Table 3: Comparisons of F_1 scores (%) for binary classification. (symbol **+** means EXP. with *Entrel*)

previous words of the sequence. Thus, for the rest models, CNN will be selected to compute sentence vectors. In the third group, we will explore how the character-based embeddings will perform without conventional word-level embeddings. The character-based embeddings learned from **Char-CNN** model are individually calculated and lacks of the information of surrounding words, thus stacking BiLSTMs improves the accuracies because the recurrent layer could effectively capture rich context characteristics. Not surprisingly, utilizing only character-level embedding performs not that good, even the simple adding-word-vectors method gives better accuracies. This suggests that conventional word-level embeddings should not be abandoned because a word is only meaningful at the word-level. However, the character forming of a word could be also helpful, especially when we are dealing with rare words. Thus in the fourth group, we will explore the combination of character-level and word-level representations. The **Char+Word-Concat** model that concatenates the sentence vectors (from different CNNs) indeed improves the performance. The proposed model, **Char+Word-Enhanced**, combines the two modules at the word representation level, this is different from the **Concat** model because the influence of character-level representations are directly integrated into the final word representations before fed to CNN. The proposed model outperforms all the others, which shows the character-based representations do make extra helps.

4.2 Binary Classification

In order to compare with some previous work, we run our model on the binary implicit relation classification task. The dataset is also from PDTB and conventional splitting for binary classification is followed: Section 2-20 for training, 0-1 for development and Section 21-22 for testing. For the training set, since the number of negative examples is much greater than the number of positive examples, extra negative examples are extracted randomly to provide balanced training set. All examples in sections 21 and 22 are included for testing. Following previous work, the evaluation metric for binary classification will be Macro-F1 score. The hyper-parameters of our model are roughly the same as in multi-class classification expect that learning rate is set to 0.0002 for the binary classification task.

4.3 Results

As shown in Table 1, 2 and 3, the proposed model **Char+Word-Enhanced** outperforms most of the previous models, both for multi-class and binary classification task. This shows the effectiveness of context-aware character-enhanced embeddings and that these enhanced embeddings corporate well with sentence-level neural models.

5 Conclusion

In this paper, we propose a character-level neural module to obtain context-aware character-based embeddings for implicit discourse relation recognition. Utilizing the combined character-enhanced embeddings, our model performs well, which shows that the character-level information captured by the

proposed model may effectively improve this semantic understanding task.

References

Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from RST discourse parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2212–2218, Lisbon, Portugal, November.

Or Biran and Kathleen McKeown. 2013. Aggregated word pair features for implicit discourse relation disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 69–73, Sofia, Bulgaria, August.

Chloé Braud and Pascal Denis. 2015. Comparing word representations for implicit discourse relation classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2201–2211, Lisbon, Portugal, September.

Deng Cai and Hai Zhao. 2016. Neural Word Segmentation Learning for Chinese. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 409–420, Berlin, Germany, August.

Joyce Y Chai and Rong Jin. 2004. Discourse structure for context question answering. In *Proceedings of the Workshop on Pragmatics of Question Answering at North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*, volume 2004, pages 23–30, San Diego, USA.

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 (CONLL)*, pages 37–41, Beijing, China, July.

Jifan Chen, Qi Zhang, Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Implicit discourse relation detection via a deep architecture with gated relevance network. In *Proceedings of the 54rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Berlin, Germany, August.

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.

Robert Fisher and Reid Simmons. 2015. Spectral semi-supervised discourse relation classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 89–93, Beijing, China, July.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *International Conference on Artificial Neural Networks (ICANN)*, pages 799–804, Warsaw, Poland.

Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273–278, Olomouc, Czech Republic.

Francisco Guzmán, Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2014. Using discourse structure improves machine translation evaluation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 687–698, Baltimore, Maryland, June.

Christian Hardmeier. 2012. Discourse in statistical machine translation: A survey and a case study. *Discours-Revue de linguistique, psycholinguistique et informatique*, 11.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric P Xing. 2016a. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2410–2420, Berlin, Germany, August.

Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric P Xing. 2016b. Deep neural networks with massive learned knowledge. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin, USA, November.

Yangfeng Ji and Jacob Eisenstein. 2015. One vector is not enough: Entity-augmented distributed semantics for discourse relations. *Transactions of the Association for Computational Linguistics (TACL)*, 3:329–344.

Yangfeng Ji, Gongbo Zhang, and Jacob Eisenstein. 2015. Closing the gap: Domain adaptation from explicit to implicit discourse relations. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2219–2224, Lisbon, Portugal, September.

Yangfeng Ji, Gholamreza Haffari, and Jacob Eisenstein. 2016. A latent variable recurrent neural network for discourse-driven language models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 332–342, San Diego, California, June.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 2741–2749, Phoenix, USA.

Junyi Jessy Li and Ani Nenkova. 2014. Reducing sparsity improves the recognition of implicit discourse relations. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 199–207, Philadelphia, USA.

Zhongyi Li, Hai Zhao, Chenxi Pang, Lili Wang, and Huan Wang. 2016. A constituent syntactic parse tree based discourse parser. In *Proceedings of the Twentieth Conference on Computational Natural Language Learning - Shared Task (CONLL)*, pages 60–64, Berlin, Germany, August.

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 (EMNLP)*, pages 343–351, Suntec,Singapore.

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.

Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1520–1530, Lisbon, Portugal, September.

Sameer Maskey and Julia Hirschberg. 2005. Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization. In *the 9th biennial conference of the International Speech Communication Association (ISCA) and the 6th in the annual series of INTERSPEECH events (INTERSPEECH 2005-EUROSPEECH)*, pages 621–624, Lisbon, Portugal.

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 (3)*, pages 3111–3119, South Lake Tahoe, Nevada, USA, December.

Gabriel Murray, Steve Renals, Jean Carletta, and Johanna Moore. 2006. Incorporating speaker and discourse features into speech summarization. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (NAACL)*, pages 367–374, New York, USA.

Joonsuk Park and Claire Cardie. 2012. Improving implicit discourse relation recognition through feature set optimization. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 108–112, Seoul, South Korea, July.

Emily Pitler, Annie Louis, and Ani Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of the Joint Conference of the 47th Annual Meeting of he Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 683–691, Suntec, Singapore, August.

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 *The 6th edition of the Language Resources and Evaluation Conference (LREC)*, pages 2961–2968, Marrakech, Morocco.

Lianhui Qin, Zhisong Zhang, and Hai Zhao. 2016a. Shallow discourse parsing using convolutional neural network. In *Proceedings of the Twentieth Conference on Computational Natural Language Learning - Shared Task (CONLL)*, pages 70–77, Berlin, Germany, August.

Lianhui Qin, Zhisong Zhang, and Hai Zhao. 2016b. A stacking gated neural architecture for implicit discourse relation classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin, USA, November.

Attapol Rutherford and Nianwen Xue. 2014. Discovering implicit discourse relations through brown cluster pair representation and coreference patterns. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 645–654, Gothenburg, Sweden, April.

Attapol Rutherford and Nianwen Xue. 2015. Improving the inference of implicit discourse relations via classifying explicit discourse connectives. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL): Human Language Technologies*, pages 799–808, Denver, Colorado, May–June.

Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. 2007. Evaluating discourse-based answer extraction for why-question answering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 735–736, Amsterdam, Holland, July.

Peilu Wang, Yao Qian, Frank K Soong, Lei He, and Hai Zhao. 2015. Word embedding for recurrent neural network based TTS synthesis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883, Brisbane, Australia.

Peilu Wang, Yao Qian, Frank K. Soong, Lei He, and Hai Zhao. 2016. Learning distributed word representations for bidirectional LSTM recurrent neural network. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 527–533, San Diego, California, June.

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 (CONLL)*, pages 1–16, Beijing, China, July.

Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Bonnie Webber, Attapol Rutherford, Chuan Wang, and Hongmin Wang. 2016. The CoNLL-2016 shared task on shallow discourse parsing. In *Proceedings of the Twentieth Conference on Computational Natural Language Learning - Shared Task (CONLL)*, pages 1–19, Berlin, Germany, August.

Biao Zhang, Jinsong Su, Deyi Xiong, Yaojie Lu, Hong Duan, and Junfeng Yao. 2015a. Shallow convolutional neural network for implicit discourse relation recognition. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2230–2235, Lisbon, Portugal, September.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015b. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NIPS)*, pages 649–657, Montral, Quebec, Canada.

Zhisong Zhang, Hai Zhao, and Lianhui Qin. 2016. Probabilistic graph-based dependency parsing with convolutional neural network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1382–1392, Berlin, Germany, August.

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 (COLING)*, pages 1507–1514, Beijing, China, August.