# Tag-Enhanced Tree-Structured Neural Networks for Implicit Discourse Relation Classification

**Yizhong Wang**[1] **Sujian Li**[1,2] **Jingfeng Yang**[1] **Xu Sun**[1] **Houfeng Wang**[1,2]
[1]Key Laboratory of Computational Linguistics, Peking University, MOE, China
[2]Collaborative Innovation Center for Language Ability, Xuzhou, Jiangsu, China
{yizhong, lisujian, yjfllpyym, xusun, wanghf}@pku.edu.cn

## Abstract

Identifying implicit discourse relations between text spans is a challenging task because it requires understanding the meaning of the text. To tackle this task, recent studies have tried several deep learning methods but few of them exploited the syntactic information. In this work, we explore the idea of incorporating syntactic parse tree into neural networks. Specifically, we employ the Tree-LSTM model and Tree-GRU model, which are based on the tree structure, to encode the arguments in a relation. Moreover, we further leverage the constituent tags to control the semantic composition process in these tree-structured neural networks. Experimental results show that our method achieves state-of-the-art performance on PDTB corpus.

## 1 Introduction

It is widely agreed that text units such as clauses or sentences are usually not isolated. Instead, they correlate with each other to form coherent and meaningful discourse together. To analyze how text is organized, discourse parsing has gained much attention from both the linguistic (Weiss and Wodak, 2007; Tannen, 2012) and computational (Marcu, 1997; Soricut and Marcu, 2003) communities, but the current performance is far from satisfactory. The most challenging part is to identify the discourse relations between text spans, especially when the discourse connectives (e.g., "because" and "but") are not explicitly shown in the text. Due to the absence of such evident linguistic clues, trying to model and understand the meaning of the text becomes the key point in identifying such implicit relations.

**Arg1:** The index is intended to measure future economic performance

**Arg2:** A figure above 50 indicates the economy is likely to expand

(Expansion.Restatement.Specification, wsj_0233)

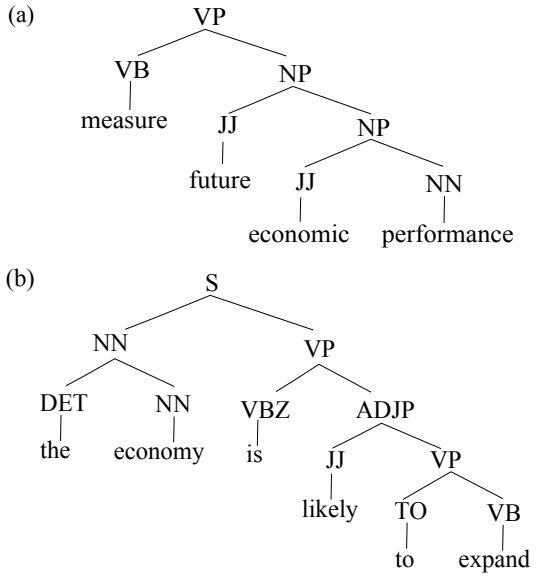


Figure 1: An example of two sentences with their discourse relation as *Expansion.Restatement.Specification*. Subfigure (a) and (b) are partial parse trees of the two important phrases with yellow background.

Previous studies in this field treat the task of recognizing implicit discourse relations as a classification problem and various techniques in semantic modeling have been adopted to encode the arguments in each relation, ranging from traditional feature-based models (Lin et al., 2009; Pitler et al., 2009) to the currently prevailing deep learning methods (Ji and Eisenstein, 2014; Liu and Li, 2016; Qin et al., 2017). Despite of the superior ability of the deep learning models, the syntactic



*Proceedings of the The 8th International Joint Conference on Natural Language Processing*, pages 496–505,
Taipei, Taiwan, November 27 – December 1, 2017 

information, which proves to be helpful for identifying discourse relations in many early studies (Subba and Di Eugenio, 2009; Lin et al., 2009), is seldom employed by recent work. Therefore we are curious to explore whether such missing syntactic information can be leveraged in deep learning methods to further improve the semantic modeling for implicit discourse relation classification.

Tree-structured neural networks, which recursively compose the representation of smaller text units into larger text spans along the syntactic parse tree, can tactfully combine syntatic tree structure with neural network models and recently achieve great success in several semantic modeling tasks (Eriguchi et al., 2016; Kokkinos and Potamianos, 2017; Chen et al., 2017). One useful property of these models is that the representation of phrases can be naturally captured while computing the representations from bottom up. Taking Figure 1 for an example, those highlighted phrases could provide important signals for classifying the discourse relation. Therefore, we will employ two latest tree-structuerd models, i.e. the Tree-LSTM model (Tai et al., 2015; Zhu et al., 2015) and the Tree-GRU model (Kokkinos and Potamianos, 2017), in our work. Hopefully, these models can learn to preserve or highlight such helpful phrasal information while encoding the arguments.

Another important syntactic signal comes from the constituent tags on the tree nodes (e.g., NP, VP, ADJP). Those tags, derived from the production rules, describe the generative process of text and therefore could indicate which part is more important in each constituent. For example, considering a node tagged with NP, its child node tagged with DT is usually neglectable. Thus we propose to incorporate this tag information into the tree-structured neural networks, where those constituent tags can be used to control the semantic compostion process.

Therefore, in this paper, we will approach the discourse relation classification task with two tree-structured neural networks proposed recently (Tree-LSTM and Tree-GRU ). To our knowledge, this is the first time these models are applied to discourse relation classification. Moreover, we further enhance these models by leveraging the constituent tags to compute the gates in these models. Experiments on PDTB 2.0 (Prasad et al., 2008) show that the models we propose can achieve state-of-the-art results.

## 2 Related Work

### 2.1 Implicit Discourse Relation Classication

Discourse relation identification is an important but difficult sub-component of discourse analysis. One fundamental step forward recently is the release of the large-scale Penn Discourse TreeBank (PDTB) (Prasad et al., 2008), which annotates discourse relations with their two textual arguments over the 1 million word Wall Street Journal corpus. The discourse relations in PDTB are broadly categorized as either "Explicit" or "Implicit" according to whether there are connectives in the original text that can indicate the sense of the relations. In the absence of explicit connectives, identifying the sense of the relations has proved to be much more difficult (Park and Cardie, 2012; Rutherford and Xue, 2014) since the inferring is solely based on the arguments.

Prior work usually tackles this task of implicit discourse relation identification as a classification problem with the classes defined in PDTB corpus. Early attempts use traditional various feature-based methods and the work inspiring us most is Lin et al. (2009), in which they show that the syntactic parse structure can provide useful signals for discourse relation classification. More specifically they employ the production rules with constituent tags (e.g., SBJ) as features and get competitive performance. Recently, with the popularity of deep learning methods, many cutting-edge models are also applied to our task of implicit discourse relation classification. Qin et al. (2016) tries to model the sentences with Convolutional Neural Networks. Liu and Li (2016) encodes the text with Long Short Term Memory model and employ multi-level attention mechanism to capture important signals. Qin et al. (2017) proposes a framework based on adversarial network to incorporate the connective information. To be noted, Ji and Eisenstein (2014) adopts Recursive Neural Network to exploit the representation of sentences and entities, which is the first yet simple tree-structured neural network applied in this task.

### 2.2 Tree-Structured Neural Networks

Tree-structured neural networks are one of the most widely-used deep learning models in natural language processing. Such neural networks usually recursively computes the representation of larger text spans from its constituent units according to the syntactic parse tree. Thanks to this

compositional nature of text, tree-structured neural network models show superior ability in a variety of semantic modeling tasks, such as sentiment classification (Kokkinos and Potamianos, 2017), natural language inference (Chen et al., 2017) and machine translation (Eriguchi et al., 2016).

The earliest and simplest tree-structure neural network is the Recursive Neural Network proposed by Socher et al. (2011), in which a global matrix is learned to linearly combine the contituent vectors. This work is further extended by replacing the global matrix with a global tensor to form the Recursive Neural Tensor Network (Socher et al., 2013). Based on them, Qian et al. (2015) first proposes to incorporate tag information, which is very similar as our idea described in Section 3.2, by either choosing a composition function according to the tag of a phrase (Tag-Guided RNN/RNTN) or combining the tag embeddings with word embeddings (Tag-Embedded RNN/RNTN). Our method of incorporating tag information improves from theirs and somewhat combines these two methods by using the tag embedding to dynamically determine the composition function via the gates in LSTM or GRU.

One fatal weakness of vanilla RNN/RNTN is the well-known gradient exploding or vanishing problem due to the multiple computation steps in the vertical direction. Therefore Tai et al. (2015) and Zhu et al. (2015) propose to import the Long Short Term Memory into tree structured neural networks and design a novel network architecture called Tree-LSTM. The adoption of the memory cell enables the Tree-LSTM model to preserve information even though the tree becomes very high. Similar to Tree-LSTM, Kokkinos and Potamianos (2017) introduces the so-called Tree-GRU network, which replace the LSTM unit with Gated Recurrent Unit (GRU). With less parameters to train, Tree-GRU achieves better performance on the sentiment analysis task. In this work, we will experiment with these Tree-LSTM and Tree-GRU models for the semantic modeling in implicit relation classification.

## 3 Our Method

This section details the models we use for implicit discourse relation classification. Given two textual arguments without explicit connectives, our task is to classify the discourse relation between them. It can be viewed as two parts: 1) modeling the semantics of the two arguments; 2) classifying the relations based on the semantics. Our main contribution concentrates on the semantic modeling part with two types of tree-structured neural networks described in Section 3.1 and we further illustrate how to leverage the constituent tags to enhance these two models in Section 3.2. In Section 3.3, we will shortly introduce the relation classifier and the training procedure of our model. The architecture of our system is illustrated in Figure 2.

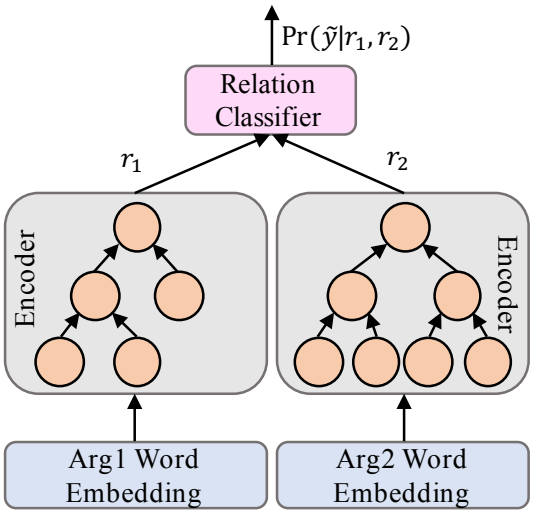


Figure 2: Architecture of our discourse relation classification model. Layers with the same color share the same parameters.

### 3.1 Modeling the Arguments with Tree-Structured Neural Networks

In a typical tree-structured neural network, given a parse tree of the text, the semantic representations of smaller text units are recursively composed to compute the representation of larger text spans and finally compute the representation for the whole text (e.g., sentence). In this work, we will construct our models based on the constituency parse tree, as is shown in Figure 1. Following previous convention (Eriguchi et al., 2016; Zhao et al., 2017), we convert the general parse tree, where the branching factor may be arbitrary, into a binary tree so that we only need to consider the left and right children at each step. Then the following Tree-LSTM and Tree-GRU models can be used to obtain a vector representation of each argument.

**Tree-LSTM Model.** In a standard sequential LSTM model, the LSTM unit is repeated at each

step to take the word at current step and previous output as its input, update its memory cell and output a new hidden vector. In the Tree-LSTM model, a similar LSTM unit is applied to each node in the tree in a bottom-up manner. Since each internal node in the binary parse tree has two children, the Tree-LSTM unit has to consider information from two preceding nodes, as opposed to the single preceding node in the sequential LSTM model. Each Tree-LSTM unit (indexed by $j$) contains an input gate $i_j$, a forget gate $f_j$ [1] and an output gate $o_j$. The computation equations at node $j$ are as follows:

$$i_j = \sigma\left(W^{(i)}x_j + U^{(i)}\left[h_j^L, h_j^R\right]\right) \quad (1)$$

$$f_j = \sigma\left(W^{(f)}x_j + U^{(f)}\left[h_j^L, h_j^R\right]\right) \quad (2)$$

$$o_j = \sigma\left(W^{(o)}x_j + U^{(o)}\left[h_j^L, h_j^R\right]\right) \quad (3)$$

$$u_j = \tanh\left(W^{(u)}x_j + U^{(u)}\left[h_j^L, h_j^R\right]\right) \quad (4)$$

$$c_j = i_j \odot u_j + f_j \odot c_j^L + f_j \odot c_j^R \quad (5)$$

$$h_j = o_j \odot \tanh\left(c_j\right) \quad (6)$$

where $x_j$ is the embedded word input at current node $j$, $\sigma$ denotes the logistic sigmoid function and $\odot$ denotes element-wise multiplication. $h_j^L$, $h_j^R$ are the output hidden vectors of the left and right children, and $c_j^L$, $c_j^R$ are the memory cell states from them, respectively. To save space, we leave out all the bias terms in affine transformations and the same is true for other affine transformations in this paper.

Intuitively, $u_j$ can be regarded as a summary of the inputs at current node, which is then filtered by $i_j$. The memory from left and right children are forgotten by $f_j$ and then we compose them together with the new inputs to form the the new memory $c_j$. At last, part of the information in memory $c_j$ is exposed by $o_j$ to generate the output vector $h_j$ for current step. Another thing to note is that only leaf nodes in the constituency tree have words as its input, so $x_j$ is set to a zero vector in other cases.

**Tree-GRU Model.** Similar to Tree-LSTM, the Tree-GRU model extends the sequential GRU model to tree structures. The only difference between Tree-GRU and Tree-LSTM is how they modulate the flow of information inside the unit. Specifically, Tree-GRU unit removes the separate memory cell and only uses two gates to simulate the reset and update procedure in information gathering. The computation equations in each Tree-GRU unit are the following:

$$r_j = \sigma\left(W^{(r)}x_j + U^{(r)}\left[h_j^L, h_j^R\right]\right) \quad (7)$$

$$z_j = \sigma\left(W^{(z)}x_j + U^{(z)}\left[h_j^L, h_j^R\right]\right) \quad (8)$$

$$\tilde{h}_j = \tanh\left(W^{(h)}x_j + U^{(h)}\left[h_j^L \odot r_j, h_j^R \odot r_j\right]\right) \quad (9)$$

$$h_j = z_j \odot \tilde{h}_j + (1 - z_j) \odot \left(h_j^L + h_j^R\right) \quad (10)$$

where $r_j$ is the reset gate and $z_j$ is the update gate. The reset gate allows the network to forget previous computed representations, while the update gate decides the degree of update to the hidden state. There is no memory cell in Tree-GRU, with only $h_j^L$ and $h_j^R$ as the hidden states from the left and right children.

### 3.2 Controlling the Semantic Composition with Constituent Tags

The constituent tag in a parse tree describes the grammatical role of its corresponding constituent in the context. Bies et al. (1995) defines several types of constituent tags, including clause-level tags (e.g., SBAR, SINV, SQ), phrase-level tags (e.g., NP, VP, PP) and word-level tags (e.g., NN, VP, JJ). These constituent tags greatly interleave with the semantics and in some ways can provide determinant signals for the importance of a constituent. For example, for most of the time, constituents with PP (prepositional phrase) tag are less important than those with VP (verb phrase) tag. Therefore we argue that these tags are worth considering when we compose the semantics in the tree-structured neural networks.

One way to leverage such tags is using tag-specific composition functions but that would lead to large number of parameters and some tags are very sparse so it's very hard to train their corresponding parameters sufficiently. To solve this problem, we propose to use tag embeddings and dynamically control the composition process via the gates in our model.

Gates in Tree-LSTM and Tree-GRU units control the flow of information and thus determine how the semantics from child nodes are composed

[1] The original Binary Tree-LSTM in (Tai et al., 2015) contains separate forget gates for different child nodes but we find single forget gate performs better in our task.

to a new representation. Furthermore, these gates are computed dynamically according to the inputs at a certain step. Therefore, it's natural to incorporate the tag embeddings in the computation of these gates. Based on this idea, we propose the Tag-Enhanced Tree-LSTM model, where the input, forget and output gates in each unit are calculated as follows:

$$i_j = \sigma\left(W^{(i)}x_j + M^{(i)}t_j + U^{(i)}\left[h_j^L, h_j^R\right]\right) \tag{11}$$

$$f_j = \sigma\left(W^{(f)}x_j + M^{(f)}t_j + U^{(f)}\left[h_j^L, h_j^R\right]\right) \tag{12}$$

$$o_j = \sigma\left(W^{(o)}x_j + M^{(o)}t_j + U^{(o)}\left[h_j^L, h_j^R\right]\right) \tag{13}$$

Similarly, we can have the Tag-Enhanced Tree-GRU model with new reset and update gates:

$$r_j = \sigma\left(W^{(r)}x_j + M^{(r)}t_j + U^{(r)}\left[h_j^L, h_j^R\right]\right) \tag{14}$$

$$z_j = \sigma\left(W^{(z)}x_j + M^{(z)}t_j + U^{(z)}\left[h_j^L, h_j^R\right]\right) \tag{15}$$

where $t_j$ is the embedding of the tag at current node (indexed by $j$).

### 3.3 Relation Classification and Training

In our work, the two arguments are encoded with the same network in order to reduce the number of parameters. After that we get a vector representation for each argument, which can be denoted as $r_1$ for argument 1 and $r_2$ for argument 2. Supposing that there are totally $n$ relation types, the predicted probability distribution $\hat{y} \in \mathbb{R}^n$ is calculated as:

$$\hat{y} = softmax\left(W^{(\hat{y})}\left[r_1, r_2\right] + b^{(\hat{y})}\right) \tag{16}$$

To train our model, the training objective $J$ is dened as the cross-entropy loss with $L2$ regularization:

$$E\left(\hat{y}, y\right) = -\sum_{j=1}^{n} y_j \times \log \hat{y}_j \tag{17}$$

$$J\left(\theta\right) = \frac{1}{N}\sum_{k=1}^{N} E\left(\hat{y}, y\right) + \frac{\lambda}{2}\|\theta\|^2 \tag{18}$$

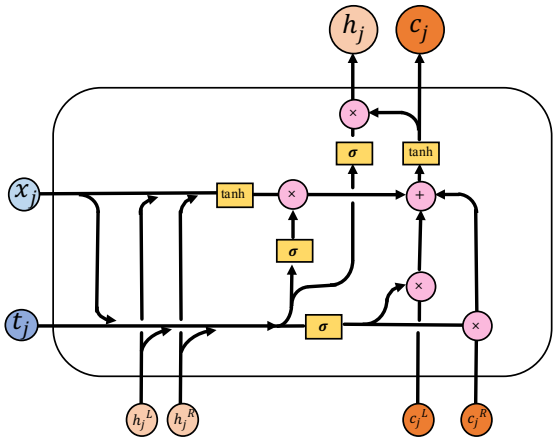


Figure 3: Flow of Information in Tag-Enhanced Tree-LSTM unit.

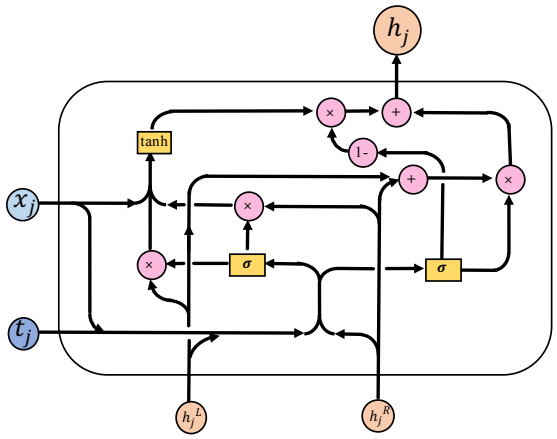


Figure 4: Flow of Information in Tag-Enhanced Tree-GRU unit.

where $\hat{y}$ is the predicted probability distribution, $y$ is the one-hot representation of the gold label and $N$ is the number of training samples.

## 4 Experiments

### 4.1 Experiment Setup

**Corpus.** We evaluate our method on the Penn Discourse Treebank (PDTB) (Prasad et al., 2008), which provides annotations of discourse relations over the Wall Street Journal corpus. Each relation instance consists of two arguments, typically adjacent pairs of sentences in a text. As is mentioned in Section 2.1, the relations in PDTB are generally categorized into either explicit or implicit and our work focuses on the more challenging implicit relation classification task. Totally, there are 16,224 implicit relation instances in PDTB dataset, with a three-level hierarchy. The first level is defined as 4 major classes of the relation, including: *Tem-*

| Models | Level-1 Classification | | Level-2 Classification | |
|---|---|---|---|---|
| | Dev | Test | Dev | Test |
| Bi-LSTM | 55.10 | 56.88 | 35.02 | 42.44 |
| Bi-GRU | 55.21 | 57.01 | 35.34 | 42.46 |
| Tree-LSTM | 56.04 | 58.89 | 35.76 | 43.02 |
| Tree-GRU | 55.36 | 58.98 | 36.09 | 43.78 |
| Tag-Enhanced Tree-LSTM | **56.97** | **59.85** | 35.92 | **45.21** |
| Tag-Enhanced Tree-GRU | 56.63 | 59.75 | **36.93** | 44.55 |

Table 1: The accuracy score of multi-class classification

*poral*, *Contingency*, *Comparison* and *Expansion*. Then for each class, it is further divided into different types, which is supposed to provide finer pragmatic distinctions. This totally yields 16 relation types at the second level. At last, a third level of subtypes is defined for some types according to the semantic contribution of each argument.

**Preprocesssing.** Following the common setup convention (Rutherford and Xue, 2014; Ji and Eisenstein, 2014; Liu and Li, 2016), we split the dataset into training set (Sections 2-20), development set (Sections 0-1), and test set (Section 21-22). For preprocessing, we employ the Stanford CoreNLP toolkit (Manning et al., 2014) to lemmatize all the words and get the constituency parse tree for each sentence. Then we convert the parse tree into binary with right branching and we remove the internal nodes that have only one child so that our binary tree-structured models can be applied. Small portion of the arguments in PDTB are composed of multiple sentences. In such cases, we add a new "Root" node and link the original root nodes of those sentences to this shared "Root" before converting the tree into binary.

**Multi-Class Classification** There are mainly two ways to set up the classification tasks in previous work. Early studies (Pitler et al., 2009; Park and Cardie, 2012) train and evaluate separate "one-versus-all" classifiers for each discourse relation since the classes are extremely imbalanced in PDTB. However, recent work put more emphasis on the multi-class classification, where the goal is to identify a discourse relation from all possible choices. According to Rutherford and Xue (2014), the multi-class classification setting is more natural and realistic. Moreover, the multi-class classifier can directly serve as one building block of a complete discourse parser (Qin et al., 2017). Therefore, in this work, we will focus on the multi-class classification task. Moreover, we will experiment on the classification of both the Level-1 classes and the Level-2 types, so that we can compare with most of previous systems and thoroughly analyze the performance of our method.

It should be noted that roughly 2% of the implicit relation instances are annotated with more than one types. Following Ji and Eisenstein (2014), we treat these different types as multiple instances while training our model. During testing, if the classifier hits either of the annotated types, we consider it to be correct. We also follow previous studies (Lin et al., 2009; Ji and Eisenstein, 2014) to remove the instances of 5 very rare relation types in the second level. Therefore we have totally 11 types to classify for Level-2 classification and 4 classes for Level-1 classification.

### 4.2 Model Settings

We tune the hyper-parameters of our Tag-Enhanced Tree-LSTM model based on the development set and other models share the same set of hyper-parameters. The best-validated hyper-parameters, including the size of the word embeddings $\omega$, the size of the tag embeddings $\tau$, the dimension of the Tree-LSTM or Tree-GRU hidden state $d$, the learning rate $\eta$, the weight of L2 regularization term $\lambda$ and the batch size $b$ are shown in Table 2.

| $\omega$ | $\tau$ | $d$ | $\eta$ | $\lambda$ | $b$ |
|---|---|---|---|---|---|
| 50 | 50 | 250 | 0.01 | 0.0001 | 10 |

Table 2: Hyper-parameters of our model

The Pre-trained 50-dimentional Glove Vectors (Pennington et al., 2014), which is case-insensitive, are used for initializing the word embeddings and they are tuned together with other parameters in the same learning rate during training.

We adopt the AdaGrad optimizer (Duchi et al., 2011) for training our model and we validate the performance every epoch. It takes around 5 hours (5 epochs) for the Tag-Enhanced Tree-LSTM and 4 hours (6 epochs) for the Tag-Enhanced Tree-GRU model to converge to the best performance, using one INTEL(R) Core(TM) I7 3.4GHz CPU and one NVIDIA GeForce GTX 1080 GPU.

### 4.3 Results

The evaluation results of our models on both the Level-1 classification and the Level-2 classification are reported in Table 1. Accuracy score is used to measure the overall performance and we present our performance on both the development set and the test set. In addition to the four tree-structured neural networks described in Section 3, we also implement two baseline models: the bi-directional LSTM model and the bi-directional GRU model. The hyper-parameters of these two models are tuned separately from other models. Due to the space limitation, we don't present the details here. Comparison of these sequential models with the tree-structured models are expected to show the effects of tree structures.

From Table 1, we can see that the sequential Bi-LSTM model and Bi-GRU model perform worst for our task, which confirms our hypothesis that the tree-structured neural networks can really capture some important signals that are missing in the sequential models.

Furthermore, if we add the tag information to tree-structured models, both the Tag-Enhanced Tree-LSTM model and the Tag-Enhanced Tree-GRU model provide conspicuous improvement (around 1%) compared with the no-tag version. This demonstrates the usefulness of those constituent tags and the effectiveness of our method to incorporate this important feature. Especially, since both the Tree-LSTM model and Tree-GRU model rely on the gating machanism to control the flow of information, this double confirms that the tag information can help with the computation of such gates and therefore can be leveraged to control the semantic composition process.

Another discovery from our results is that the GRU models performs similarly as its corresponding variant of LSTM model. This conforms to previous empirical observation in sequential models that LSTM and GRU have comparable capability (Chung et al., 2014). However, the Tree-GRU model have less parameters to train which could alleviate the problem of overfitting and also cost less training time.

| Systems | Accuracy |
|---|---|
| Zhang et al. (2015) | 55.39 |
| Rutherford and Xue (2014) | 55.50 |
| Rutherford and Xue (2015) | 57.10 |
| Liu et al. (2016) | 57.27 |
| Liu and Li (2016) | 57.57 |
| Ji et al. (2016) | 59.50 |
| Tag-Enhanced Tree-LSTM | **59.85** |
| Tag-Enhanced Tree-GRU | 59.75 |

Table 3: Accuracy (%) for Level-1 multi-class classification on the test set, compared with other state-of-the-art systems.

| Systems | Accuracy |
|---|---|
| Lin et al. (2009) | 40.66 |
| Ji and Eisenstein (2014) | 44.59 |
| Qin et al. (2016) | 45.04 |
| Qin et al. (2017) | **46.23** |
| Tag-Enhanced Tree-LSTM | 45.21 |
| Tag-Enhanced Tree-GRU | 44.55 |

Table 4: Accuracy (%) for Level-2 multi-class classification on the test set, compared with other state-of-the-art systems.

### 4.4 Comparison with Other Systems

For a comprehensive study, we compare our models with other state-of-the-art systems. The systems that conduct Level-1 classification are reported in Table 3, including:

- Zhang et al. (2015) proposes to use convolutional neural networks to encode the arguments.
- Rutherford and Xue (2014) manually extracts features to represent the arguments and use a maximum entropy classifier for classification. Rutherford and Xue (2015) further exploits discourse connectives to enrich the training data.
- Liu et al. (2016) employs a multi-task framework that can leverage other discourse-related data to help with the training of discourse relation classifier.
- Liu and Li (2016) represents arguments with LSTM and introduces a multi-level attention

mechanism to model the interaction between the two arguments.

- Ji et al. (2016) treats the discourse relation as latent variable and proposes to model them jointly with the sequences of words using a latent variable recurrent neural network architecture.

And in Table 4, we present the following systems, which focus on Level-2 classification:

- Lin et al. (2009) uses traditional feature-based model to classify relations. Especially, constituent and dependency parse trees are exploited.
- Ji and Eisenstein (2014) models both the semantics of argument and the meaning of entity mention with two recursive neural networks, which are then combined to classify relations.
- Qin et al. (2016) utilize convolutional neural network for argument modeling and a collaborative gated neural network to model their interaction.
- Qin et al. (2017) proposes to incorporate the connective information via a novel pipelined adversarial framework.

The comparison with these latest work shows that our system achieves currently best performance for the Level-1 classification and ranks second for the Level-2. With the state-of-the-art performance on both levels, we can verify the effectiveness of our method.

### 4.5 Qualitative Analysis

To get a deeper insight into the proposed tag-enhanced models, we project the constituent tag embeddings learned in our Tag-Enhanced Tree-LSTM model into two dimensions using the t-SNE method (Maaten and Hinton, 2008) and normalize the values in each dimension. The projected embeddings are visualized in Figure 5 and we highlight some representative tags that may share some kind of commonality in their functions.

According to the definition in Bies et al. (1995), the tags with red color are all verb-related, while those with blue color describes noun-related syntax. Despite of some noisy points, we can see that they are roughly separated into two groups.

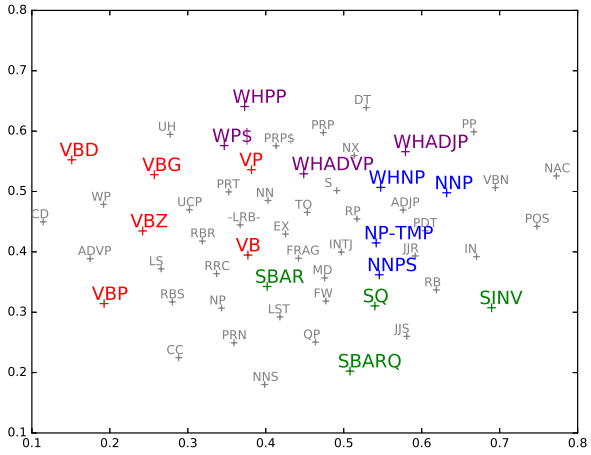


Figure 5: t-SNE Visualization (Maaten and Hinton, 2008) of the constituent tag embeddings

In addition, the purple tags correspond to words or phrase that are wh-related (e.g., what, when, where, which) and we can see they are distributed similarly. Moreover, the green tags are, which are located in the right-bottom corner, all describe the clause-level constituents.

Therefore, from this visualization, we can conclude that the tag embeddings we learned are somewhat meaningful and really capture some functionalities of these tags. Since these embeddings are used to compute the gates and the gates further determine the flow of information, we argue that these tags can indeed help to control the semantic composition process in our tree-structured networks.

## 5 Conclusion

In this work, we propose to use two latest tree-structured neural networks to model the arguments for discourse relation classification. The syntactic parse tree are exploited from two aspects: first, we leverage the tree structure to recursively compose semantics in a bottom-up manner; second, the constituent tags are used to control the semantic composition process at each step via gating mechanism. Comprehensive experiments show the effectiveness of our proposed method and our system achieves state-of-the-art performance for the challenging task of implicit discourse relation classification. For future work, we will try other types of syntax embeddings and we are also working on incorporating structural attention mechanism into our tree-based models.

## Acknowledgments

We thank all the anonymous reviewers for their insightful comments on this paper. This work was partially supported by National Natural Science Foundation of China (61572049 and 61333018). The correspondence author of this paper is Sujian Li.

## References

Ann Bies, Mark Ferguson, Karen Katz, Robert MacIntyre, Victoria Tredinnick, Grace Kim, Mary Ann Marcinkiewicz, and Britta Schasberger. 1995. Bracketing guidelines for treebank ii style penn treebank project. *University of Pennsylvania*, 97:100.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proc. ACL*.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

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

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016. Tree-to-sequence attentional neural machine translation. *arXiv preprint arXiv:1603.06075*.

Yangfeng Ji and Jacob Eisenstein. 2014. One vector is not enough: Entity-augmented distributional semantics for discourse relations. *arXiv preprint arXiv:1411.6699*.

Yangfeng Ji, Gholamreza Haffari, and Jacob Eisenstein. 2016. A latent variable recurrent neural network for discourse relation language models. *arXiv preprint arXiv:1603.01913*.

Filippos Kokkinos and Alexandros Potamianos. 2017. Structural attention neural networks for improved sentiment analysis. *arXiv preprint arXiv:1701.01811*.

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.

Yang Liu and Sujian Li. 2016. Recognizing implicit discourse relations via repeated reading: Neural networks with multi-level attention. *arXiv preprint arXiv:1609.06380*.

Yang Liu, Sujian Li, Xiaodong Zhang, and Zhifang Sui. 2016. Implicit discourse relation classification via multi-task neural networks. In *AAAI*, pages 2750–2756.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Daniel Marcu. 1997. The rhetorical parsing of natural language texts. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 96–103. Association for Computational Linguistics.

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

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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 the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 683–691. Association for Computational Linguistics.

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 *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*.

Qiao Qian, Bo Tian, Minlie Huang, Yang Liu, Xuan Zhu, and Xiaoyan Zhu. 2015. Learning tag embeddings and tag-specific composition functions in recursive neural network. In *ACL (1)*, pages 1365–1374.

Lianhui Qin, Zhisong Zhang, and Hai Zhao. 2016. A stacking gated neural architecture for implicit discourse relation classification. In *EMNLP*, pages 2263–2270.

Lianhui Qin, Zhisong Zhang, Hai Zhao, Zhiting Hu, and Eric P Xing. 2017. Adversarial connective-exploiting networks for implicit discourse relation classification. *arXiv preprint arXiv:1704.00217*.

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, 2014*, pages 645–654.

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

Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 129–136.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, Christopher Potts, et al. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642.

Radu Soricut and Daniel Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 149–156. Association for Computational Linguistics.

Rajen Subba and Barbara Di Eugenio. 2009. An effective discourse parser that uses rich linguistic information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 566–574. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Deborah Tannen. 2012. Discourse analysis–what speakers do in conversation. *Linguistic Society of America*.

Gilbert Weiss and Ruth Wodak. 2007. *Critical discourse analysis*. Springer.

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

Kai Zhao, Liang Huang, and Mingbo Ma. 2017. Textual entailment with structured attentions and composition. *CoRR*, abs/1701.01126.

Xiaodan Zhu, Parinaz Sobihani, and Hongyu Guo. 2015. Long short-term memory over recursive structures. In *International Conference on Machine Learning*, pages 1604–1612.