EMNLP 2019

Motivation: DST Corpus is small, annotating is expensive

DST is a fundamental task can be use to downstream task like sentiment analysis

So maybe sentiment info can be use to DST task

Predicting Discourse Structure using Distant Supervision from Sentiment

Patrick Huber and Giuseppe Carenini
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada, V6T 1Z4
{huberpat, carenini}@cs.ubc.ca

Abstract

Discourse parsing could not yet take full advantage of the neural NLP revolution, mostly due to the lack of annotated datasets. We propose a novel approach that uses distant supervision on an auxiliary task (sentiment classification), to generate abundant data for RST-style discourse structure prediction. Our approach combines a neural variant of multiple-instance learning, using document-level supervision, with an optimal CKY-style tree generation algorithm. In a series of experiments, we train a discourse parser (for only structure prediction) on our automatically generated dataset and compare it with parsers trained on human-annotated corpora (news domain RST-DT and Instructional domain). Results indicate that while our parser does not yet match the performance of a parser trained and tested on the same dataset (*intra-domain*), it does perform remarkably well on the much more difficult and arguably more useful task of *inter-domain* discourse structure prediction, where the parser is trained on one domain and tested/applied on another one.

1 Introduction

Discourse parsing is a fundamental NLP task known to enhance key downstream tasks, such as sentiment analysis (Bhatia et al., 2015; Nejat et al., 2017; Hogenboom et al., 2015), text classification (Ji and Smith, 2017) and summarization (Gerani et al., 2014).

In essence, a discourse parser should reveal the structure underlying coherent text as postulated by a discourse theory, of which the two most popular are Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) and PDTB (Prasad et al., 2008). In this paper, we focus on RST-style parsing, but the proposed approach is theory agnostic and could be applied to PDTB as well.

The RST discourse theory assumes a complete hierarchical discourse tree for a given document, where leaf nodes are clause-like sentence fragments, called elementary-discourse-units (EDUs), while internal tree nodes are labelled with discourse relations. In addition, each node is given a nuclearity attribute, which encodes the importance of the node in its local context.

In the past decade, traditional, probabilistic approaches, such as Support Vector Machines (SVM) (Hernault et al., 2010; Ji and Eisenstein, 2014) and Conditional Random Fields (CRF) (Joty et al., 2015; Feng and Hirst, 2014), have dominated the field. More recently, neural approaches (Braud et al., 2016; Li et al., 2016; Braud et al., 2017; Yu et al., 2018; Liu and Lapata, 2017, 2018) have been explored, with limited success (Morey et al., 2017; Ferracane et al., 2019). The main reason why recent advances in deep learning have not enhanced discourse parsing to the same extend as they have revolutionized many other areas of NLP is the small amount of training data available. Existing corpora in English (Carlson et al., 2002; Subba and Di Eugenio, 2009) only comprise of a few hundred annotated documents, each typically containing a few dozen EDUs, strictly limiting the application of deep learning methodologies. Although in principle new corpora could be created, the annotation process is expensive and time consuming. It requires sophisticated linguistic expertise and is therefore not suitable for crowd-sourcing efforts.

Another limiting issue with the available training data is the restriction to only a few domains, such as news articles (Carlson et al., 2002) or instructions (Subba and Di Eugenio, 2009). This impairs the performance of existing discourse parsers when transferred into new domains.

To overcome the mentioned limitations, we propose a novel approach that uses distant supervi-

1. Using a sentiment corpus to Train a Sentiment SOTA Model
2. Using the Sentiment Model Hidden -> Generate CKY style tree
3. Evaluation in inter-domain DST task

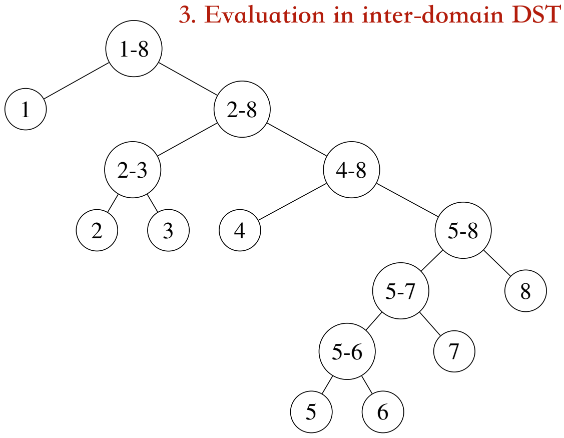

Figure 1: Example output of a strongly negative restaurant review in the Yelp'13 corpus:
[Panera bread wannabes.]$_1$ [Food was okay and coffee]$_2$ [was eh.]$_3$ [Not large portions for the price.]$_4$ [The free chocolate chip cookie was a nice touch]$_5$ [and the orange scone was good.]$_6$ [Broccoli cheddar soup was pretty good.]$_7$ [I would not come back.]$_8$

sion on the auxiliary task of sentiment classification to generate abundant data for RST-style discourse structure prediction.

We draw intuition from previous work using discourse parsing as an auxiliary task to enhance sentiment analysis (Bhatia et al., 2015; Nejat et al., 2017; Hogenboom et al., 2015). Our assumption is that such synergies between sentiment analysis and discourse parsing are bidirectional. In this paper, we leverage the synergy effects in the opposite direction by using sentiment analysis to create discourse structures.

Figure 1 illustrates the discourse structure of a strongly negative Yelp review generated by our system. While solely based on sentiment information, the structure nevertheless resembles a well aligned discourse tree. It can be observed that EDUs with negative sentiment are generally located at a higher level in the tree, while for example EDUs ⑤ and ⑥, with positive sentiment, are at the bottom of a deep subtree. This way, EDUs with negative sentiment strongly influence the overall sentiment, while EDUs ⑤ and ⑥ only have little impact. At the same time, semantically related EDUs generally have a shorter distance than semantically unrelated EDUs.

Our approach combines a neural variant of multiple-instance learning (MILNet) (Angelidis and Lapata, 2018), with an optimal CKY-style tree generation algorithm (Jurafsky and Martin, 2014). First, MILNet computes fine-grained sentiment values and un-normalized attention scores (Ji and Smith, 2017) on EDU-level, by solely relying on distant supervision signals from document-level annotations. These annotations are abundantly available from several published open source datasets such as Yelp'13 (Tang et al., 2015), IMDB (Diao et al., 2014) or Amazon (Zhang et al., 2015). Then, the sentiment values and attention scores are aggregated to guide the discourse-tree construction, optimized on the document gold-label sentiment, using optimal CKY-style parsing.

Following this approach, we generate a new corpus annotated with "silver standard" discourse trees, which comprises of 100k documents (two orders of magnitude more than any existing corpora). To test the quality of our new corpus, we run a series of experiments, where we train the top performing discourse parser by Wang et al. (2017) on our corpus for discourse structure prediction and compare it with the same parser trained on human annotated corpora in the news domain (RST-DT) and in the instructional domain. Results indicate that while training a parser on our corpus does not yet match the performance of a parser trained and tested on the same dataset (*intra-domain*), it does perform remarkably well on the significantly more difficult and arguably more useful task of *inter-domain* discourse structure prediction, where the parser is trained on one domain and tested/applied on another one. Our results on *inter-domain* discourse parsing, shown in Section 4, strongly suggest that if anyone wants to leverage discourse parsing in a domain without annotated data, it is advantageous to use a discourse parser which has been trained on our new corpus, rather than, for instance, on RST-DT.

2 Related Work

Our approach to address the lack of annotated data in discourse parsing lies at the intersection of RST-style parsing, sentiment analysis and multiple-instance learning (MIL).

A large number of highly diverse discourse parsers have been proposed in previous work, with non-neural ones achieving the best performance. In this paper, we consider a set of top-performing parsers, which follow fundamentally different intuitions on how the parsing process should be modelled. Joty et al. (2015) and Ji and Eisenstein (2014) argue that discourse parsing should use a single model for structure, nuclearity and relation modelling. Joty et al. (2015) further propose to separate the task "vertically" on sentence- and document-level, while Ji and Eisenstein (2014) are using a single Shift-Reduce parser based on lexical features. The current state-of-the-art system by Wang et al. (2017) follows an opposing intuition, namely that the task should be separated "horizontally" into two sequenced components. The first classifier models the structure and nuclearity, while the second classifier builds the relation model. Apart from being the state-of-the-art model, Wang et al. (2017) has the ideal architecture for our experiments. With the horizontal separation between structure/nuclearity and relation prediction classifiers, it can be easily tailored to just make discourse structure predictions when trained on our new corpus of discourse trees.

The second related area is sentiment analysis, which we use as our auxiliary task. Previous studies, e.g., Bhatia et al. (2015); Ji and Smith (2017), have shown that sentiment prediction can be enhanced by leveraging discourse information, as the tree structure can influence the significance of certain clauses in the document and boost the overall performance. In particular, Bhatia et al. (2015) use handcrafted discourse features on the sentiment classification task to score clauses depending on the level in the discourse tree. Ji and Smith (2017) use discourse trees generated by a discourse parser (Ji and Eisenstein, 2014) to inform a recursive neural network and automatically learn the model weights for sentiment prediction. In this paper, we exploit the relation between sentiment analysis and discourse parsing in the opposite direction by using sentiment annotations to create discourse structures.

The third area of related work is distant supervision aimed at automatically generating fine-grained annotations. Distant supervision has previously been used to retrieve sentiment (Marchetti-Bowick and Chambers, 2012; Tabassum et al., 2016) and emotion classes (Abdul-Mageed and Ungar, 2017) from opinionated text, showing the potential of distant supervision with user generated content. A common technique for distant supervision is multiple-instance learning (Keeler and Rumelhart, 1992), where the general idea is to retrieve fine-grained information from high-level signals. High-level signals are called *bags* and fine-grained information is referred to as *instances*. The task is defined as the generation of *instance* labels solely based on the given *bag* labels. We follow the approach by Angelidis and Lapata (2018), who train their MILNet system on the publicly available Yelp'13 (Tang et al., 2015) and IMDB (Diao et al., 2014) datasets. Results indicate that MILNet can capture EDU-level sentiment information as well as the relative importance of EDUs when evaluated on datasets annotated on EDU-level. In this paper, we adapt MILNet to generate information useful for deriving discourse trees from corpora with document-level sentiment annotations.

3 Our Approach

To generate a large number of discourse structures via distant supervision from sentiment, we propose a four-step approach, shown in Figures 2 and 3. Figure 2 illustrates the first stage of the approach, where for each document in the dataset, (a) the document is segmented into EDUs and (b) our adaptation of MILNet is trained on the document-level sentiment. Next, shown in Figure 3, we again (a) segment the document into EDUs and use (b) the MIL network to generate fine-grain sentiment and importance scores. Then in (c), we prepare those scores to be used in (d), the CKY-like parser, which generates an optimal RST discourse-tree for the document, based on the EDU-level scores and the gold label document sentiment.

3.1 Segmentation and Preprocessing

We initially separate the sentiment documents into a disjoint sequence of EDUs. The segmentation is obtained using the discourse segmenter by Feng and Hirst (2012) as generated and published by Angelidis and Lapata (2018). We preprocess the EDUs by removing infrequent- and stop-words and subsequently apply lemmatization.

3.2 Multiple-Instance Learning (MIL)

Our MIL model is closely related to the methodology described in Angelidis and Lapata (2018), as well as the papers by Yang et al. (2016) and Ji and Smith (2017). The computation is based on the initial segmentation described in section 3.1 and is shown in further detail in Figure 4.

Our model consists of two levels of Recurrent Neural Networks (RNN) inspired by Yang et al. (2016) and a sentiment- and attention-module. The computational flow in the model is defined

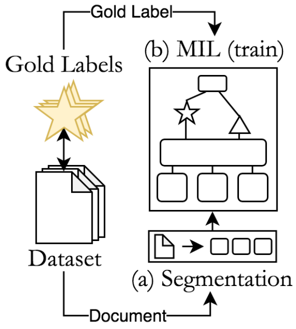

Figure 2: First stage, training the MIL model on the document-level sentiment prediction task

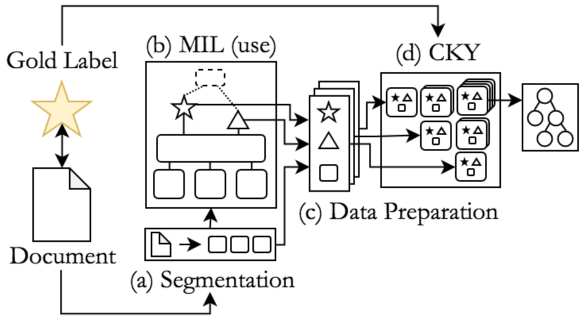

Figure 3: Second stage, using the neural MIL model to retrieve fine-grained sentiment and attention scores (star/triangle), used for the CKY computation to generate the optimal discourse tree

from bottom to top in Figure 4. In a first step, the sparse one-hot word representations are transformed into dense vector-representations w_i using a pretrained GloVe word embedding matrix (Pennington et al., 2014). The dense word representations of a single EDU $E_i = (w_j, ..., w_k)$ are used as the sequential input for the EDU-level RNN, implemented as a bi-directional GRU module with a standard attention mechanism (Bahdanau et al., 2014). The attention-weighted hidden-states $R_{w_i} = H_{w_i} * A_{w_i}$ are concatenated along the time axis to represent R_{E_i}. The second RNN on document-level subsequently uses the distributed EDU representations $(R_{E_1}, ..., R_{E_s})$ as inputs for the neural network. Based on the sequence of computed hidden-states $(H_{E_1}, ..., H_{E_s})$ in the bi-directional GRU network, two parallel model components are executed, as follows:

The non-competitive attention score module was proposed by Ji and Smith (2017) to leverage discourse structure for sentiment prediction. By following the same intuition, we replace the softmax activation on the attention weights by a sigmoid function. This way, each attention weight A_{E_i} is still limited within the range $(0, 1)$, but the sum of all attention scores is not necessarily bound by 1. We use the attention weight A_{E_i} as the importance scores of EDU$_\text{i}$.

The sentiment score module is also executed directly on the hidden-states $(H_{E_1}, ..., H_{E_s})$ generated by the document-level RNN. To be able to interpret the dense hidden representations as sentiment predictors, we use a single feed-forward neural network layer S with $|C|$ neurons, representing the disjoint sentiment classes $(C_1, ..., C_m)$ in the dataset[1]. We add a sigmoid activation $sigm$ after the feed-forward layer to obtain the final, internal EDU sentiment prediction $S_{E_i} = sigm(S(H_{E_i}))$.

The output of the two parallel modules is multiplied EDU-wise and summed up along the time axis to calculate the final sentiment prediction of our MILNet model as $O_D = \sum_{E_i \in D} S_{E_i} * A_{E_i}$ (see top of Figure 4).

To train our MILNet model, we use the cross-entropy loss function to compare O_D with the gold document-level sentiment label of the review and train using the Adadelta optimizer (Zeiler, 2012). By separating the sentiment and attention components and directly computing the final output based solely on these two values, the neural network implicitly learns the sentiment and attention scores on EDU-level as a by-product of the document-level prediction. For more information on this technique, we refer to Angelidis and Lapata (2018). The hyper-parameter setting of our model also mostly follows the implementation of previous work. We use a batch-size of 200 documents and train the model for 25 epochs. The bidirectional GRU layers contain 100 neurons and the model inputs are preprocessed by limiting the number of EDUs in a document to 150 and defining the maximum length of an EDU to be 20 words. With these settings, we capture over 90% of the data by significantly decreasing the training efforts. We apply 20% dropout on the internal sentiment layer.

[1] In the Yelp'13 dataset, the feed-forward operation S results in 5 real-valued outputs.

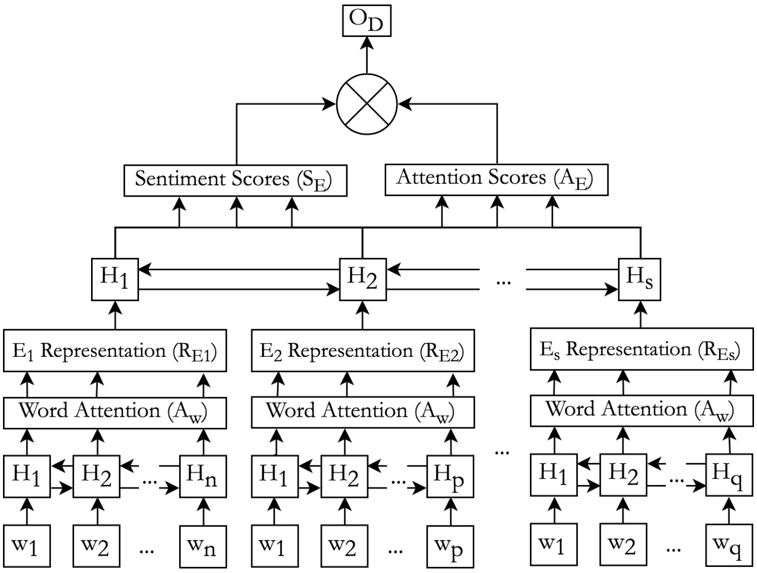

Figure 4: MIL Network Topology (For readability, we leave out the second subscript of the hidden representations H_{w_i} and H_{E_i})

3.3 Information Extraction and Transformation

Once our MILNet model is trained, we can use it to obtain the attention score A_{E_i} and the sentiment score S_{E_i} for each EDU E_i in a document (see (c) in Figure 3). However, while each A_{E_i} is already a scalar, S_{E_i} is a vector with $|C|$ elements, one for each sentiment class C_i. In order to effectively combine the attention and sentiment scores for further processing, we transform S_{E_i} into a scalar polarity score *pol*, centered around 0 and uniformly distribute the $|C|$ sentiment classes within the interval of $[-1, 1]$. For instance, if $|C| = 5$, this would result in the five classes $pol_{coeff} = [-1, -0.5, 0, 0.5, 1]$. The polarity pol_{E_i} of EDU E_i is computed by calculating the element-wise product of the sentiment score S_{E_i} and the uniform distribution pol_{coeff}.

$$pol_{E_i} = \sum_{c \in C} S_{E_i}^{(c)} * pol_{coeff}^{(c)} \qquad (1)$$

We transform the gold labels in the same way to keep the representations consistent. With the polarity scores pol_{E_i} replacing the original sentiment scores S_{E_i}, a neutral sentiment document now receives a sentiment polarity of 0, while heavily positive or negative EDUs are mapped onto the scores $+1$ and -1 respectively. This way, the obtained attention scores A_{E_i} and the calculated polarities pol_{E_i} can be combined to create a weighted sentiment score with high attention values resulting in stronger polarities.

3.4 CKY Tree Generation

The final step in our approach (see (d) in Figure 3) takes the tuples of EDU-level attention scores and the generated polarities from the MILNet model to create a set of possible discourse trees. We then select the discourse tree that most precisely computes the overall sentiment of the document. To find the globally best tree, we are computing all possible tree structures (with some constraints) using a dynamic programming approach closely related to the widely used CKY algorithm (Jurafsky and Martin, 2014).

To create discourse trees bottom-up using CKY, we define the necessary aggregation rules for local trees. For each binary subtree, we need to define a function $p(c_l, c_r)$ on how to aggregate the information of the two children c_l and c_r to represent the parent node p. For *sentiment*, we use the intuitive attention-weighted average of the children's sentiments, defined by:

$$p_s(c_l, c_r) = \frac{c_{l_s} * c_{l_a} + c_{r_s} * c_{r_a}}{c_{l_a} + c_{r_a}} \qquad (2)$$

This way, the parent sentiment does not only depend on the sentiment of its children, but also their relative importance.

For the *attention* computation we consider three different aggregation functions:

(1) The sum of the children's attentions (Eq. 3). This way, the combined importance of the children is inherited by the parent node, making the node as important as the combination of all sub-nodes.

$$p_{a_{sum}}(c_l, c_r) = (c_{l_a} + c_{r_a}) * (1 - \lambda) \quad (3)$$

where λ represents a damping factor to penalize lower sub-trees, empirically chosen to be 1% using grid-search.

(2) The maximum of the children's attentions.

$$p_{a_{max}}(c_l, c_r) = max(c_{l_a}, c_{r_a}) \quad (4)$$

As shown in equation 4, the attention of the parent node is calculated as the maximum attention value of the two children. This aggregation function follows the intuition that the parent node is only as relevant as the most important child node.

(3) The average of the children's attentions.

$$p_{a_{avg}}(c_l, c_r) = \frac{c_{l_a}, c_{r_a}}{2} \quad (5)$$

This aggregation function (Eq. 5) assigns the average importance of the two children to their parent.

To create RST-style discourse documents, which can be used by existing parsers, we need to also provide nuclearity and relation labels for every tree node. While we leave the general task of nuclearity- and relation-prediction for future work, we still need to assign those attributes to tree nodes. We assign nuclearity solely depending on the attention value of the children nodes, making the following binary decision:

$$c_{l_n} = \begin{cases} \text{``}Nucleus\text{''}, & \text{if } c_{l_a} \geq c_{r_a} \\ \text{``}Satellite\text{''}, & \text{otherwise} \end{cases} \quad (6)$$

This simple approach cannot assign "Nucleus-Nucleus" nuclearity attributes, but always requires one child to be the satellite. Finally, for the necessary rhetorical relation attribute, we simply assign the *span* relation to every node.

Due to the high complexity of the optimal CKY algorithm, to keep the process manageable by our computational resources[2], we introduce two constraints on the generated discourse trees:

- We prohibit inter-sentence relations, unless the complete sentence is represented by a single node (as shown to capture the vast majority of discourse relations by Joty et al. (2015))

- We only process documents with less or equal to 20 EDUs per document

With the aggregation functions and restrictions described above, we run the CKY-style dynamic programming approach and compare the sentiment at the root node of each of the complete discourse trees with the dataset gold-label for the document. The discourse tree with the smallest distance from the gold-label is selected as the discourse structure representation of the document (see Figure 3, on the right) and saved in a serializable RST-DT format. This way, we generate a dataset of 100k discourse trees.

4 Evaluation

We now describe the evaluation of our new discourse structure dataset. We start with the datasets and discourse parsers used to train and test our approach. Next, we describe the evaluation metrics, finishing with experiments and results.

Dataset	#Documents	#EDU/Doc	Vocab
Yelp'13(2015)	335,018	19.1	183,614
RST-DT(2002)	385	56.0	15,503
Instr-DT(2009)	176	32.6	3,453

Table 1: Dataset size

4.1 Datasets

We use three datasets to train and evaluate our approach. Table 1 summarizes the most important dataset dimensions.

Yelp'13 is a review dataset collected for the Yelp Dataset Challenge in 2013 (Tang et al., 2015). Every datapoint in the corpus consists of a review along with a star-rating on a 5-point scale. We use the discourse segmented version of the corpus by Angelidis and Lapata (2018) to train our system on the auxiliary sentiment prediction task.

RST-DT is the largest and most frequently used corpus to train RST-style discourse parsers (Carlson et al., 2002). The dataset consists of news articles from Wall Street Journal. We use the standard data split with 90% training data (*RST-DT$_{train}$*) and 10% test data (*RST-DT$_{test}$*) to test the performance of our approach against competitive baselines.

Instructional Dataset is another RST-style dataset to evaluate discourse parsers on the domain of home-repair instructions (Subba and Di Eugenio, 2009). For convenience, we refer to this corpus as Instr-DT from here on. We separate the data into 90% training data (*Instr-DT$_{train}$*) and 10% test data (*Instr-DT$_{test}$*).

[2] Intel Core i9-9820X, RTX 2080 Ti, 128 GB RAM

Vocabulary Overlap is measured using the Jaccard similarity index. We show the absolute vocabulary sizes of the datasets in Table 1 and visualize the overlap in Table 2. The vocabulary overlap between the Yelp'13 corpus (containing reviews), the RST-DT dataset (on news articles) and the Instr-DT corpus (containing home-repair instructions) is predictably low, given the different domains of the datasets. While this would be a problem for models solely basing their prediction on raw input words, our system goes beyond just words as inputs. During training, we use pretrained word embeddings to encode the inputs and the state-of-the-art discourse parser (Wang et al., 2017) uses a combination of syntactical and lexical features to represent words.

Yelp'13 ↔ RST-DT	6.28%
Yelp'13 ↔ Instr-DT	1.73%
RST-DT ↔ Instr-DT	11.65%

Table 2: Vocabulary overlap between datasets

4.2 Discourse Parsers

In our experiments, we apply four simple baselines and four competitive discourse parsers, often used in previous work for comparison studies.

Right/Left Branching Baselines: predict a binary, fully right- or left-branching tree for every document in the dataset.

Hierarchical Right/Left Branching Baselines: predict a binary, fully right- or left-branching tree on sentence-level and combine the sentence-level trees in right- or left-branching manner for every document in the dataset.

HILDA: a classic, greedy, bottom-up parser using linear SVMs (Hernault et al., 2010).

DPLP: a SVM-based shift-reduce parser build on linear projections of lexical features (Ji and Eisenstein, 2014).

CODRA: a CKY-based chart parser combined with Dynamic Conditional Random Fields, separating the computation on sentence- and document-level (Joty et al., 2015).

Two-stage Parser: current state-of-the-art parser by Wang et al. (2017). Employs two separate SVM classifiers for structure/nuclearity and relations, reaching the best performance for structure and nuclearity. This is the parser we rely on in our experiments due to its performance advantage compared to other discourse parsers and its separate computation of the structure/nuclearity and the discourse relation. We use the publicly available code provided by Wang et al. (2017) and remove the relation classification module.

4.3 Metrics

Consistent with previous work, e.g., Wang et al. (2017); Joty et al. (2015) and following the recent analysis by Morey et al. (2017), our key metric is the average micro precision on span level, computed as the global overlap of the discourse structure prediction and the gold structure. We traverse both discourse trees $tree_{pred_i}$ and $tree_{gold_i}$ of each document i in post-order and compute:

$$precision = \frac{\sum_i tree_{pred_i} \cap tree_{gold_i}}{\sum_i |tree_{gold_i}|} \quad (7)$$

Notice that the choice of precision over recall and F-score has no impact on the results when using manual segmentation, as shown in previous work, e.g., Wang et al. (2017); Joty et al. (2015).

4.4 Experiments and Results

We run experiments in two phases. In the first phase, the state-of-the-art discourse parser by Wang et al. (2017) is individually trained on each of the datasets and tested on the two corpora containing gold discourse annotations. In the second phase, the best results are placed in the broader context of competitive discourse parsers.

Phase 1: We train the state-of-the-art discourse parser on five different corpora and perform tests on two corpora. The five training corpora are: RST-DT$_{\text{train}}$, Instr-DT$_{\text{train}}$ and the three versions of our novel dataset, generated using the different attention aggregation functions (avg, max, sum) discussed in Section 3.4. The two corpora used for testing are RST-DT$_{\text{train}}$ and Instr-DT$_{\text{train}}$ (for which gold standard annotations are available). Notice that whenever training and testing are performed on the same corpus, the model is trained on the training portion of the dataset (RST-DT$_{\text{train}}$ or Instr-DT$_{\text{train}}$) and evaluated on the test data (RST-DT$_{\text{test}}$ or Instr-DT$_{\text{test}}$). Finally, since one of the key benefits of our approach is the ability to generate large dataset, we also assess the relation between the dataset size and the parser performance in this

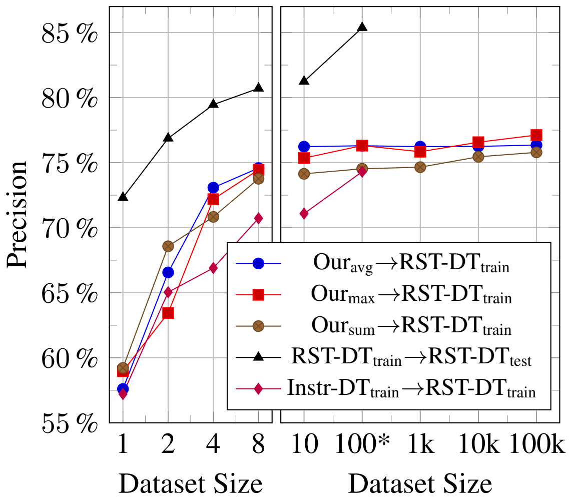

Figure 5: Results of training and testing on the datasets listed in the legend (*Complete dataset was used for RST-DT(385 documents) and Instr-DT(176 documents))

phase. The results of all our experiments in the first phase are shown in Figure 5 and Figure 6.

In both figures, the left side shows the performance using small subsets of sizes 2^n for $n = 0, 1, ..$ of the training data, while the right side shows the performance on large subsets of sizes 10^n, as well as the full datasets. The precision value displayed for each subset is the average of 10 randomly selected samples from the full corpus. Figure 7 shows the variance within the 10 samples, highlighting the increasing reliability of larger subsets[3].

The results shown in the two figures reveal several important findings:

(1) While training the parser on our corpus does not yet match the performance of the parser trained and tested on the same dataset (*intra-domain*, see black lines in Figure 5 and 6), it does achieve the best performance in the *inter-domain* discourse structure prediction.

(2) The performance generally increases with more training data. Larger datasets could therefore further increase the performance.

(3) Tested on the full *inter-domain* training datasets (100k), the *avg* attention-aggregation function achieves the most consistent performance on the corpora. When evaluated only on RST-DT, the *max* aggregation function reaches the overall best performance, while the *avg* attention-aggregation function reaches the best performance

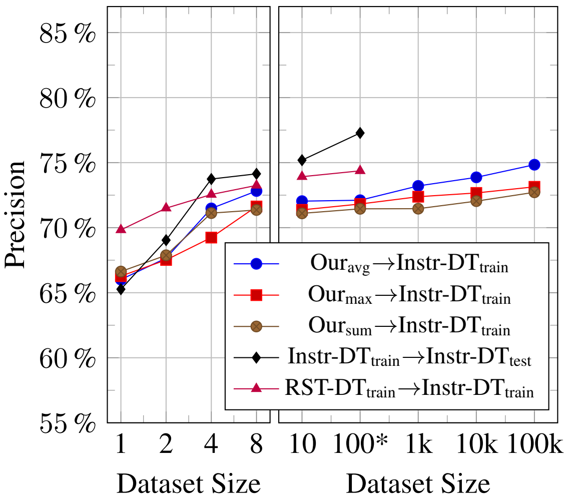

Figure 6: Results of training and testing on the datasets listed in the legend (*Complete dataset was used for RST-DT(385 documents) and Instr-DT(176 documents))

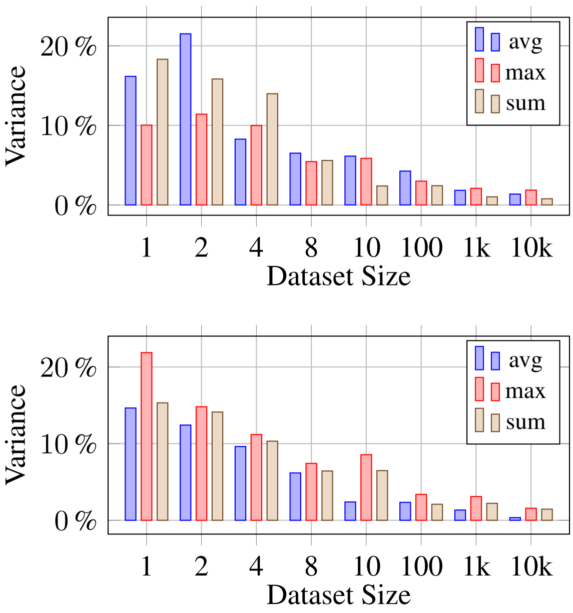

Figure 7: Sample variance across different subset sizes on the RST-DT dataset (top) and Instr-DT corpus (bottom)

when only evaluating on Instr-DT.

(4) Small subsets of training data (4, 8 documents) already achieve relatively good results, especially on Instr-DT. One possible explanation for this behaviour is the large number of training datapoints within a single document, which generates $2*n-1$ binary training instances, where n is the number of EDUs in the document.

(5) Generally, performance is highly dependent on the dataset domain and structure. In particu-

[3] The complete dataset has not been subsampled and therefore does not have a variance defined.

lar, the performance on the Instr-DT is generally lower and saturates earlier than on RST-DT. Both effects could result from Instr-DT containing less and shorter documents than RST-DT.

Phase 2: We further analyze our findings with respect to baselines and existing discourse parsers. The first set of results in Table 3 shows that the hierarchical right/left branching baselines dominate the completely right/left branching ones. However, their performance is still significantly worse than any discourse parser (*intra-* and *inter-domain*).

Approach	RST-DT$_{\text{test}}$	Instr-DT$_{\text{test}}$
Right Branching	54.64	58.47
Left Branching	53.73	48.15
Hier. Right Branch.	**70.82**	**67.86**
Hier. Left Branch.	70.58	63.49
Intra-Domain Evaluation		
HILDA(2010)	83.00	—
DPLP(2014)	82.08	—
CODRA(2015)	83.84	**82.88**
Two-Stage(2017)	**86.00**	77.28
Inter-Domain Evaluation		
Two-Stage$_{\text{RST-DT}}$	×	73.65
Two-Stage$_{\text{Instr-DT}}$	74.48	×
Two-Stage$_{\text{Ours(avg)}}$	76.42	**74.22**
Two-Stage$_{\text{Ours(max)}}$	**77.24**	73.12
Human (2017)	88.30	—

Table 3: Discourse structure prediction results; tested on RST-DT$_{\text{test}}$ and Instr-DT$_{\text{test}}$. Subscripts in *inter-domain* evaluation sub-table indicate the training set. Best performance in the category is **bold**. Consistently best model for *inter-domain* discourse structure prediction is underlined

The second set of results show the performance of existing discourse parsers when trained and tested on the same dataset (*intra-domain*). We use the results published in the original paper whenever possible. The Two-Stage approach by Wang et al. (2017) achieves the best performance with 86% on the structure prediction using the RST-DT dataset. On the Instructional dataset, the CODRA discourse parser by Joty et al. (2015) achieves the highest score with 82.88%.

The third set in the table shows the key results from Phase 1 on the *inter-domain* performance.

Our models, learning the discourse structure solely from the *inter-domain* Yelp'13 review dataset through distant supervision, reach better performance than the human annotated datasets (RST-DT and Instr-DT) when trained *inter-domain*, despite the low vocabulary overlap between the Yelp'13 corpus and RST-DT/Instr-DT (Table 2). While the avg attention-aggregation function achieves the most consistent performance on both evaluation corpora, the max function should not be dismissed, as it performs better on the larger RST-DT dataset, which is arguably more related to the Yelp'13 corpus than the sentiment-neutral Instr-DT. Furthermore, the fact that our best models reach a performance of only 8.76% and 8.66% below the best *intra-domain* performances (tested on RST-DT and Instr-DT, respectively), shows the potential of our approach, even when compared to *intra-domain* results.

5 Conclusions and Future Work

In this paper, we address a key limitation to further progress in discourse parsing: the lack of annotated datasets. We show promising initial results to overcome this limitation by creating a large-scale dataset using distant supervision on the auxiliary task of sentiment analysis. Experiments indicate that a parser trained on our new dataset outperforms parsers trained on human annotated datasets on the challenging and very useful task of *inter-domain* discourse structure prediction.

There are several directions for future work. First, given that we can now create large datasets, we intend to experiment on structure prediction with neural discourse parsers, which so far have delivered rather disappointing results. Second, an obvious next step is working on the integration of nuclearity and relation prediction to create complete RST annotations for documents from auxiliary tasks and to extend our evaluations (Zeldes, 2017). Third, we will study synergies between discourse parsing and further auxiliary tasks, eventually creating a single, joint system to generate globally high-quality discourse trees. Finally, instead of creating discourse structures and training existing discourse parsers on the data, we will design and implement an end-to-end system to train the complete process holistically.

Acknowledgments

We want to thank the anonymous reviewers for their insightful comments and suggestions. This research was supported by the Language & Speech Innovation Lab of Cloud BU, Huawei Technologies Co., Ltd.

References

Muhammad Abdul-Mageed and Lyle Ungar. 2017. Emonet: Fine-grained emotion detection with gated recurrent neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 718–728.

Stefanos Angelidis and Mirella Lapata. 2018. Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association for Computational Linguistics*, 6:17–31.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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

Chloé Braud, Maximin Coavoux, and Anders Søgaard. 2017. Cross-lingual rst discourse parsing. *arXiv preprint arXiv:1701.02946*.

Chloé Braud, Barbara Plank, and Anders Søgaard. 2016. Multi-view and multi-task training of rst discourse parsers. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1903–1913.

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

Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 193–202. ACM.

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 (Volume 1: Long Papers)*, pages 60–68.

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)*, volume 1, pages 511–521.

Elisa Ferracane, Greg Durrett, Junyi Jessy Li, and Katrin Erk. 2019. Evaluating discourse in structured text representations. *arXiv preprint arXiv:1906.01472*.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1602–1613.

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

Alexander Hogenboom, Flavius Frasincar, Franciska De Jong, and Uzay Kaymak. 2015. Using rhetorical structure in sentiment analysis. *Commun. ACM*, 58(7):69–77.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 13–24.

Yangfeng Ji and Noah A Smith. 2017. Neural discourse structure for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1005.

Shafiq Joty, Giuseppe Carenini, and Raymond T Ng. 2015. Codra: A novel discriminative framework for rhetorical analysis. *Computational Linguistics*, 41(3).

Dan Jurafsky and James H Martin. 2014. *Speech and language processing*, volume 3. Pearson London.

Jim Keeler and David E Rumelhart. 1992. A self-organizing integrated segmentation and recognition neural net. In *Advances in neural information processing systems*, pages 496–503.

Qi Li, Tianshi Li, and Baobao Chang. 2016. Discourse parsing with attention-based hierarchical neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 362–371.

Yang Liu and Mirella Lapata. 2017. Learning contextually informed representations for linear-time discourse parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1289–1298.

Yang Liu and Mirella Lapata. 2018. Learning structured text representations. *Transactions of the Association for Computational Linguistics*, 6:63–75.

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.

Micol Marchetti-Bowick and Nathanael Chambers. 2012. Learning for microblogs with distant supervision: Political forecasting with twitter. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 603–612.

Mathieu Morey, Philippe Muller, and Nicholas Asher. 2017. How much progress have we made on rst discourse parsing? a replication study of recent results on the rst-dt. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1319–1324.

Bita Nejat, Giuseppe Carenini, and Raymond Ng. 2017. Exploring joint neural model for sentence level discourse parsing and sentiment analysis. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 289–298.

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.

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

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.

Jeniya Tabassum, Alan Ritter, and Wei Xu. 2016. Tweetime: A minimally supervised method for recognizing and normalizing time expressions in twitter. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 307–318.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432.

Yizhong Wang, Sujian Li, and Houfeng Wang. 2017. A two-stage parsing method for text-level discourse analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 184–188.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Nan Yu, Meishan Zhang, and Guohong Fu. 2018. Transition-based neural rst parsing with implicit syntax features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 559–570.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.