Recursive Deep Models for Discourse Parsing

Jiwei Li[1], **Rumeng Li**[2] **and Eduard Hovy**[3]
[1]Computer Science Department, Stanford University, Stanford, CA 94305, USA
[2]School of EECS, Peking University, Beijing 100871, P.R. China
[3]Language Technology Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA
jiweil@stanford.edu alicerumeng@foxmail.com ehovy@andrew.cmu.edu

Abstract

Text-level discourse parsing remains a challenge: most approaches employ features that fail to capture the intentional, semantic, and syntactic aspects that govern discourse coherence. In this paper, we propose a recursive model for discourse parsing that jointly models distributed representations for clauses, sentences, and entire discourses. The learned representations can to some extent learn the semantic and intentional import of words and larger discourse units automatically,. The proposed framework obtains comparable performance regarding standard discoursing parsing evaluations when compared against current state-of-art systems.

1 Introduction

In a coherent text, units (clauses, sentences, and larger multi-clause groupings) are tightly connected semantically, syntactically, and logically. Mann and Thompson (1988) define a text to be coherent when it is possible to describe clearly the role that each discourse unit (at any level of grouping) plays with respect to the whole. In a coherent text, no unit is completely isolated. Discourse parsing tries to identify how the units are connected with each other and thereby uncover the hierarchical structure of the text, from which multiple NLP tasks can benefit, including text summarization (Louis et al., 2010), sentence compression (Sporleder and Lapata, 2005) or question-answering (Verberne et al., 2007).

Despite recent progress in automatic discourse segmentation and sentence-level parsing (e.g., (Fisher and Roark, 2007; Joty et al., 2012; Soricut and Marcu, 2003), document-level discourse parsing remains a significant challenge. Recent attempts (e.g., (Hernault et al., 2010b; Feng and Hirst, 2012; Joty et al., 2013)) are still considerably inferior when compared to human gold-standard discourse analysis. The challenge stems from the fact that compared with sentence-level dependency parsing, the set of relations between discourse units is less straightforward to define. Because there are no clause-level 'parts of discourse' analogous to word-level parts of speech, there is no discourse-level grammar analogous to sentence-level grammar. To understand how discourse units are connected, one has to understand the communicative function of each unit, and the role it plays within the context that encapsulates it, taken recursively all the way up for the entire text. Manually developed features relating to words and other syntax-related cues, used in most of the recent prevailing approaches (e.g., (Feng and Hirst, 2012; Hernault et al., 2010b)), are insufficient for capturing such nested intentionality.

Recently, deep learning architectures have been applied to various natural language processing tasks (for details see Section 2) and have shown the advantages to capture the relevant semantic and syntactic aspects of units in context. As word distributions are composed to form the meanings of clauses, the goal is to extend distributed clause-level representations to the single- and multi-sentence (discourse) levels, and produce the hierarchical structure of entire texts.

Inspired by this idea, we introduce in this paper a deep learning approach for discourse parsing. The proposed parsing algorithm relies on a recursive neural network to decide (1) whether two discourse units are connected and if so (2) by what relation they are connected. Concretely, the parsing algorithm takes as input a document of any length, and first obtains the distributed representation for each of its sentences using recursive convolution based on the sentence parse tree. It then proceeds

bottom-up, applying a binary classifier to determine the probability of two adjacent discourse units being merged to form a new subtree followed by a multi-class classifier to select the appropriate discourse relation label, and calculates the distributed representation for the subtree so formed, gradually unifying subtrees until a single overall tree spans the entire sentence. The compositional distributed representation enables the parser to make accurate parsing decisions and capture relations between different sentences and units. The binary and multi-class classifiers, along with parameters involved in convolution, are jointly trained from a collection of gold-standard discourse structures.

The rest of this paper is organized as follows. We present related work in Section 2 and describe the RST Discourse Treebank in Section 3. The sentence convolution approach is illustrated in Section 4 and the discourse parser model in Section 5. We report experimental results in Section 6 and conclude in Section 7.

2 Related Work

2.1 Discourse Analysis and Parsing

The basis of discourse structure lies in the recognition that discourse units (minimally, clauses) are related to one another in principled ways, and that the juxtaposition of two units creates a joint meaning larger than either unit's meaning alone. In a coherent text this juxtaposition is never random, but serves the speaker's communicative goals.

Considerable work on linguistic and computational discourse processing in the 1970s and 80s led to the development of several proposals for relations that combine units; for a compilation see (Hovy and Maier, 1997). Of these the most influential is Rhetorical Structure Theory RST (Mann and Thompson, 1988) that defines about 25 relations, each containing semantic constraints on its component parts plus a description of the overall functional/semantic effect produced as a unit when the parts have been appropriately connected in the text. For example, the SOLUTIONHOOD relation connects one unit describing a problem situation with another describing its solution, using phrases such as "the answer is"; in successful communication the reader will understand that a problem is described and its solution is given.

Since there is no syntactic definition of a problem or solution (they can each be stated in a single clause, a paragraph, or an entire text), one has to characterize discourse units by their communicative (rhetorical) function. The functions are reflected in text as signals of the author's intentions, and take various forms (including expressions such as "therefore", "for example", "the answer is", and so on; patterns of tense or pronoun usage; syntactic forms; etc.). The signals govern discourse blocks ranging from a clause to an entire text , each one associated with some discourse relation.

In order to build a text's hierarchical structure, a discourse parser needs to recognize these signals and use them to appropriately compose the relationship and nesting. Early approaches (Marcu, 2000a; LeThanh et al., 2004) rely mainly on overt discourse markers (or cue words) and use hand-coded rules to build text structure trees, bottom-up from clauses to sentences to paragraphs.... Since a hierarchical discourse tree structure is analogous to a constituency based syntactic tree, modern research explored syntactic parsing techniques (e.g., CKY) for discourse parsing based on multiple text-level or sentence-level features (Soricut and Marcu, 2003; Reitter, 2003; Baldridge and Lascarides, 2005; Subba and Di Eugenio, 2009; Lin et al., 2009; Luong et al., 2014).

A recent prevailing idea for discourse parsing is to train two classifiers, namely a binary structure classifier for determining whether two adjacent text units should be merged to form a new subtree, followed by a multi-class relation classifier for determining which discourse relation label should be assigned to the new subtree. The idea is proposed by Hernault and his colleagues (Duverle and Prendinger, 2009; Hernault et al., 2010a) and followed by other work using more sophisticated features (Feng and Hirst, 2012; Hernault et al., 2010b). Current state-of-art performance for relation identification is achieved by the recent representation learning approach proposed by (Ji and Eisenstein, 2014). The proposed framework presented in this paper is similar to (Ji and Eisenstein, 2014) for transforming the discourse units to the abstract representations.

2.2 Recursive Deep Learning

Recursive neural networks constitute one type of deep learning frameworks which was first proposed in (Goller and Kuchler, 1996). The recursive framework relies and operates on structured inputs (e.g., a parse tree) and computes the representation for each parent based on its children iteratively in a bottom-up fashion. A series of variations of RNN has been proposed to tailor different task-specific requirements, including Matrix-Vector RNN (Socher et al., 2012) that represents every word as both a vector and a matrix, or Recursive Neural Tensor Network (Socher et al., 2013) that allows the model to have greater interactions between the input vectors. Many tasks have benefited from the recursive framework, including parsing (Socher et al., 2011b), sentiment analysis (Socher et al., 2013), textual entailment (Bowman, 2013), segmentation (Wang and Mansur, 2013; Houfeng et al., 2013), and paraphrase detection (Socher et al., 2011a).

3 The RST Discourse Treebank

There are today two primary alternative discourse treebanks suitable for training data: the Rhetorical Structure Theory Discourse Treebank RST-DT (Carlson et al., 2003) and the Penn Discourse Treebank (Prasad et al., 2008). In this paper, we select the former. In RST (Mann and Thompson, 1988), a coherent context or a document is represented as a hierarchical tree structure, the leaves of which are clause-sized units called Elementary Discourse Units (EDUs). Adjacent nodes (siblings in the tree) are linked with discourse relations that are either binary (hypotactic) or multi-child (paratactic). One child of each hypotactic relation is always more salient (called the NUCLEUS); its sibling (the SATELLITE) is less salient compared and may be omitted in summarization. Multi-nuclear relations (e.g., CONJUNCTION) exhibit no distinction of salience between the units.

The RST Discourse Treebank contains 385 annotated documents (347 for training and 38 for testing) from the Wall Street Journal. A total of 110 fine-grained relations defined in (Marcu, 2000b) are used for tagging relations in RST-DT. They are subtypes of 18 original high-level RST categories. For fair comparison with existing systems, we use in this work the 18 coarse-grained relation classes, which with nuclearity attached form a set of 41 distinct relations. Non-binary relations are converted into a cascade of right-branching binary relations.

Conventionally, discourse parsing in RST-DT involves the following sub-tasks: (1) EDU segmentation to segment the raw text into EDUs, (2) tree-building. Since the segmentation task is essentially clause delimitation and hence relatively easy (with state-of-art accuracy at most 95%), we focus on the latter problem. We assume that the gold-standard EDU segmentations are already given, as assumed in other past work (Feng and Hirst, 2012).

4 EDU Model

In this section, we describe how we compute the distributed representation for a given sentence based on its parse tree structure and contained words. Our implementation is based on (Socher et al., 2013). As the details can easily be found there, we omit them for brevity.

Let s denote any given sentence, comprised of a sequence of tokens $s = \{w_1, w_2, ..., w_{n_s}\}$, where n_s denotes the number of tokens in s. Each token w is associated with a specific vector embedding $\mathbf{e_w} = \{e_w^1, e_w^2, ..., e_w^K\}$, where K denotes the dimension of the word embedding. We wish to compute the vector representation h_s for current sentence, where $h_s = \{h_s^1, h_s^2, ..., h_s^K\}$.

Parse trees are obtained using the Stanford Parser[1], and each clause is treated as an EDU. For a given parent p in the tree and its two children c_1 (associated with vector representation h_{c_1}) and c_2 (associated with vector representation h_{c_2}), standard recursive networks calculate the vector for parent p as follows:

$$h_p = f(W \cdot [h_{c_1}, h_{c_2}] + b) \quad (1)$$

where $[h_{c_1}, h_{c_2}]$ denotes the concatenating vector for children representations h_{c_1} and h_{c_2}; W is a $K \times 2K$ matrix and b is the $1 \times K$ bias vector; and $f(\cdot)$ is the function $tanh$. Recursive neural models compute parent vectors iteratively until the root node's representation is obtained, and use the root embedding to represent the whole sentence.

[1] http://nlp.stanford.edu/software/lex-parser.shtml

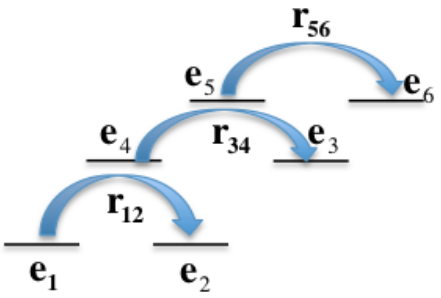

Figure 1: RST Discourse Tree Structure.

5 Discourse Parsing

Since recent work (Feng and Hirst, 2012; Hernault et al., 2010b) has demonstrated the advantage of combining the binary structure classifier (determining whether two adjacent text units should be merged to form a new subtree) with the multi-class classifier (determining which discourse relation label to assign to the new subtree) over the older single multi-class classifier with the additional label NO-REL, our approach follows the modern strategy but trains binary and multi-class classifiers jointly based on the discourse structure tree.

Figure 2 illustrates the structure of a discourse parse tree. Each node e in the tree is associated with a distributed vector h_e. e_1, e_2, e_3 and e_6 constitute the leaves of trees, the distributed vector representations of which are assumed to be already obtained from convolution in Section 4. Let N_r denote the number of relations and we have $N_r = 41$.

5.1 Binary (Structure) Classification

In this subsection, we train a binary (structure) classifier, which aims to decide whether two EDUs or spans should be merged during discourse tree reconstruction.

Let $t_{\text{binary}}(e_i, e_j)$ be the binary valued variable indicating whether e_i and e_j are related, or in other words, whether a certain type of discourse relations holds between e_i and e_j. According to Figure 2, the following pairs constitute the training data for binary classification:

$$t_{\text{binary}}(e_1, e_2) = 1, \quad t_{\text{binary}}(e_3, e_4) = 1,$$
$$t_{\text{binary}}(e_2, e_3) = 0, \quad t_{\text{binary}}(e_3, e_6) = 0,$$
$$t_{\text{binary}}(e_5, e_6) = 1$$

To train the binary classifier, we adopt a three-layer neural network structure, i.e., input layer, hidden layer, and output layer. Let $H = [h_{e_i}, h_{e_j}]$ denote the concatenating vector for two spans e_i and e_j. We first project the concatenating vector H to the hidden layer with N_{binary} hidden neurons. The hidden layer convolutes the input with non-linear tanh function as follows:

$$L^{\text{binary}}_{(e_i,e_j)} = f(G_{\text{binary}} * [h_{e_i}, h_{e_j}] + b_{\text{binary}})$$

where G_{binary} is an $N_{\text{binary}} * 2K$ convolution matrix and b_{binary} denotes the bias vector.

The output layer takes as input $L^{\text{binary}}_{(e_i,e_j)}$ and generates a scalar using the linear function $U_{binary} \cdot L^{\text{binary}}_{(e_i,e_j)} + b$. A $sigmod$ function is then adopted to project the value to a [0,1] probability space. The execution at the output layer can be summarized as:

$$p[t_{\text{binary}}(e_i, e_j) = 1] = g(U_{binary} \cdot L^{\text{binary}}_{(e_i,e_j)} + b^*_{\text{binary}}) \quad (2)$$

where U_{binary} is an $N_{\text{binary}} \times 1$ vector and b^*_{binary} denotes the bias. $g(\cdot)$ is the sigmod function.

5.2 Multi-class Relation Classification

If $t_{binary}(e_i, e_j)$ is determined to be 1, we next use variable $r(e_i, e_j)$ to denote the index of relation that holds between e_i and e_j. A multi-class classifier is train based on a three-layer neural network, in the similar way as binary classification in Section 5.1. Concretely, a matrix G_{Multi} and bias vector b_{Multi} are first adopted to convolute the concatenating node vectors to the hidden layer vector $L^{\text{multi}}_{(e_i,e_j)}$:

$$L^{\text{multi}}_{(e_i,e_j)} = f(G_{\text{multi}} * [h_{e_i}, h_{e_j}] + b_{\text{multi}}) \quad (3)$$

We then compute the posterior probability over labels given the hidden layer vector L using the softmax and obtain the N_r dimensional probability vector $P_{(e_1,e_2)}$ for each EDU pair as follows:

$$S_{(e_i,e_j)} = U_{\text{multi}} \cdot L^{\text{multi}}_{(e_i,e_j)} \quad (4)$$

$$P_{(e_1,e_2)}(i) = \frac{\exp(S_{(e_1,e_2)}(i))}{\sum_k \exp(S_{(e_1,e_2)})(k)} \quad (5)$$

where U_{multi} is the $N_r \times 2K$ matrix. The i^{th} element in $P_{(e_1,e_2)}$ denotes the probability that i^th relation holds between e_i and e_j. To note, binary and multi-class classifiers are trained independently.

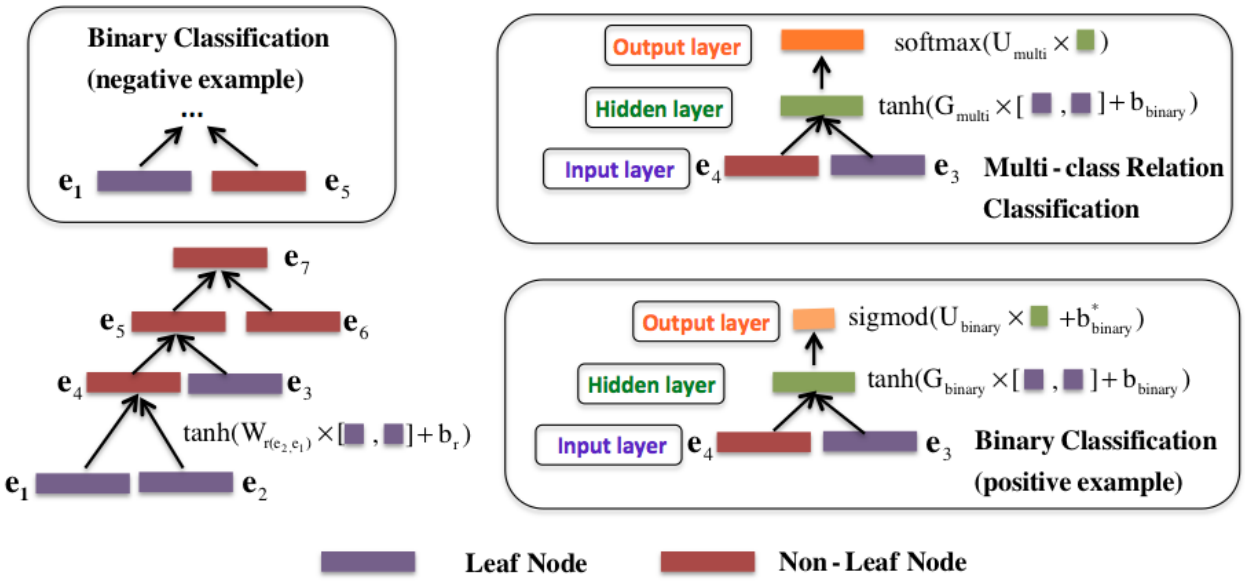

Figure 2: System Overview.

5.3 Distributed Vector for Spans

What is missing in the previous two subsections are the distributed vectors for non-leaf nodes (i.e., e_4 and e_5 in Figure 1), which serve as structure and relation classification. Again, we turn to recursive deep learning network to obtain the distributed vector for each node in the tree in a bottom-up fashion.

Similar as for sentence parse-tree level compositionally, we extend a standard recursive neural network by associating each type of relations r with one specific $K \times 2K$ convolution matrix W_r. The representation for each node within the tree is calculated based on the representations for its children in a bottom-up fashion. Concretely, for a parent node p, given the distributed representation h_{e_i} for left child, h_{e_j} for right child, and the relation $r(e_1, e_2)$, its distributed vector h_p is calculated as follows:

$$h_p = f(W_{r(e_1,e_2)} \cdot [h_{e_i}, h_{e_j}] + b_{r(e_1,e_2)}) \quad (6)$$

where $b_{r(e_1,e_2)}$ is the bias vector and $f(\cdot)$ is the nonlinear tanh function.

To note, our approach does not make any distinction between within-sentence text spans and cross-sentence text spans, different from (Feng and Hirst, 2012; Joty et al., 2013)

5.4 Cost Function

The parameters to optimize include sentence-level convolution parameters $[W, b]$, discourse-level convolution parameters $[\{W_r\}, \{b_r\}]$, binary classification parameters $[G_{\text{binary}}, b_{\text{binary}}, U_{\text{binary}}, b^*_{\text{binary}}]$, and multi-class parameters $[G_{\text{multi}}, b_{\text{multi}}, U_{\text{multi}}]$.

Suppose we have M_1 binary training samples and M_2 multi-class training examples (M_2 equals the number of positive examples in M_1, which is also the non-leaf nodes within the training discourse trees). The cost function for our framework with regularization on the training set is given by:

$$J(\Theta_{\text{binary}}) = \sum_{(e_i,e_j)\in\{\text{binary}\}} J_{\text{binary}}(e_i, e_j) + Q_{\text{binary}} \cdot \sum_{\theta\in\Theta_{\text{binary}}} \theta^2 \quad (7)$$

$$J(\Theta_{\text{multi}}) = \sum_{(e_i,e_j)\in\{\text{multi}\}} J_{\text{multi}}(e_i, e_j) + Q_{\text{multi}} \cdot \sum_{\theta\in\Theta_{\text{multi}}} \theta^2 \quad (8)$$

where

$$\begin{aligned} J_{\text{binary}}(e_i, e_j) &= -t(e_i, e_j) \log p(t(e_i, e_j) = 1) \\ &\quad - (1 - t(e_i, e_j)) \log[1 - p(t(e_i, e_j) = 1)] \\ J_{\text{multi}}(e_i, e_j) &= -\log[p(r(e_i, e_j) = r)] \end{aligned} \quad (9)$$

5.5 Backward Propagation

The derivative for parameters involved is computed through backward propagation. Here we illustrate how we compute the derivative of $J_{\text{multi}}(e_i, e_j)$ with respect to different parameters.

For each pair of nodes $(e_i, e_j) \in$ multi, we associate it with a N_r dimensional binary vector $R(e_i, e_j)$, which denotes the ground truth vector with a 1 at the correct label $r(e_i, e_j)$ and all other entries 0. Integrating softmax error vector, for any parameter θ, the derivative of $J_{\text{multi}}(e_i, e_j)$ with respect to θ is given by:

$$\frac{\partial J_{\text{multi}}(e_i, e_j)}{\partial \theta} = [P_{(e_i,e_j)} - R_{(e_i,e_j)}] \otimes \frac{\partial S_{(e_i,e_j)}}{\partial \theta} \tag{10}$$

where $\otimes$ denotes the Hadamard product between the two vectors. Each training pair recursively back-propagates its error to some node in the discourse tree through $[\{W_r\}, \{b_r\}]$, and then to nodes in sentence parse tree through $[W, b]$, and the derivatives can be obtained according to standard backpropagation (Goller and Kuchler, 1996; Socher et al., 2010).

5.6 Additional Features

When determining the structure/multi relation between individual EDUs, additional features are also considered, the usefulness of which has been illustrated in a bunch of existing work (Feng and Hirst, 2012; Hernault et al., 2010b; Joty et al., 2012). We consider the following simple text-level features:

- Tokens at the beginning and end of the EDUs.
- POS at the beginning and end of the EDUs.
- Whether two EDUs are in the same sentence.

5.7 Optimization

We use the diagonal variant of AdaGrad (Duchi et al., 2011) with minibatches, which is widely applied in deep learning literature (e.g.,(Socher et al., 2011a; Pei et al., 2014)). The learning rate in AdaGrad is adapted differently for different parameters at different steps. Concretely, let g_τ^i denote the subgradient at time step t for parameter θ_i obtained from back-propagation, the parameter update at time step t is given by:

$$\theta_\tau = \theta_{\tau-1} - \frac{\alpha}{\sum_{t=0}^{\tau} \sqrt{g_\tau^{i2}}} g_\tau^i \tag{11}$$

where α denotes the learning rate and is set to 0.01 in our approach.

Elements in $\{W_r\}$, W, G_{binary}, G_{multi}, U_{binary}, U_{multi} are initialized by randomly drawing from the uniform distribution $[-\epsilon, \epsilon]$, where ϵ is calculated as suggested in (Collobert et al., 2011). All bias vectors are initialized with 0. Word embeddings $\{e\}$ are borrowed from Senna (Collobert et al., 2011; Collobert, 2011).

5.8 Inference

For inference, the goal is to find the most probable discourse tree given the EDUs within the document. Existing inference approach basically include the approach adopted in (Feng and Hirst, 2012; Hernault et al., 2010b) that merges the most likely spans at each step and SPADE (Fisher and Roark, 2007) that first finds the tree structure that is globally optimal, then assigns the most probable relations to the internal nodes.

In this paper, we implement a probabilistic CKY-like bottom-up algorithm for computing the most likely parse tree using dynamic programming as are adopted in (Joty et al., 2012; Joty et al., 2013; Jurafsky and Martin, 2000) for the search of global optimum. For a document with n EDUs, as different relations are characterized with different compositions (thus leading to different vectors), we use a $N_r \times n \times n$ dynamic programming table Pr, the cell $Pr[r, i, j]$ of which represents the span contained EDUs from i to j and stores the probability that relation r holds between the two spans within i to j. $Pr[r, i, j]$ is computed as follows:

$$\begin{aligned} Pr[r, i, j] =& \text{max}_{r_1, r_2, k} Pr[r_1, i, k] \cdot Pr[r_2, k, j] \\ & \times P(t_{\text{binary}}(e_{[i,k]}, e_{[k,j]}) = 1) \\ & \times P(r(e_{[i,k]}, e_{[k,j]}) = 1) \end{aligned} \tag{12}$$

At each merging step, a distributed vector for the merged point is calculated according to Eq. 13 for different relations. The CKY-like algorithms finds the global optimal. To note, the worst-case running time of our inference algorithm is O($N_r^2 n^3$), where n denotes the number of sentences within the document, which is much slower than the greedy search. In this work, for simplification, we simplify the framework by maintaining the top 10 options at each step.

6 Experiments

A measure of the performance of the system is realized by comparing the structure and labeling of the RS-tree produced by our algorithm to gold-standard annotations.

Standard evaluation of discourse parsing output computes the ratio of the number of identical tree constituents shared in the generated RS-trees and the gold-standard trees against the total number of constituents in the generated discourse trees[2], which is further divided to three matrices: **Span** (on the blank tree structure), **nuclearity** (on the tree structure with nuclearity indication), and **relation** (on the tree structure with rhetorical relation indication but no nuclearity indication).

The **nuclearity** and **relation** decisions are made based on the multi-class output labels from the deep learning framework. As we do not consider nuclearity when classifying different discourse relations, the two labels *attribute[N][S]* and *attribute[S][N]* made by multi-class classifier will be treated as the same relation label ATTRIBUTE. Also, we do not train a separate classifier for NUCLEUS and SATELLITE identification. The nuclearity decision is made based on the relation type produced by the multi-class classifier.

6.1 Parameter Tuning

The regularization parameter Q constitutes the only parameter to tune in our framework. We tune it on the 347 training documents. Concretely, we employ a five-fold cross validation on the RST dataset and tune Q on 5 different values: 0.01, 0.1, 0.5, 1.5, 2.5. The final model was tested on the testing set after parameter tuning.

6.2 Baselines

We compare our model against the following currently prevailing discourse parsing baselines:

HILDA A discourse parser based on support vector machine classification introduced by Hernault et al. (Hernault et al., 2010b). HILDA uses the binary and multi-class classifier to reconstruct the tree structure in a greedy way, where the most likely nodes are merged at each step. The results for HILDA are obtained by running the system with default settings on the same inputs we provided to our system.

[2]Conventionally, evaluation matrices involve precision, recall and F-score in terms of the comparison between tree structures. But these are the same when manual segmentation is used (Marcu, 2000b).

Approach	Span	Nuclearity	Relation
HILDA	75.3	60.0	46.8
Joty et al.	82.5	68.4	55.7
Feng and Hirst	**85.7**	71.0	58.2
Ji and Eisenstein	82.1	**71.1**	**61.6**
Unified (with feature)	82.0	70.0	57.1
Ours (no feature)	82.4	69.2	56.8
Ours (with feature)	84.0	70.8	58.6
human	88.7	77.7	65.7

Table 1: Performances for different approaches. Performances for baselines are reprinted from (Joty et al., 2013; Feng and Hirst, 2014; Ji and Eisenstein, 2014).

Joty et al The discourse parser introduced by Joty et al. (Joty et al., 2013). It relies on CRF and combines intra-sentential and multi-sentential parsers in two different ways. Joty et al. adopt the global optimal inference as in our work. We reported the performance from their paper (Joty et al., 2013).

Feng and Hirst The linear-time discourse parser introduced in (Feng and Hirst, 2014) which relies on two linear-chain CRFs to obtain a sequence of discourse constituents.

Ji and Eisenstein The shift-reduce discourse parser introduced in (Ji and Eisenstein, 2014) which parses document by relying on the distributed representations obtained from deep learning framework.

Additionally, we implemented a simplified version of our model called **unified** where we use a unified convolutional function with unified parameters $[W_{sen}, b_{sen}]$ for span vector computation. Concretely, for a parent node p, given the distributed representation h_{e_i} for left child, h_{e_j} for right child, and the relation $r(e_1, e_2)$, rather than taking the inter relation between two children, its distributed vector h_p is calculated:

$$h_p = f(W_{sen} \cdot [h_{e_i}, h_{e_j}] + b_{sen}) \qquad (13)$$

6.3 Performance

Performances for different models approaches reported in Table 1. And as we can observe, although the proposed framework obtains comparable result compared with existing state-of-state performances regarding all evaluating parameters for discourse parsing. Specifically, as for the three measures, no system achieves top performance on all three, though some systems outperform all others for one of the measures. The proposed system achieves high overall performance on all three, although it does not achieve top score on any measure. The system gets a little bit performance boost by considering text-level features illustrated in Section 5.6. The simplified version of the original model underperforms against the original approach due to lack of expressive power in convolution. Performance plummets when different relations are uniformly treated, which illustrates the importance of taking into consideration different types of relations in the span convolution procedure.

7 Conclusion

In this paper, we describe an RST-style text-level discourse parser based on a neural network model. The incorporation of sentence-level distributed vectors for discourse analysis obtains comparable performance compared with current state-of-art discourse parsing system.

Our future work will focus on extending discourse-level distributed presentations to related tasks, such as implicit discourse relation identification or dialogue analysis. Further, once the tree structure for a document can be determined, the vector for the entire document can be obtained in bottom-up fashion, as in this paper. One can now investigate whether the discourse parse tree is useful for acquiring a single document-level vector representation, which would benefit multiple tasks, such as document classification or macro-sentiment analysis.

Acknowledgement The authors want to thank Vanessa Wei Feng and Shafiq Joty for helpful discussions regarding RST dataset. We also want to thank Richard Socher, Zhengyan He and Pradeep Dasigi for the clarification of deep learning techniques.

References

Jason Baldridge and Alex Lascarides. 2005. Probabilistic head-driven parsing for discourse structure. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 96–103. Association for Computational Linguistics.

Samuel R Bowman. 2013. Can recursive neural tensor networks learn logical reasoning? *arXiv preprint arXiv:1312.6192*.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2003. *Building a discourse-tagged corpus in the framework of rhetorical structure theory*. Springer.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Ronan Collobert. 2011. Deep learning for efficient discriminative parsing. In *International Conference on Artificial Intelligence and Statistics*, number EPFL-CONF-192374.

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.

David A Duverle and Helmut Prendinger. 2009. A novel discourse parser based on support vector machine classification. 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 665–673. Association for Computational Linguistics.

Vanessa Wei Feng and Graeme Hirst. 2012. Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 60–68. Association for Computational Linguistics.

Vanessa Wei Feng and Graeme Hirst. 2014. A linear time bottom-up discourse parser with constraints and post-editing. In *ACL*.

Seeger Fisher and Brian Roark. 2007. The utility of parse-derived features for automatic discourse segmentation. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 488.

Christoph Goller and Andreas Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *Neural Networks, 1996., IEEE International Conference on*, volume 1, pages 347–352. IEEE.

Hugo Hernault, Danushka Bollegala, and Mitsuru Ishizuka. 2010a. A semi-supervised approach to improve classification of infrequent discourse relations using feature vector extension. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 399–409. Association for Computational Linguistics.

Hugo Hernault, Helmut Prendinger, Mitsuru Ishizuka, et al. 2010b. Hilda: a discourse parser using support vector machine classification. *Dialogue & Discourse*, 1(3).

Wang Houfeng, Longkai Zhang, and Ni Sun. 2013. Improving chinese word segmentation on micro-blog using rich punctuations.

Eduard H Hovy and Elisabeth Maier. 1997. Parsimonious or profligate: How many and which discourse structure relations. *Discourse Processes*.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing.

Shafiq Joty, Giuseppe Carenini, and Raymond T Ng. 2012. A novel discriminative framework for sentence-level discourse analysis. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 904–915. Association for Computational Linguistics.

Shafiq Joty, Giuseppe Carenini, Raymond Ng, and Yashar Mehdad. 2013. Combining intra-and multi-sentential rhetorical parsing for document-level discourse analysis. In *Proceedings of the 51st annual meeting of the association for computational linguistics (ACL)*, pages 486–496.

Dan Jurafsky and James H Martin. 2000. *Speech & Language Processing*. Pearson Education India.

Huong LeThanh, Geetha Abeysinghe, and Christian Huyck. 2004. Generating discourse structures for written texts. In *Proceedings of the 20th international conference on Computational Linguistics*, page 329. Association for Computational Linguistics.

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

Annie Louis, Aravind Joshi, and Ani Nenkova. 2010. Discourse indicators for content selection in summarization. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 147–156. Association for Computational Linguistics.

Minh-Thang Luong, Michael C Frank, and Mark Johnson. 2014. Parsing entire discourses as very long strings: Capturing topic continuity in grounded language learning.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Daniel Marcu. 2000a. The rhetorical parsing of unrestricted texts: A surface-based approach. *Computational Linguistics*, 26(3):395–448.

Daniel Marcu. 2000b. *The theory and practice of discourse parsing and summarization*. MIT Press.

Wenzhe Pei, Tao Ge, and Chang Baobao. 2014. Max-margin tensor neural network for chinese word segmentation. In *Proceedings of ACL*.

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

David Reitter. 2003. Simple signals for complex rhetorics: On rhetorical analysis with rich-feature support vector models. In *LDV Forum*, volume 18, pages 38–52.

Richard Socher, Christopher D Manning, and Andrew Y Ng. 2010. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, pages 1–9.

Richard Socher, Eric H Huang, Jeffrey Pennington, Andrew Y Ng, and Christopher D Manning. 2011a. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *NIPS*, volume 24, pages 801–809.

Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. 2011b. 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, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642.

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.

Caroline Sporleder and Mirella Lapata. 2005. Discourse chunking and its application to sentence compression. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 257–264. 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.

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. ACM.

Longkai Zhang Houfeng Wang and Xu Sun Mairgup Mansur. 2013. Exploring representations from unlabeled data with co-training for chinese word segmentation.