Discourse-Aware Hierarchical Attention Network for Extractive Single-Document Summarization

Tatsuya Ishigaki[⋆] **Hidetaka Kamigaito**[⋆] **Hiroya Takamura**[⋆◇] **Manabu Okumura**[⋆]
[⋆]Tokyo Institute of Technology
[◇]National Institute of Advanced Industrial Science and Technology
{ishigaki, kamigaito}@lr.pi.titech.ac.jp, {takamura, oku}@pi.titech.ac.jp

Abstract

Discourse relations between sentences are often represented as a tree, and the tree structure provides important information for summarizers to create a short and coherent summary. However, current neural network-based summarizers treat the source document as just a sequence of sentences and ignore the tree-like discourse structure inherent in the document. To incorporate the information of a discourse tree structure into the neural network-based summarizers, we propose a discourse-aware neural extractive summarizer which can explicitly take into account the discourse dependency tree structure of the source document. Our discourse-aware summarizer can jointly learn the discourse structure and the salience score of a sentence by using novel hierarchical attention modules, which can be trained on automatically parsed discourse dependency trees. Experimental results showed that our model achieved competitive or better performances against state-of-the-art models in terms of ROUGE scores on the DailyMail dataset. We further conducted manual evaluations. The results showed that our approach also gained the coherence of the output summaries.

1 Introduction

Document summarization is the task of automatically shortening a source document while retaining its salient information. In this paper, we present a recurrent neural network (RNN)-based extractive summarizer taking into account the discourse structure inherent in the source document.

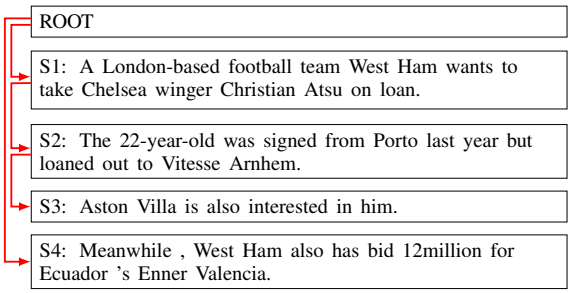

Figure 1: Example of discourse dependency structure.

The discourse structure consists of discourse relations between units in the input, and discourse information has been shown useful for summarization tasks. An example of a Rhetorical Structure Theory (RST) (Mann and Thompson, 1988)-based discourse structure, expressed as a dependency tree, is illustrated in Figure 1. In the figure, each node corresponds to a sentence. Regarding the relations between the sentences, sentence $S2$ elaborates the fact mentioned in sentence $S1$. In addition, $S2$ is further elaborated by $S3$. $S4$ is a contrast to the mention $S1$. Such relations are essential cues for generating a concise and coherent summary. For example, elaborated sentences tend to be more important than elaborating sentences, and the elaborated sentences should be included in the summary while the elaborating sentences are not.

Several Integer Linear Programming (ILP)-based summarizers (Hirao et al., 2013; Kikuchi et al., 2014) use the discourse information given by a discourse parser (Hernault et al., 2010). Thus, the performance of the summarizers is strongly affected by the performance of the discourse parsers. The performance of the parsers deteriorates especially when they are applied to documents of a domain different from the one which they were trained on.

Proceedings of Recent Advances in Natural Language Processing, pages 497–506,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_059

RNN-based approaches have achieved the state-of-the-art performance in document summarization (Cheng and Lapata, 2016; Nallapati et al., 2017). However, RNN-based summarizers treat the source document just as a sequence of sentences, and ignore the discourse tree structure inherent in the document. The lack of such information limits the ability to correctly compute relative importance between sentences and reduces the coherence of output summaries. Cohan et al. (2018) might be the only exception to the above, showing that the effectiveness of incorporating discourse information into an RNN-based summarizer for scientific papers by treating the source document as a sequence of sections such as "Introduction" or "Conclusion". However, they were not able to show how the tree-like discourse structure is effective in RNN-based approaches for extractive single-document summarization.

To effectively avoid the influence of parse errors and take advantage of the recent advances in neural network-based approaches, we propose a model that jointly learns the discourse tree structure of the source document and a scoring function for sentence extraction. Our model represents the discourse tree structure as an attention distribution and the probability of including a sentence in a summary as the softmax layer. In addition, recursive attention modules in our model can consider multi-hop dependencies between sentences. Therefore, our model can capture the relationships between sentences effectively and create a summary without losing the coherence between sentences.

We used an existing RST parser (Hernault et al., 2010) to add discourse dependency structure annotations to the DailyMail dataset (Hermann et al., 2015) and thereby obtained a large-scale annotated dataset to train the model. One of the advantages of our model is that we do not need the RST annotations in the inference phase because the model automatically infers the latent discourse tree structure of the source document and outputs the probability for each sentence as a salience score.

We empirically compared our model with other models. The results showed that discourse information improves the performance, and also that our models perform competitively with or better than state-of-the-art neural network-based extractive summarizers.

2 Related Work

There have long been many attempts at tackling extractive single-document summarization (Luhn, 1958), but there is still room for improvements in terms of ROUGE scores (Hirao et al., 2017). The recent focus has been on RNN-based approaches (Cheng and Lapata, 2016; Nallapati et al., 2017; Narayan et al., 2018). We further extend the attention mechanism used in RNN-based summarizers to capture a discourse structure.

RNN-based approaches were introduced to natural language processing tasks by the pioneering work by Bahdanau et al. (2015) and Luong et al. (2015), originally for machine translation. Rush et al. (2015) applied the approach to a sentence compression task. Nallapati et al. (2016) extended the model to abstractive document summarization. The DailyMail dataset (Hermann et al., 2015) has been commonly used for training abstractive summarizers. Cheng and Lapata (2016) and Nallapati et al. (2017) later proposed the methods to automatically annotate the binary labels, enabling us to train extractive models. Cohan et al. (2018) demonstrated the usefulness of incorporating discourse information into RNN-based summarizers. Unlike their model, our attention module explicitly captures the hierarchical tree structure inherent in the document.

Nallapati et al. (2016) and Yang et al. (2016) also used a hierarchical attention that consists of two simple attention modules; one is for words and the other is for sentences. Our attention mechanism differs from them in that ours captures discourse tree structures by new hierarchical attention networks, inspired by the models for capturing sentence-level dependency structures, e.g. machine translation (Hashimoto and Tsuruoka, 2017), dependency parsing (Zhang et al., 2017), constituency parsing (Kamigaito et al., 2017) and sentence compression (Kamigaito et al., 2018). Note that these models were designed for sentence-level tasks while we focus on the document-level summarization task.

Sentence selection modules that consider discourse structures of documents have been shown to be useful in ILP-based summarizers. Hirao et al. (2013) attempted to incorporate discourse information in ILP-based sentence extractors. Kikuchi et al. (2014) later proposed another ILP model that takes into account the discourse structure. Their model jointly selects and compresses sentences in

an ILP summarizer. Unlike the researches above, our focus is on incorporating discourse information into RNN-based summarizers.

Penn Discourse TreeBank (PDTB) (Prasad et al., 2008) and RST are the most commonly used framework to represent a discourse structure. PDTB focuses on the relation between two sentences, and the annotated structure for a document is not necessarily a tree. In contrast, RST is forced to represent a document as a tree. Discourse parsers for both schema are available (Hernault et al., 2010; Feng and Hirst, 2014; Wang and Lan, 2015). There are at least two methods to convert an RST-based tree structure to a dependency structure (Hirao et al., 2002; Li et al., 2014). Hayashi et al. (2016) compared these methods and mentioned that DEP-DT by Hirao et al. (2002) has an advantage for applying to summarization tasks. We use DEP-DT for this research since we focus on integrating the tree structure into a summarizer.

We found only one model that jointly learns RST parsing and document summarization (Goyal and Eisenstein, 2016). They used the SampleRank algorithm (Wick et al., 2011), a stochastic structure prediction model, while our main focus is to take into account discourse structures in RNN-based summarizers.

3 Problem Formulation

We formulate extractive document summarization as a sentence tagging problem. We first briefly explain the notation in the paper and describe the details of the model in the following sections.

The source document $\boldsymbol{x}$ is represented as a sequence of sentences $x_1, ..., x_N$. Each sentence x_i is composed of a sequence of words $w_{i,j}$ $(1 \leq j \leq M_i)$, where M_i is the number of words in x_i. We also consider x_0 as a dummy root node. The summarizer outputs a sequence of binary decisions $\boldsymbol{y} = y_1, ..., y_N$, where $y_i = 1$ for the i-th sentence x_i to be included in the summary and $y_i = 0$ for the sentence not to be included. The binary decisions $\boldsymbol{y}$ are made by using a neural network-based probability distribution function $p(y_i|\boldsymbol{x}, \theta)$, where θ is the set of learned parameters. The model finds the best decisions $\boldsymbol{y}$ by a simple greedy search to maximize the sum of the probabilities within the length constraint.

Thus, our goal is to construct a better function $p(y_i|\boldsymbol{x}, \theta)$ given training data D. Each instance in D is a triple $(\boldsymbol{x}, \boldsymbol{E}, \boldsymbol{y})$, where $\boldsymbol{E}$ is a matrix to represent the discourse dependency tree of $\boldsymbol{x}$. Specifically, element $E_{k,l}$ equals 1 if the edge from x_k to x_l exists in the discourse tree; otherwise $E_{k,l} = 0$.

Note that we use the discourse structure matrices $\boldsymbol{E}$ only in the training phase. The model does not require the RST annotations of the source document when calculating the probability distribution $p(y_i|\boldsymbol{x}, \theta)$.

4 RNN-Based Extractive Summarizer

In this section, we first explain the base model and give the details of our proposed attention module in the following section. The base model is composed of two main components: a neural network-based hierarchical document encoder and a decoder-based sentence scorer. The document encoder is further split into two components; a sentence reader and a document reader. The hierarchical architecture is commonly used in recent neural network-based models (Cheng and Lapata, 2016; Nallapati et al., 2017; Cohan et al., 2018).

4.1 Word Reader

The goal of the Word reader is to convert sentence x_i to a sentence embedding h_i. For each word $w_{i,j}$ in a sentence x_i, the word reader first convert every word embedding $emb(w_{i,j})$ to hidden states $\overrightarrow{e}_{i,j} = LSTM(\overrightarrow{e}_{i,j-1}, emb(w_{i,j}))$ and $\overleftarrow{e}_{i,j} = LSTM(\overleftarrow{e}_{i,j+1}, emb(w_{i,j}))$ by using bi-directional Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997). Then, $\overleftarrow{e}_{i,j}$ and $\overrightarrow{e}_{i,j}$ are concatenated into a hidden state $h_{i,j} = [\overrightarrow{e}_{i,j}; \overleftarrow{e}_{i,j}]$, where $[\,]$ represents a concatenation operation of a vector. After that, all $h_{i,j}$ in the sentence x_i are averaged and represented as a sentence embedding h_i.

4.2 Sentence Reader

Once we obtain sentence embeddings h_i for each sentence x_i, the Sentence reader then reads sentence embeddings h_i by another bi-directional LSTM and generates context-aware sentence representation H_i for each $\boldsymbol{x_i}$. Specifically, two vectors generated by the forward recurrent neural network $\overrightarrow{H}_i = LSTM(\overrightarrow{H}_{i-1}, h_i)$ and the backward $\overleftarrow{H}_i = LSTM(\overleftarrow{H}_{i+1}, h_i)$ are concatenated into sentence representation H_i for x_i:

$$H_i = [\overrightarrow{H}_i; \overleftarrow{H}_i]. \quad (1)$$

We now obtain the context-aware sentence representations $\boldsymbol{H} = \{H_1, ..., H_N\}$. Finally, all H_i are averaged to make a document embedding K.

4.3 Decoder-Based Sentence Scorer

This module outputs the probability of including x_i in the summary, $p(y_i = 1|\boldsymbol{x}, \theta)$, by using an LSTM-based decoder. At each time step t ($1 \leq t \leq N$), the previous state of the decoder s_{t-1} and the sentence representation h_t are fed into the LSTM, and the LSTM outputs a new state; $s_t = LSTM(s_{t-1}, h_t)$. The initial state s_0 is initialized by the last states of the backward LSTM in the document reader $\overleftarrow{H_0}$.

Extractive document summarization often adopts a "hard attention", which focuses only on the encoder hidden state H_t in the decoding time step t (Cheng and Lapata, 2016). In addition, document representation K is also important to decode summaries (Nallapati et al., 2017). Based on them, the output layer calculates the probability distribution of x_t being included in the summary as:

$$p(y_t|\mathbf{x}, \theta) = softmax(\mathbf{W}_o tanh(\mathbf{W}_c[H_t; s_t; K])). \quad (2)$$

5 Discourse-Aware Hierarchical Attention Network

We assume that taking into account the discourse dependency structure is also useful in determining whether the summary includes a target sentence or not. Here, we make the model capable of accounting for the information of the parent sentences on the discourse dependency structure by incorporating our proposed hierarchical attention mechanism into the RNN-based extractive summarizer.

As shown in Figure 2, the goal of our attention mechanism is to generate an attention vector Ω_i containing the information from the parent sentences of x_i through the three-step attention modules. Below, we first give an overview of each step in the procedure and then formulate the components after that.

Step1: Parent Attention Module This module calculates the probability of x_k being the parent of x_i for all combinations of k and i where $k \neq i$. We denote this probability as $p(k|i, \boldsymbol{H})$. In the figure, the starting point of an edge is the parent, and the end point is the child. The probability $p(k|i, \boldsymbol{H})$ is used as the weight for the edge from H_k to H_i. The edge weights are passed to the Recursive Attention Module.

Step2: Recursive Attention Module This module outputs the weighted sum vectors $\gamma_{d,i}$ over $\boldsymbol{H}$,

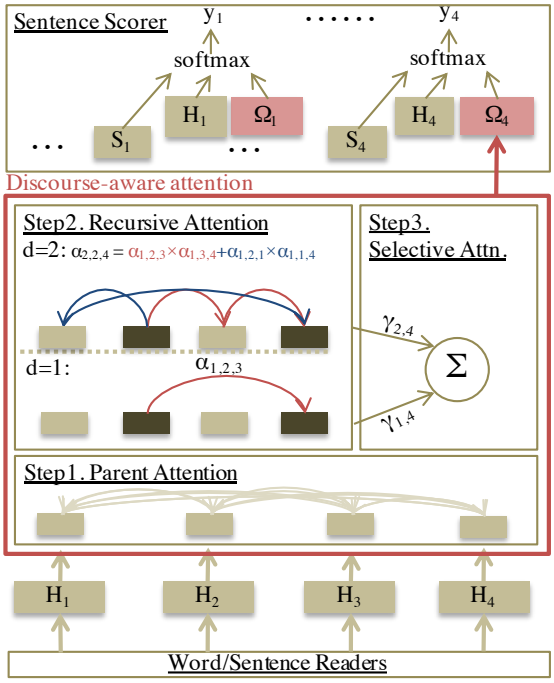

Figure 2: Overview of hierarchical attention mechanism for generating attention vector Ω_4. `Parent Attention` first calculates how likely the sentence x_i is the parent of x_j for all combinations. `Recursive Attention` then generates weighted sum vectors $\gamma_{d,4}$ over encoder hidden states H_i considering how likely the sentence x_i is the d-th order parent of x_4. `Selective Attention` finally generates another weighted sum vector Ω_4 over $\gamma_{d,4}$.

taking into account the d-th-order parents of x_i[1].

The module starts the calculation with the setting $d = 1$. Correspondingly, the module only considers the 1st-order parents of x_i. The Parent Attention Module has already calculated the probability of x_k being the parents of x_i. Thus, the Recursive Attention Module simply uses the probabilities as the weights $\alpha_{1,k,i}$ for every H_k and outputs the weighted sum vector $\gamma_{1,i}$.

When $d = 2$, the weights $\alpha_{2,k,i}$ for every H_k are calculated on the basis of how likely x_k becomes the 2nd-order parent of x_i. Here, d-th-order refers to the distance between x_k and x_i. For example, suppose there are two different paths connecting two nodes, and that their distances are both 2, illustrated by the path colored blue and red in Figure 2. The module multiplies the weights of the edges on each path, $\alpha_{1,2,3} \times \alpha_{1,3,4}$ for the red path and $\alpha_{1,2,1} \times \alpha_{1,1,4}$ for the blue path, and then the module sums the multiplied values;

[1]We use the plural form "parents" here because how likely a sentence becomes the parent of x_i is represented as a probability distribution in our model and multiple parents can be considered.

$\alpha_{2,2,4} = \alpha_{1,2,3} \times \alpha_{1,3,4} + \alpha_{1,2,1} \times \alpha_{1,1,4}$. We consider $\alpha_{2,2,4}$ to be the probability of x_2 being the 2nd-order parent of x_4. Then, the module uses the value as the weight and outputs the weighted sum vector $\gamma_{2,4}$.

When $d > 2$, the module recursively calculates the weight $\alpha_{d,k,i}$ by using the previously calculated weight $\alpha_{d-1,k,i}$ as shown in the next section.

Step3: Selective Attention Module Once weighted sum vectors $\gamma_{d,i}$ have been obtained taking into account the d-th-order parents of x_i, this module calculates the weights of each order d to select a suitable order. The module again calculates the weights for every order d and generates a weighted sum vector Ω_i.

5.1 Formulation of Attention Modules

Here, we describe the formulation of each attention module. The Parent Attention Module calculates the probability of x_k being the parents of x_i for all combinations of k and i where $k \neq i$:

$$\begin{aligned} p(k|i, \mathbf{H}) &= softmax(g(k, i)), \\ g(k, i) &= v_a^{\mathrm{T}} \tanh(U_a \cdot H_k + W_a H_i), \end{aligned} \tag{3}$$

where v_a, U_a and W_a are weight matrices.

The Recursive Attention Module recursively calculates the probability of x_k being the d-th-order parents of x_i:

$$\alpha_{d,k,i} = \begin{cases} p(k|i, \mathbf{H}) & (d = 1), \\ \sum_{l=0}^{N} \alpha_{d-1,k,l} \times \alpha_{1,l,i} & (d > 1). \end{cases} \tag{4}$$

Furthermore, in a discourse dependency tree, `ROOT` should not have any parent, and a sentence should not depend on itself. To satisfy these constraints, we impose the following on $\alpha_{1,k,i}$:

$$\alpha_{1,k,i} = \begin{cases} 1 & (k = 0, i = 0), \\ 0 & (k = i, i \neq 0). \end{cases} \tag{5}$$

The first equation constrains the `ROOT` node not to have any parent sentence. The second constraint ensures that a sentence does not depend on itself.

The calculated probabilities $\alpha_{d,k,i}$ are then used to weigh the vectors in $\mathbf{H}$, and the weighted sum vector $\gamma_{d,i}$ is generated as:

$$\gamma_{d,i} = \sum_{k=0}^{N} \alpha_{d,k,i} H_k. \tag{6}$$

Once the weighted sum vector $\gamma_{d,i}$ is obtained for each order d, the Selective Attention Module calculates the weights $\beta_{d,i}$ for each $\gamma_{d,i}$ to find a suitable order:

$$\beta_{d,i} = softmax(\mathbf{W}_\beta [H_i; s_i; K]), \tag{7}$$

where $\mathbf{W}_\beta$ is a weight matrix. The attention vector is obtained as a weighted sum of $\gamma_{d,t}$:

$$\Omega_i = \sum_d \beta_{d,i} \gamma_{d,i}. \tag{8}$$

Finally, the output layer receives the concatenated vector of H_i and Ω_i:

$$p(y_i|\mathbf{x}, \theta) = softmax(\mathbf{W}_o tanh(\mathbf{W}_{c'} [H_i; s_t; K; \Omega_i])). \tag{9}$$

5.2 Objective

The training updates the parameters to maximize both the label probability and 1st-order attention distribution $\alpha_{1,k,l}$. Specifically, we use the following loss function for optimization:

$$-\log p(\mathbf{y}|\mathbf{x}) - \lambda \cdot \sum_{k=1}^{N} \sum_{i=1}^{N} E_{k,i} \log \alpha_{1,k,i}. \tag{10}$$

In this equation, $E_{k,i}$ is 1 if the edge from x_k to x_i exists in the training instance. Thus, all the parameters are updated to reproduce the correct labels and edges appearing in the training data D. λ is a parameter to control the priority of the output labels or the edges given by an RST parser.

6 Experiments

Data and Preprocessing: We used two different datasets for the experiments; the DailyMail dataset for training and evaluation, and the DUC2002 test set[2] only for evaluation.

The DailyMail dataset (Hermann et al., 2015) consists of news articles extracted from Daily Mail Online[3] and their "story highlights" created by human writers. Nallapati et al. (2016) regarded the highlights as human-generated abstractive summaries. For training extractive summarization models, we need to annotate sentences with binary labels for sentence extraction. To do this, Cheng and Lapata (2016) used a rule-based approach considering the similarity between the original document and extracted sentences. On the other hand, Nallapati et al. (2017) proposed a simple heuristic for labeling sentences to be included in the summary by maximizing the ROUGE scores, using the highlights as reference summaries. We used the latter scheme to annotate the binary labels for sentence extraction.

As a preprocessing, we applied the HILDA parser (Hernault et al., 2010) to annotate RST-based discourse information for all the documents. The RST trees were then converted into dependency structures by using the method described in Hirao et al. (2013). The parser requires the features extracted from word surfaces and the information on paragraph boundaries. However,

[2] https://www-nlpir.nist.gov/projects/duc/guidelines/2002.html

[3] http://www.dailymail.co.uk/

the preprocessed DailyMail dataset[4] provided by Cheng and Lapata (2016) and commonly used for summarization tasks was not suitable for our use. The dataset is anonymized; all named entities were replaced by the special token `@entity`, and paragraph boundaries were deleted. Therefore, we used the non-anonymized version of the dataset provided by Hermann et al. (2015). We obtained 196,557 training documents, 12,147 validation documents and 10,396 test documents from the DailyMail dataset.

The DUC2002 test set consists of 116 pairs of source documents and their extractive summaries, and 567 pairs of source documents and their abstractive summaries. We used the dataset for evaluation on out-of-domain data.

Compared Models: We compared our models with various baseline models. `DIS w/ PAR` is our model with the model parameter $\lambda > 0$. With this setting, all the parameters are tuned to reproduce the correct labels and the edges given by the RST parser. `DIS fixed` is the discourse-aware model with the attention vector $\Omega_t = H_{t-1}$. Thus, this model always treats the preceding sentence as the parent. `DIS w/o PAR` is the model with the model parameter $\lambda = 0$. Note that the objective function in this model does not take into account the RST annotations given by the RST parser. Thus, all the discourse structures are learned to reproduce the correct sentence labels without the information from the parser.

We compared the above models with the model without any discourse-aware attention mechanisms (`no-attn`) to verify the effectiveness of our attention mechanisms. `Lead-3` is a common baseline to select the first three sentences. `SummaRuNNer` is a well-known RNN-based summarizer by Nallapati et al. (2017). This model uses some types of information that we do not use, such as the similarity between the source document and the target sentence, and the novelty score of the target sentence, while our approach incorporates the information on the parent sentence of the target sentence. `NeuralSum` is also a neural network-based summarizer which uses convolutional neural networks in the encoder. `Refresh` is a state-of-the-art method using reinforcement learning (Narayan et al., 2018) [5].

In addition to the above methods, we compared our models with previously reported performances on the DUC2002 test set. `LREG` is a feature-rich logistic regression based approach used as a baseline in Cheng and Lapata (2016). `ILP` is a phrase-based extraction system proposed by Woodsend and Lapata (2010). The approach extracts the phrases and recombines them subject to the constraints in the ILP such as length, coverage or grammaticality. Both `TGRAPH` (Parveen et al., 2015) and `URANK` (Wan, 2010) are graph-based sentence extraction approaches, that perform well on the DUC2002 corpus.

Evaluation Metrics: We conducted both automatic evaluation and human evaluation. In automatic evaluation, we adopted ROUGE scores (Lin, 2004). We specifically calculated ROUGE-1, ROUGE-2 and ROUGE-L by using the Pyrouge library[6]. The highlights in the Dailymail dataset were treated as reference summaries when we calculated the scores. We used three length constraints; 75 bytes, 275 bytes (Nallapati et al., 2017; Cheng and Lapata, 2016) and the bytes of reference summaries. We truncated generated summaries in the middle to conform to the length constraints. We adopted the last constraint to evaluate whether a model can include sufficient information within the ideal summary length. For the evaluation on out-of-domain data, we report the ROUGE scores on the DUC2002 abstractive and extractive test sets. Our models are trained on the DailyMail dataset and tested on DUC2002.

We additionally carried out human evaluation because ROUGE scores cannot capture the coherence, though our attention modules are designed to improve the coherence of summaries. We used Amazon Mechanical Turk to conduct human evaluation. Specifically, randomly selected 100 documents and their four summaries generated by `DIS w/ PAR`, `Lead-3`, `no-attn`, and `SummaRuNNer` were shown to the workers. Five workers were asked to rate each summary on a 1-5 scale in terms of coherence and informativeness. The instruction shown to the workers follows the DUC quality question[7].

Training Details: We used Adam (Kingma and Ba, 2015) for the optimizer, where the learning rate was set to 0.001. In accordance with the model parameters used in Nallapati et al. (2017),

[4] http://homepages.inf.ed.ac.uk/mlap/index.php?page=resources

[5] We used the implementation provided by the authors.

[6] The options for the Rouge script were "-a -c 95 -m -n 2 -b 75" and "-a -c 95 -m -n 2 -b 275".

[7] https://duc.nist.gov/duc2007/quality-questions.txt

we limited the vocabulary size of the input to 150,000 and replaced the out-of-vocabulary words with the token UNK. The size of the mini-batch was set to 8. We used the size of 100 for hidden layers in the LSTMs and 300 for word embeddings, which were initialized with pre-trained embeddings, word2vec-slim. Note that the previous researches (Nallapati et al., 2017; Cheng and Lapata, 2016) also used pre-trained embeddings. We filtered the training instances consisting of 50 or more sentences in the source document, following Nallapati et al. (2017). The parameter λ of all the discourse-aware models was tuned on the validation set. We tried the following values for λ: 0.01, 0.1, 1.0, 2.0, 5.0 and 10.0.

7 Results and Discussion

ROUGE scores on DailyMail dataset: Table 1 shows the ROUGE scores evaluated on the DailyMail test set. For fair comparison, we also re-trained the baseline models by using our non-anonymized dataset.

DIS w/o PAR achieved better ROUGE scores than no-attn in all variations of length constraint except for ROUGE-L score on the setting with $d = \{1, 2\}$. Furthermore, we obtained better scores for DIS fixed than DIS w/o PAR. Incorporating the simple discourse information which treats the preceding sentence as the parent in the objective function improved the performance. Exploiting the discourse information given by the RST parser (DIS w/ PAR) further improved the scores in most settings. These observations suggest that discourse information is useful in RNN-based summarizers.

In the setting with the length constraint of 75 bytes, we observed a statistically significant difference between DIS w/ PAR and other neural network-based models (SummaRuNNer , Refresh and NeuralSum) on the settings with $d = \{1, 2\}$ and $d = \{1, 2, 3\}$. We also observed the similar tendency in the setting with the length constraint of reference summaries. Furthermore, we did not observe a statistically significant difference between DIS w/ PAR and SummaRuNNer in the setting with the length constraint of 275 bytes. Those facts would suggest that our models achieve a performance similar to the other baseline models in the setting with longer length constraints, and can perform better with shorter length constraints.

ROUGE scores on DUC2002 dataset: The results are shown in Table 2. Neural network-based approaches achieved similar scores on the abstractive test set because the length constraint is long; specifically it was set to 100-words. However, graph-based approaches (TGRAPH and URANK) performed better than the neural network-based approaches. As reported in Nallapati et al. (2017), neural network-based approaches suffer the difficulties in achieving high performance on out-of-domain data due to its high capability to fit in-domain data. Another possible reason might be the method for creating the binary labels for the training dataset. The binary decisions on the training dataset were made to maximize the ROUGE-F scores. Thus, the labels are strongly affected by the length of reference summaries in the DailyMail dataset. Since the average length of the reference summaries in the DUC test set is longer than the average length in the DailyMail dataset, the models trained on the DailyMail dataset might face difficulties. Our proposed models achieved the significantly better performances among the neural network-based approaches on the extractive test sets, which are for the settings with shorter length constraints (50 and 100 words) .

Human Evaluation: Table 3 shows the results. DIS w/ PAR were evaluated better than no-attn and SummaRuNNer in terms of coherence in the settings with all the different length constraints. These differences are statistically significant with the sign test ($p < 0.05$). Thus, human evaluation also supports the effectiveness of incorporating discourse information. Lead-3 is inherently strong in terms of coherence because this model is constrained to extract consecutive sentences while other models possibly extract non-consecutive ones. It was evaluated better in the setting with 75 bytes length constraint.

Analysis: Table 4 shows an example of the source document and outputs of two models; the sentences selected by our model are colored red and those selected by SummaRuNNer are blue, and those selected by both are purple. In this example, $S7$ elaborates $S6$. Our summarizer successfully extracted $S6$, that made the output summary more similar to the gold summary.

8 Conclusion

We presented a hierarchical attention network that captures the discourse dependency structure of the

	75			275			Ref.		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
DIS w/ PAR, d={1}	22.9	9.6	12.1	41.9	17.4	**35.2**	+41.1	+**16.9**	+36.7
DIS w/ PAR, d={1,2}	+**25.0**	+11.1	+13.4	41.7	17.4	35.0	41.0	16.7	+36.7
DIS w/ PAR, d={1,2,3}#	+24.6	+**11.2**	+**13.5**	41.8	17.2	**35.2**	+41.1	+16.8	+36.8
DIS w/ PAR, d={1,2,3,4}	+24.1	10.8	+13.2	41.1	**17.5**	35.1	+**41.2**	+**16.9**	+**37.0**
DIS fixed, d={1}	23.4	10.3	12.7	39.9	16.1	33.5	39.4	15.7	35.4
DIS fixed, d={1,2}	23.5	10.4	12.8	40.3	15.9	33.6	39.7	16.1	35.8
DIS fixed, d={1,2,3}	22.9	9.8	12.3	40.3	16.4	33.8	39.6	15.9	35.6
DIS fixed, d={1,2,3,4}	22.6	9.2	11.7	39.8	15.6	33.4	39.3	16.1	35.9
DIS w/o PAR, d={1}	21.2	8.1	11.0	40.1	15.8	33.7	39.6	15.5	35.5
DIS w/o PAR, d={1,2}	21.1	7.5	10.6	40.0	15.8	33.0	39.6	15.6	35.5
DIS w/o PAR, d={1,2,3}	20.9	7.9	10.9	40.5	16.1	34.1	40.0	15.8	35.8
DIS w/o PAR, d={1,2,3,4}	21.1	8.0	10.9	40.2	15.7	33.6	39.6	15.5	35.6
Lead-3	23.0	9.4	11.8	41.9	17.0	32.5	40.4	16.3	36.1
no-attn	20.1	7.1	10.4	39.6	15.4	33.3	39.3	15.3	35.2
SummaRuNNer (re-run)	23.2	9.6	11.0	**42.0**	17.2	32.5	37.6	14.8	33.7
Refresh (re-run)	23.1	10.9	12.6	37.9	16.5	31.4	36.6	15.8	34.1
NeuralSum (re-run)	22.4	9.1	11.8	40.8	16.3	34.8	40.3	15.9	36.1

Table 1: ROUGE Scores on DailyMail dataset. The models are **trained and tested on DailyMail dataset**. The length constraints are set to 75 bytes, 275 bytes and the reference length. The best scores among the models in bold. The symbol + indicates statistical significance using 95% confidence interval with respect to the nearest baseline, estimated by the ROUGE script. # indicates the model that achieved the best score in ROUGE-2 among the same methods with different **d** in the development dataset.

	Abstracts			Extracts (50 words)			Extracts (10 words)		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
DIS w/ PAR	44.7	21.8	38.2	**43.7**	**14.2**	**38.6**	**21.4**	**8.2**	**19.5**
DIS w/o PAR	46.1	23.3	**43.6**	42.1	13.5	36.8	19.0	7.5	17.3
no-attn	43.3	20.9	41.0	42.5	13.3	37.3	19.1	7.1	17.3
SummaRuNNer	46.6	23.1	43.0	42.1	13.4	36.7	20.9	7.8	19.0
Refresh	-	-	-	-	-	-	-	-	-
NeuralSum	-	-	-	-	-	-	-	-	-
LEAD-3	43.6	21.0	40.2	43.4	14.1	38.4	21.3	8.1	19.4
LREG	43.8	20.7	40.3	-	-	-	-	-	-
ILP	45.4	21.3	42.8	-	-	-	-	-	-
TGRAPH	48.1	**24.3**	-	-	-	-	-	-	-
URANK	**48.5**	21.5	-	-	-	-	-	-	-

Table 2: ROUGE scores on DUC2002 dataset. All neural network-based models are trained on DailyMail dataset and tested on DUC2002 test set.

	75		275		Ref	
	C	I	C	I	C	I
DIS w/ PAR	*3.86	3.57	***4.11**	***3.97**	***3.98**	3.78
SummaRuNNer	3.61	3.57	2.98	3.77	2.81	3.16
no-attn	3.73	3.52	3.92	3.86	3.80	3.70
LEAD-3	**3.98**	**3.69**	4.06	**3.97**	3.94	**3.80**

Table 3: Human evaluation on randomly selected 100 documents from DailyMail dataset. C and I stand for coherence and informativeness respectively. The mark * indicates that DIS w/ PAR achieved statistically significant difference, calculated by the sign test ($p < 0.05$), from both SummaRuNNer and no-attn.

Document:
S1: Bayern Munich is interested in Chelsea defender Branislav Ivanovic but are unlikely to make a move until Jan.
S2: The Serbia captain has yet to open talks over a new contract at Chelsea and his current deal runs out in 2016.
S3: Chelsea defender Branislav Ivanovic could be targeted by Bayern Munich in the January transfer window.
S4: Bayern like Ivanovic but don't expect Chelsea to sell yet they know he will be free to talk to foreign clubs from Jan.
S5: Paris Saint-germain will make a 7million offer for Chelsea goalkeeper Petr Cech this summer.
S6: The 32-year-old is poised to leave Stamford Bridge and wants to play for a champions league.
S7: Contender PSG are set to make a 7million bid for Ivanovic's Chelse a team-mate Petr Cech in the summer.
Gold Summary:
Branislav Ivanovic's contract at Chelsea expires at the end of next season.The 31-year-old has yet to open talks over a new deal at Stanford bridge. Petrcech is poised to leave Chelsea at the end of the season

Table 4: Example of the extracted sentences. The sentences in bold are included in our summary. The sentences colored red were selected by our model, blue were by SummaRuNNer and purple were by both models.

source document. The experiments showed that incorporating discourse information into RNN-based extractive summarizers improves coherence and informativeness evaluated by human judges in addition to ROUGE scores. Our models outperformed or achieved competitive performances against the state-of-the-art methods. Improving the performance on out-of-domain data will be one of our future directions.

Acknowledgements

We thank to Tsutomu Hirao for helpful comments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR2015*.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of ACL2016*, pages 484–494, Berlin, Germany.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of NAACL2018*, pages 615–621.

Vanessa Wei Feng and Graeme Hirst. 2014. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of ACL2014*, pages 511–521.

Naman Goyal and Jacob Eisenstein. 2016. A joint model of rhetorical discourse structure and summarization. In *Proceedings of the Workshop on Structured Prediction for NLP*, pages 25–34.

Kazuma Hashimoto and Yoshimasa Tsuruoka. 2017. Neural machine translation with source-side latent graph parsing. In *Proceedings of EMNLP2017*, pages 125–135.

Katsuhiko Hayashi, Tsutomu Hirao, and Masaaki Nagata. 2016. Empirical comparison of dependency conversions for rst discourse trees. In *Proceedings of SIGDIAL2016*, pages 128–136.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of NIPS2015*, pages 1693–1701.

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

Tsutomu Hirao, Hideki Isozaki, Eisaku Maeda, and Yuji Matsumoto. 2002. Extracting important sentences with support vector machines. In *Proceedings of COLING2002*, pages 1–7.

Tsutomu Hirao, Masaaki Nishino, and Masaaki Nagata. 2017. Oracle summaries of compressive summarization. In *Proceedings of ACL2017*, pages 275–280.

Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. 2013. Single-document summarization as a tree knapsack problem. In *Proceedings of EMNLP2013*, pages 1515–1520.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hidetaka Kamigaito, Katsuhiko Hayashi, Tsutomu Hirao, and Masaaki Nagata. 2018. Higher-order syntactic attention network for longer sentence compression. In *Proceedings of NAACL2018*, pages 1716–1726.

Hidetaka Kamigaito, Katsuhiko Hayashi, Tsutomu Hirao, Hiroya Takamura, Manabu Okumura, and Masaaki Nagata. 2017. Supervised attention for sequence-to-sequence constituency parsing. In *Proceedings of IJCNLP2018 (Volume 2)*, pages 7–12.

Yuta Kikuchi, Tsutomu Hirao, Hiroya Takamura, Manabu Okumura, and Masaaki Nagata. 2014. Single document summarization based on nested tree structure. In *Proceedings of ACL2014*, pages 315–320.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR2015*.

Sujian Li, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014. Text-level discourse dependency parsing. In *Proceedings of ACL2014*, pages 25–35.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of ACL2004 Workshop*, pages 74–81.

H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, (2):159–165.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP2015*, pages 1412–1421.

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.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of AAAI2017*, pages 3075–3081.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Ça glar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of CoNLL2016*, pages 280–290.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of NAACL2018*, pages 1747–1759.

Daraksha Parveen, Hans-Martin Ramsl, and Michael Strube. 2015. Topical coherence for graph-based extractive summarization. In *Proceedings of EMNLP2015*, pages 1949–1954.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The penn discourse treebank 2.0. In *Proceedings of LREC2008*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for sentence summarization. In *Proceedings of EMNLP2015*, pages 379–389.

Xiaojun Wan. 2010. Towards a unified approach to simultaneous single-document and multi-document summarizations. In *Proceedings of COLING2010*, pages 1137–1145.

Jianxiang Wang and Man Lan. 2015. A refined end-to-end discourse parser. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning-Shared Task*, pages 17–24.

Michael L Wick, Khashayar Rohanimanesh, Kedar Bellare, Aron Culotta, and Andrew McCallum. 2011. Samplerank: Training factor graphs with atomic gradients. In *Proceedings of ICML2011*, pages 777–784.

Kristian Woodsend and Mirella Lapata. 2010. Automatic generation of story highlights. In *Proceedings of ACL2010*, pages 565–574.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL2016*, pages 1480–1489.

Xingxing Zhang, Jianpeng Cheng, and Mirella Lapata. 2017. Dependency parsing as head selection. In *Proceedings of EACL2017*, volume 1, pages 665–676.