# Exploring Joint Neural Model for Sentence Level Discourse Parsing and Sentiment Analysis

**Bita Nejat** **Giuseppe Carenini** **Raymond Ng**
Department of Computer Science, University of British Columbia
Vancouver, BC, V6T 1Z4, Canada
{nejatb, carenini, rng}@cs.ubc.ca

## Abstract

Discourse Parsing and Sentiment Analysis are two fundamental tasks in Natural Language Processing that have been shown to be mutually beneficial. In this work, we design and compare two Neural models for jointly learning both tasks. In the proposed approach, we first create a vector representation for all the text segments in the input sentence. Next, we apply three different Recursive Neural Net models: one for discourse structure prediction, one for discourse relation prediction and one for sentiment analysis. Finally, we combine these Neural Nets in two different joint models: Multi-tasking and Pre-training. Our results on two standard corpora indicate that both methods result in improvements in each task but Multi-tasking has a bigger impact than Pre-training. Specifically for Discourse Parsing, we see improvements in the prediction on the set of contrastive relations.

## 1 Introduction

This paper focuses on studying two fundamental NLP tasks, Discourse Parsing and Sentiment Analysis. The importance of these tasks and their wide applications (e.g., (Gerani et al., 2014), (Rosenthal et al., 2014)) has initiated much interest in studying both, but no method yet exists that can come close to human performance in solving them.

Discourse parsing is the task of parsing a piece of text into a tree (called a Discourse Tree), the leaves of which are typically clauses (called Elementary Discourse Units or EDUs in short) and nodes (Discourse Units) represent text spans that are concatenations of their corresponding subtrees' text spans [1]. Nodes also have labels identifying discourse relationships ("*contrast*", "*evidence*", etc.) between their corresponding subtrees. The relation also specifies nucliearity of the children. Nuclei are the core parts of the relation and Satellites are the supportive ones.

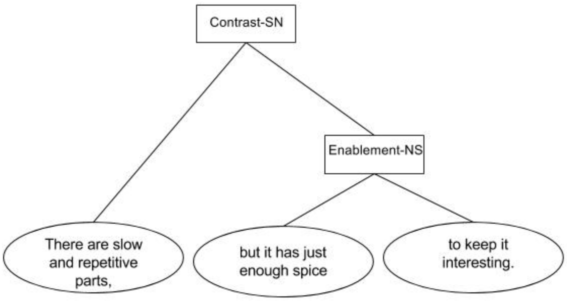
Contrast-SN
Enablement-NS
There are slow and repetitive parts,
but it has just enough spice
to keep it interesting.

Figure 1: The Discourse Tree of a sentence from Sentiment Treebank dataset

A Relation can take one of the following forms: (1) Satellite-Nucleus: First Discourse Unit is Satellite and second Discourse Unit is Nucleus. (2) Nucleus-Satellite: First Discourse Unit is Nucleus and second Discourse Unit is Satellite. (3) Nucleus-Nucleus: Both Discourse Units are Nuclei. In this approach relation identification and nuclearity assignment is done simultaneously. Figure 1 shows the Discourse Tree of a sample sentence. In this sentence, the Discourse Unit "There are slow and repetitive parts," holds a "*Contrast*" relationship with "but it has just enough spice to keep it interesting.". Furthermore, we can see that the former Discourse Unit is the satellite of the relation and the later part is the Nucleus.

Discourse Parsing is such a critical task in NLP because previous work has shown that information

[1] A text span is a piece of text consisting of one or more clauses (or EDUs).

*Proceedings of the SIGDIAL 2017 Conference*, pages 289–298,
Saarbrücken, Germany, 15-17 August 2017. ©2017 Association for Computational Linguistics

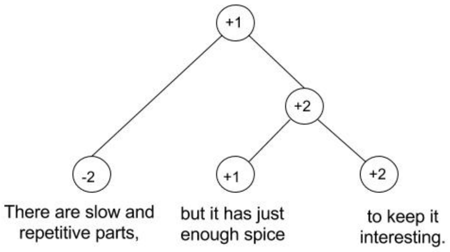
+1
+2
-2
+1
+2
There are slow and repetitive parts,
but it has just enough spice
to keep it interesting.

Figure 2: The Sentiment annotation (over Discourse Tree structure) of a sentence from Sentiment Treebank dataset

contained in the resulting Discourse Tree can benefit many other NLP tasks including but not restricted to automatic summarization (e.g., (Gerani et al., 2014), (Marcu and Knight, 2001), (Louis et al., 2010)), machine translation (e.g., (Meyer and Popescu-Belis, 2012),(Guzmán et al., 2014)) and question answering (e.g., (Verberne et al., 2007)). In contrast to traditional syntactic and semantic parsing, Discourse Parsing can generate structures that cover not only a single sentence but also multi-sentential text. However, the focus of this paper is on sentence level Discourse Parsing, leaving the study of extensions to multi-sentential text as future work.

The second fundamental task we consider in this work is assigning a contextual polarity label to text (sentiment analysis). Analyzing the overall polarity of a sentence is a challenging task due to the ambiguities that can be introduced by combinations of words and phrases. For example in the movie review excerpt shown in Figure 2, the phrase "There are slow and repetitive parts" has a negative sentiment. However when it is combined with the positive phrase "but it has just enough spice to keep it interesting", it results in an overall positive sentence.

It has been suggested that the information extracted from Discourse Trees can help with Sentiment Analysis (Bhatia et al., 2015) and likewise, knowing the sentiment of two pieces of text might help with the identification of discourse relationships between them (Lazaridou et al., 2013). For instance, taking the sentence in Figure 1 as an example, knowing that the two text spans "There are slow and repetitive parts" and "but it has just enough spice to keep it interesting" are in a *Contrast* relationship to each other, also signals that the sentiment of the two text spans is less likely to be of the same type[2]. Likewise, knowing that the sentiment of the former text span is "very negative", while the sentiment of the later text span is "very positive", helps to narrow down the choice of discourse relation between these two text spans to the *Contrastive* group which contains relations *Contrast*, *Comparison*, *Antithesis*, *Antithesis-e*, *Consequence-s*, *Concession* and *Problem-Solution*.

To the best of our knowledge there is no previous work that learns both of these tasks in a joint model, using deep learning architectures. The main contribution of this paper is to address this gap by investigating how the two tasks can benefit from each other at the sentence level within a deep learning joint model. More specific contributions include:

(i) The development of three independent recursive neural nets: two for the key sub-tasks of discourse parsing, namely structure prediction and relation prediction; the third net for sentiment prediction.

(ii) The design and experimental comparison of two alternative neural joint models, Multi-tasking and Pre-training, that have been shown to be effective in previous work for combining other tasks in NLP (e.g., (Collobert and Weston, 2008),(Erhan et al., 2010),(Liu et al., 2016a)).

Our results indicate that a joint model performs better than individual models in either of the tasks with Multi-tasking outperforming Pre-training. Upon closer inspection, we also find that the improvement of Multi-tasking system in Relation prediction is mainly for the Contrastive set of relations, which confirms our hypothesis that knowing the sentiment of two text spans can help narrow down the choice of discourse relations that holds between them.

## 2 Previous Work

Traditionally, **Discourse Parsing and Sentiment Analysis** have been approached by applying machine learning methods with predetermined, engineered features that were carefully chosen by studying the properties of the text.

[2] Contrast can also hold between factual clauses as in [But from early on, Tigers workers unionized,] and [while Federals never have.] (wsj_1394 from RST-DT).

Examples of effective sentence level and document level Discourse Parsers include CODRA (Joty et al., 2015) and the parser of (Feng and Hirst, 2014) . These parsers use organizational, structural, contextual, lexical and N-gram features to represent Discourse Units and apply graphical models for learning and inference (i.e. Conditional Random Fields). The performance of these parsers critically depends on a careful selection of informative and relevant features, something that is instead performed automatically in the neural models we propose in this paper.

(Nakagawa et al., 2010), (Pang et al., 2008) and (Rentoumi et al., 2010), approach Sentiment Analysis using carefully engineered features as well as polarity rules. The choice of features also plays a key role in the high performance of these models.

Yet, with the rapid advancements of Neural Nets, there has been increased interest in applying them to different NLP tasks. (Socher et al., 2013) approached the problem of Sentiment Analysis by recursively assigning sentiment labels to the nodes of a binarized syntactic parse tree over a sentence. At each non-leaf node, the Sentiment Neural Net first creates a distributed embedding for the node using the embedding of its two children and then assigns a sentiment label to that node. Their approach achieves state of the art results. In our work, we borrow from the same idea of Recursive Neural Nets to learn the Sentiment labels. However, the structure over which we learn the Sentiment labels is the Discourse Tree of the sentence as opposed to the syntactic parse tree, with the goal of testing if Sentiment Analysis can benefit directly from discourse information within a neural joint model.

Motivated by Socher's success on Sentiment Analysis, (Li et al., 2014) approached the problem of Discourse Parsing by recursively building the Discourse Tree using two Neural Nets. A Structure Neural Net decides whether two nodes should be connected in the Discourse Tree or not. If two nodes are determined to be connected by the Structure Neural Net, a Relation Neural Net then decides what rhetorical relation should hold between the two nodes. Their approach also yields promising results. In terms of representation, the recursive structure of a Discourse Tree is used to learn the embedding of each non-leaf node from its children. For leaf nodes (EDUs), the representation is learned recursively using the syntactic parse tree of the node. One problem with their work is that it is unclear how they combine the labeled Discourse Structure Tree with the unlabeled syntactic parse trees to learn the vector representations for the text spans.

(Bhatia et al., 2015) trained a Recursive Neural Network for Sentiment Analysis over a Discourse Tree and showed that the information extracted from the Discourse Tree can be helpful for determining the Sentiment at document level. In their work however, they did not attempt to learn a distributed representation for the sub-document units. To represent EDUs, they used the bag-of-words features. For our work, we not only apply a Recurrent Neural Net approach to learn embeddings for the EDUs, but we also jointly learn models for the two tasks, instead of simply feeding a pre-computed discourse structure in a neural model for sentiment.

**Learning text embeddings** is a fundamental step in using Neural Nets for NLP tasks. An embedding is a fixed dimensional representation of the data (text) without the use of handpicked features. As words are the building blocks of text, previous studies have created fixed dimensional vector representations for words (Mikolov et al., 2013) that capture syntactic and semantic properties of the words. However, creating meaningful fixed dimensional vector representations for text spans is an ongoing challenge.

Both (Socher et al., 2013) and (Li et al., 2014) learn the embedding of a text span in a recursive manner, given a binary tree over the text span with leaves being the words. The embedding of a parent is computed from the embedding of its two children using a non-linear projection. The embedding is then used for training the task under study (Sentiment Analysis and Discourse Parsing respectively) and updated according to how useful it was for the task.

Recently Recurrent Neural Nets (RNNs) have become a more popular alternative for learning the embedding of a sentence (Kiros et al., 2015). In this setting, an encoder RNN encodes a sentence into a fixed vector representation that is then used by a decoder RNN to predict the following and preceding sentences and based on how good the predictions were, updates both the decoder and encoder RNNs. Once training is done, the encoder RNN can be used to create an embedding for any text span. In this paper, we have used the encoder

RNN to represent our EDUs, but we further compress the resulting embeddings with a neural based compressor to limit the number of parameters.

When training a neural model, the weights are usually initialized with random numbers taken from a uniform distribution. However, in their work, (Erhan et al., 2010) argue that **Pre-training** a neural model results in better generalization and can enhance the performance of the model. More recently, this general idea has been successfully applied in several scenarios (e.g., (Chung et al., 2015), (Seyyedsalehi and Seyyedsalehi, 2015) ). In our work, we use the trained weights of one neural model (e.g. sentiment) as an initialization form for another task (e.g. discourse structure) to see if the features learned for one can be helpful for the other.

Neural **Multi-tasking** was originally proposed by (Collobert and Weston, 2008), who experimented with the technique using deep convolutional neural networks. In essence, the basic idea is that a network is alternatively trained with instances for different tasks, so that the network is learning to perform all these tasks jointly. In (Collobert and Weston, 2008) a model is trained to perform a variety of predictions on a given sentence, including part-of-speech tags, chunks, named entity tags, semantic roles, semantically similar words and the likelihood that the sentence makes sense using a language model. They showed that multitasking using a neural net structure can improve the generalization of the shared tasks and result in better performance. Following up on this initial success, many researchers have applied the neural multi-tasking strategy to several tasks, including very recent work in vision (Kaneko et al., 2016) and NLP (e.g., text classification (Liu et al., 2016a) and the classification of implicit discourse relations (Liu et al., 2016b)).

## 3 Corpora

For the task of Discourse Parsing, we use RST-DT ((Carlson and Marcu, 2001), (Carlson et al., 2002)). This dataset contains 385 documents along with their fully labeled Discourse Trees. The annotation is based on the Rhetorical Structure Theory (RST), a popular theory of discourse originally proposed in (Mann and Thompson, 1988). All the documents in RST-DT were chosen from Wall Street Journal news articles taken from the Penn Treebank corpus (Marcus et al., 1993). Since we are focusing only on sentence-level discourse parsing, the documents as well as their Discourse Trees were first preprocessed to extract the sentences and sentence-level Discourse Trees. The sentence-level Discourse Trees were extracted from the document-level Discourse Tree by finding the sub-tree that exactly spans over the sentence. This resulted in a dataset of 6846 sentences with well-formed Discourse Trees, out of which 2239 sentences had only one EDU. Since sentences with only one EDU have trivial Discourse Trees, these sentences were excluded from our dataset, leaving a total of 4607 sentences.

For the task of Sentiment Analysis, we use the Sentiment Treebank (Socher et al., 2013). This dataset consists of 11855 sentences along with their syntactic parse trees annotated with sentiment labels at each node. For this work, since our models label sentiment over a Discourse Tree, we had to preprocess the Sentiment Treebank in the following way. For each sentence in the dataset, a Discourse Tree was created using (Joty et al., 2015). Next, for each node of the discourse tree, a sentiment label was extracted from the corresponding labeled syntactic tree by finding a subtree that exactly (or almost exactly [3]) matches the text span represented by the node in the discourse tree.

## 4 Proposed Joint Model

Our framework consists of three main sub parts. Given a segmented sentence, the first step is to create meaningful vector representations for all the EDUs. Next, we devise three different Recursive Neural Net models, each designed for one of discourse structure prediction, discourse relation prediction and sentiment analysis. Finally, we join these Neural Nets in two different ways: Multitasking and Pre-training. Below, we discuss each of these steps in more detail.

### 4.1 Learning Text Embeddings

One of the most challenging aspects of designing effective Neural Nets is to have meaningful representations for the inputs. Our inputs to the Neural Nets are text spans consisting of multiple words. Initially, we considered directly applying the Skip-

[3] Exact match was not possible when the syntactic and the discourse structures were not fully aligned, which happened in 31.9% of the instances. In this case, an approximation of the sentiment was computed by considering the sentiment of the two closest subsuming and subsumed syntactic sub-trees.

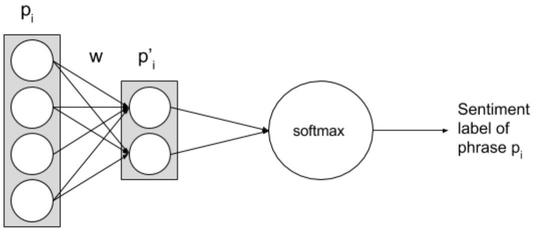

Figure 3: The Sentiment Neural Compressor

thought framework (Kiros et al., 2015) to each text span to get a generic vector representations for them, since the original Skip-thought vectors were shown in (Kiros et al., 2015) to be useful for many NLP tasks. However, given the size of our datasets (only in the thousands of instances), it was clear that using 4800-dimensional Skip-thought would have created an over-parametrized network prone to over-fitting. Based on this observation, in order to simultaneously reduce the dimensionality and to produce vectors that are meaningful for our tasks, we devised a compression mechanism that takes in the Skip-thought produced vectors and compresses them using a Neural Net. Figures 4 and 3 show the structure of these compressors for our two different tasks. Each compressor is learned on the training set used for that task.

The sentiment neural compressor (Figure 3) takes as input, the skip-thought produced vector representations for all phrases in the Sentiment Treebank. For example, consider a phrase $i$ with skip-thought produced vector $P_i \in R^{4800}$. The Sentiment Neural Compressor learns compressed vector $P'_i \in R^d$ through

$$P'_i = f(W.P_i) \tag{1}$$

where $f$ is a non-linear activation function such as $relu$ and $W \in R^{d\times 4800}$ is the matrix of weights. This Neural Net uses the sentiment of phrase $i$ for supervised learning of the weights.

Similarly, the Discourse Parsing neural compressor (Figure 4) takes the skip-thought produced vector representations for two EDUs $e_i$, $e_j$ and learns the compressed vectors $e'_i$ and $e'_j$, each with $d$ dimensions where

$$\begin{aligned} e'_i &= f(W_1.e_i) \\ e'_j &= f(W1.e_j) \end{aligned} \tag{2}$$

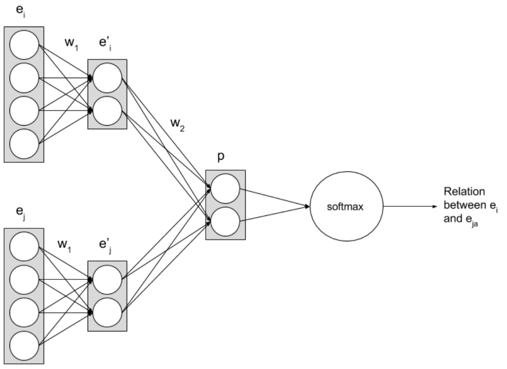

Figure 4: The Discourse Neural Compressor

where $f$ is again a non-linear activation function such as $relu$ and $W_1 \in R^{d\times 4800}$ is the matrix of weights. Note that the same set of weights are used for both EDUs because we are looking for a unique set of weights to compress an EDU.

### 4.2 Neural Net Models

Following (Socher et al., 2013)'s idea of Sentiment Analysis using recursive Neural Nets, we designed three Recursive Neural Nets for each task of Discourse Structure prediction, Discourse Relation prediction and Sentiment Analysis. All these three Neural Nets are classifiers.

The Structure Neural Net takes in the compressed vector representation ($\in R^d$) for two Discourse Units and learns whether they will be connected in the Discourse Tree (Figure 5). In this process, it also learns the vector representation for the parent of these two children. So for a parent $p$ with children $c_l$ and $c_r$, the vector representation for the parent is obtained by:

$$p = f(W_{str}[c_l, c_r] + b_{str}) \tag{3}$$

where $[c_l, c_r]$ denotes the concatenating vector for the children; $f$ is a non-linearity function; $W_{str} \in R^{d\times 2d}$ and $b_{str} \in R^d$ is the bias vector.

The Relation Neural Net takes as input the compressed vector representation for two Discourse Units that are determined to be connected in the Discourse Tree and learns the relation label for the parent node. The Relation Neural Net is the same in structure as the Structure Neural Net in Figure 5.

The Sentiment Neural Net takes as input the compressed vector representation for two Dis-

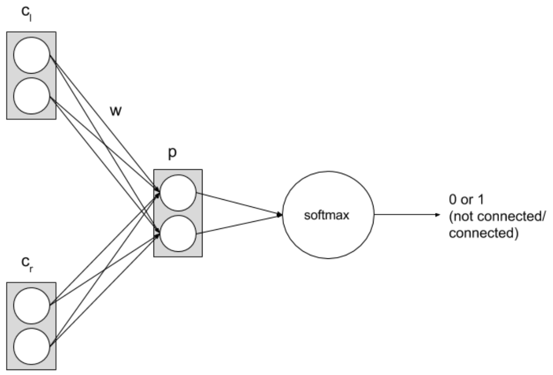

Figure 5: The Discourse Structure Neural Net

course Unit that are determined to be connected in the Discourse Tree and learns the sentiment label for the parent node. This Neural net also shares the same structure as the one in Figure 5.

### 4.3 Joining Neural Nets

Our hypothesis in creating a joint model is that the accuracy of prediction obtained in a joint design would be higher than the accuracy of prediction coming from independent Neural Nets applied to each task. We explore two ways of creating a joint model. For both approaches, we train three neural nets (Discourse Structure, Discourse Relation and Sentiment Neural Nets) that interact with one another for improved training. The input to the Structure net are all possible pairs of text spans that can be connected in a Discourse Tree. The input to the Relation and Sentiment nets are the pairs of text spans that are determined to be connected by the Structure net.

Inspired by **Multitasking** (Collobert and Weston, 2008), our goal is to find a representation for the input that will benefit all the tasks that need to be solved. Since the first layer in a Neural Net learns relevant features from the input embedding, in this approach, the first layer is shared between the three Neural Nets and training is achieved in a stochastic manner by looping over the three tasks. As shown in Figure 6, at each time step, one of the tasks is selected along with a random training example for that task. Afterwards, the neural net corresponding to this task is updated by taking a gradient step with respect to the chosen example. The end product of this design is a joint input representation that could benefit both Sentiment Analysis and Discourse Parsing.

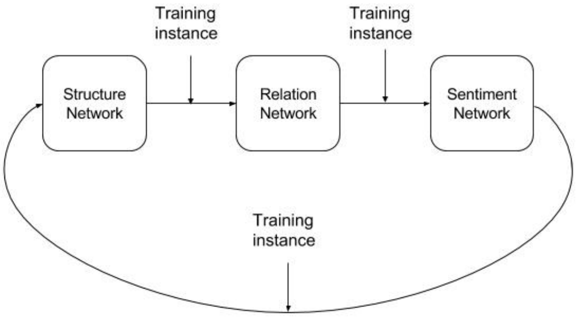

Figure 6: Multi-tasking

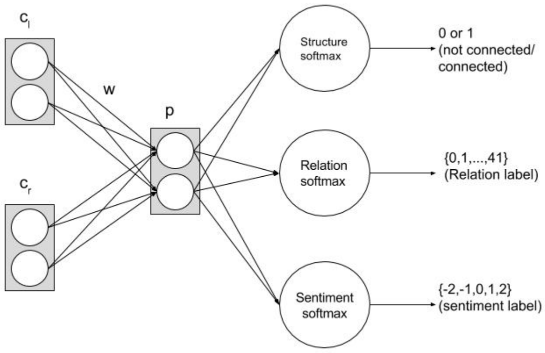

Figure 7: Multi-tasking Network

Inspired by **Pre-training Neural Nets** (Erhan et al., 2010), in this approach we study how the parameters of one Neural Net after training can be used as a form of initialization for the network applied to the other task. As shown in Figure 8, in this setting, we first fully train the Discourse Structure Neural Net, then the weights from this trained net are used to initialize the Discourse Relation Neural Net and once this net is fully trained, its weights are used to initialize the weights of the Discourse Structure Neural Net again. After another round of training the Discourse Structure Neural Net, its weights are used to initialize the Sentiment Neural Net. After training the Sentiment Neural Net, its weights are again used to initialize the Structure Neural Net. [4]

## 5 Training and Evaluating the Models

All the neural models presented in this paper were implemented using the TensorFlow python pack-

[4] We experimented with 2,3 and 10 iterations using 10-fold cross validation on the datasets and achieved best results with 3 iterations, which appears to be a good compromise between accuracy and training time.

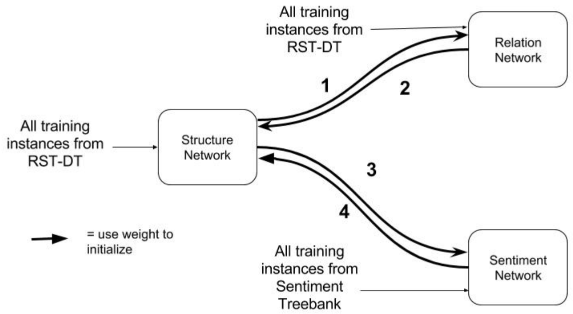
All training instances from RST-DT
Relation Network
1
2
All training instances from RST-DT
Structure Network
3
4
= use weight to initialize
All training instances from Sentiment Treebank
Sentiment Network

Figure 8: Using the weights of one network as a form of pre-training for another network

| Approach | Span | Nuclearity | Relation |
|---|---|---|---|
| Discourse Parser (Before Joining) | 93.37 | 73.38 | 57.05 |
| Joined Model Pre-training | **94.35** | 74.92 | 58.82 |
| Joined Model Multi-tasking | 94.31 | **75.91*** | **60.91*** |

Table 1: Discourse Parsing results based on manual discourse segmentation

| Setting<br>Relation | Individual | Pre-training | Multi-tasking |
|---|---|---|---|
| Comparison | 18.97 | 20.87 | 27.08 |
| Contrast | 15.19 | 17.74 | 20.83 |
| Cause | 7.6 | 8.11 | 8.61 |
| Average | 13.92 | 15.57 | 18.84 |

Table 2: Contrastive Relation Prediction results under different training settings

age (Abadi et al., 2015). We minimize the cross-entropy error using the Adam optimizer and L2-regularization on the set of weights. For the individual models (before joining), we use 200 training epochs and a batch size of 100.

We evaluate our models using 10-fold cross validation on the sentiment treebank and on RST-DT. In Table 1 and Table 3, a star indicates that there is statistical significance with a p-value less than 0.05. The significance is with respect to the joint model vs the model before joining. The results for Discourse Parsing are shown in Table 1. To build the most probable tree, a CKY-like bottom-up parsing algorithm that uses dynamic programming to compute the most likely parses is applied (Joty et al., 2015) and we have used the 41 relations outlined in (Mann and Thompson, 1988) for training and evaluation of the Relation prediction. From the results, we see some improvement on Discourse Structure prediction when we are using a joint model but the improvement is statistically significant only for the Nuclearity and Relation predictions. The improvements on the Relation predictions were mainly on the Contrastive set (Bhatia et al., 2015), specifically the class of *Contrast*, *Comparison* and *Cause* relations as defined in (Mann and Thompson, 1988). The result for each of these relations under different training settings are shown in Table 2. Notice that the accuracies may seem low, but because we train over 41 classes of relations, a random prediction would result in 2.43% accuracy. Among the contrastive relations, the *Problem-Solution* did not improve due to the fact that this relation is hardly seen at the sentence level. This confirms our hypothesis that knowing the sentiment of the two Discourse Units that are connected in a discourse tree can help with the identification of the discourse relation that holds between them.

For the task of Sentiment Analysis, the results are shown in Table 3. To train the model, we use the five classes of sentiment used in (Socher et al., 2013)[5]. We measure the accuracy of prediction in two different settings. In the fine grained setting we compute the accuracy of exact match across five classes. In the Positive/Negative setting, if the prediction and the target had the same sign, they were considered equal. Notice that this is different from training a classifier for binary classification, which is a much easier task (see (Bhatia et al., 2015)). The difference in accuracy between these two settings signals that distinguishing between *very positive* and *positive* and distinguishing between *very negative* and *negative* is rather hard. The results of sentiment shown in Table 3 are also consistent with our hypothesis. When jointly trained with Discourse Parsing, we can get a better performance on labeling nodes of the Discourse Tree with sentiment labels compared to an individual sentiment analyzer applied to a Discourse Tree.

Interestingly, if we compare the two joint models across both tasks it appears that Multi-tasking does better that Pre-training in all cases except for discourse structure. A possible explanation is that by transferring weights from one network to another (as done in Pre-training), the neural net starts learning with a possibly better initialization of the weights. However Multi-tasking performs a joint

[5] {*very negative, negative, neutral, positive, very positive*}

| Approach | Fine grained All | Fine grained Root | Positive Negative All | Positive Negative Root |
|---|---|---|---|---|
| Sentiment Analyzer (Before Joining) | 43.37 | 40.6 | 52.86 | 51.27 |
| Joined Model Pre-training | 42.46 | 40.36 | 53.82 | 53.15 |
| Joined Model Multi-tasking | **45.49*** | **44.82*** | **55.52*** | **54.72*** |

Table 3: Sentiment Analysis over Discourse Tree

learning at the finer granularity of single training instances and so an improvement in learning one task immediately affects the next.

All results in Table 1 and 3 were obtained by setting the dimension $d$ of the compressed vectors to 100. Experimentally, we found that the performance of the model was rather stable for $d \in \{1200, 600, 300, 100\}$ and was substantially lower for $d \in \{50, 25\}$.

In terms of actual runtime, Pre-training and the individual models are an order of magnitude faster than the Multi-tasking model. This is because even though they require a larger number of epochs to converge (200 for individual, vs 6 for Multi-tasking), they can be trained in parallel. Notice that training and testing of the networks is done on Sentiment Treebank for sentiment analysis and on RST-DT for discourse parsing. (Joty et al., 2015)'s Discourse parser was run on Sentiment Treebank to get the sentiment annotation at the granularity required for the joint model with discourse. However, having a gold dataset of sentiment labels corresponding to discourse units could further improve the results.

## 6 Comparison With Previous Work

Several differences between this work and previous approaches make direct comparisons challenging and possibly not very informative.

(Socher et al., 2013) use syntactic trees, as opposed to discourse trees, as recursive structures for training. Thus we cannot compare with his ”All”-level results. For ”Root”-level, (Socher et al., 2013) reports 45.7% fine-grained sentiment accuracy compared to 44.82% of our Multi-tasking. This difference is unlikely to be significant and the sentiment annotation of syntactic structure is definitely more costly than one at the EDU level.

(Bhatia et al., 2015) focuses on document level sentiment analysis, using bag-of-word features for EDUs; and only training a binary model while assuming the discourse tree as given, which is very different from our approach.

Since our work focuses on sentence-level discourse parsing, we cannot compare with (Li et al., 2014) because they only report overall results without differentiating sentence vs document level.

Finally, (Joty et al., 2015) achieves better performance on sentence level. First, we believe that with more training data, as it has been shown with other NLP tasks, we would eventually outperform CODRA. Second, the goal of our work is not to beat the state of the art on each single task, but to show how the two tasks can be jointly performed in a neural model.

## 7 Conclusion

Discourse Parsing and Sentiment Analysis are two fundamental NLP tasks that have been shown to be mutually beneficial. Evidence from previous work indicates that information extracted from Discourse Trees can help with Sentiment Analysis and likewise, knowing the sentiment of two pieces of text can help with identification of discourse relationships between them. In this paper, we show how synergies between these two tasks can be exploited in a joint neural model. The first challenge entailed learning meaningful vector representations for text spans that are the inputs for the two tasks. Since the dimension of vanilla skip-thought vectors is too high compared to the size of our corpora, in order to simultaneously reduce the dimensionality and to produce vectors that are meaningful for our tasks, we devised task specific neural compressors, that take in Skip-thought vectors and produce much lower dimensional vectors.

Next, we designed three independent Recursive Neural Nets classifiers; one for Discourse Structure prediction, one for Discourse Relation prediction and one for Sentiment Analysis. After that, we explored two ways of creating joint models from these three networks: Pre-training and Multitasking. Our experimental results show that such models do capture synergies among the three tasks with the Multi-tasking approach being the most successful, confirming that latent Discourse features can help boost the performance of a neural sentiment analyzer and that latent Sentiment features can help with identifying contrastive relations between text spans.

In the short term, we plan to verify how syntactic information could be explicitly leveraged in the three task-specific networks as well as in the joint models. Then, our investigation will move from making predictions about a single sentence to the much more challenging task of dealing with multi-sentential text, which will likely require not only more complex models, but also models with scalable time performance in both learning and inference. Next, we intend to study how pre-training and multitasking could be both exploited simultaneously in the same model, something that to the best of our knowledge has not been tried before. Finally, as another venue for future research, we plan to explore how sentiment analysis and discourse parsing could be modeled jointly with text summarization, since these three tasks can arguably inform each other and therefore benefit from joint neural models similar to the ones described in this paper.

## References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. http://tensorflow.org/.

Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from rst discourse parsing. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*. http://www.aclweb.org/anthology/D/D15/D15-1263.pdf.

Lynn Carlson and Daniel Marcu. 2001. Discourse tagging reference manual. *ISI Technical Report ISI-TR-545* 54:56.

Lynn Carlson, Mary Ellen Okurowski, and Daniel Marcu. 2002. *RST discourse treebank*. Linguistic Data Consortium, University of Pennsylvania.

Yu-An Chung, Hsuan-Tien Lin, and Shao-Wen Yang. 2015. Cost-aware pre-training for multi-class cost-sensitive deep learning. *arXiv preprint arXiv:1511.09337* .

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. ACM, pages 160–167.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research* 11(Feb):625–660.

Vanessa Wei Feng and Graeme Hirst. 2014. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 511–521. http://www.aclweb.org/anthology/P14-1048.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *EMNLP*. pages 1602–1613.

Francisco Guzmán, Shafiq R Joty, Lluís Màrquez, and Preslav Nakov. 2014. Using discourse structure improves machine translation evaluation. In *ACL (1)*. pages 687–698.

Shafiq Joty, Giuseppe Carenini, and Raymond T Ng. 2015. Codra: A novel discriminative framework for rhetorical analysis. *Computational Linguistics* .

Takuhiro Kaneko, Kaoru Hiramatsu, and Kunio Kashino. 2016. Adaptive visual feedback generation for facial expression improvement with multi-task deep neural networks. In *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, pages 327–331.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*. pages 3294–3302.

Angeliki Lazaridou, Ivan Titov, and Caroline Sporleder. 2013. A bayesian model for joint unsupervised induction of sentiment, aspect and discourse representations. In *ACL (1)*. pages 1630–1639.

Jiwei Li, Rumeng Li, and Eduard H Hovy. 2014. Recursive deep models for discourse parsing. In *EMNLP*. pages 2061–2069.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016a. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101* .

Yang Liu, Sujian Li, Xiaodong Zhang, and Zhifang Sui. 2016b. Implicit discourse relation classification via multi-task neural networks. *arXiv preprint arXiv:1603.02776* .

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*. Association for Computational Linguistics, pages 147–156.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse* 8(3):243–281.

Daniel Marcu and Kevin Knight. 2001. Discourse parsing and summarization. US Patent App. 09/854,301.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics* 19(2):313–330.

Thomas Meyer and Andrei Popescu-Belis. 2012. Using sense-labeled discourse connectives for statistical machine translation. In *Proceedings of the Joint Workshop on Exploiting Synergies between Information Retrieval and Machine Translation (ESIRMT) and Hybrid Approaches to Machine Translation (HyTra)*. Association for Computational Linguistics, pages 129–138.

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

Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. 2010. Dependency tree-based sentiment classification using crfs with hidden variables. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 786–794.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval* 2(1–2):1–135.

Vassiliki Rentoumi, Stefanos Petrakis, Manfred Klenner, George A Vouros, and Vangelis Karkaletsis. 2010. United we stand: Improving sentiment analysis by joining machine learning and rule based methods. In *LREC*.

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*. Dublin, Ireland, pages 73–80.

Seyyede Zohreh Seyyedsalehi and Seyyed Ali Seyyedsalehi. 2015. A fast and efficient pre-training method based on layer-by-layer maximum discrimination for deep neural networks. *Neurocomputing* 168:669–680.

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. *Proceedings of the conference on empirical methods in natural language processing (EMNLP)* 1631:1642.

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*. ACM, pages 735–736.