Discourse-Aware Neural Extractive Model for Text Summarization

Jiacheng Xu[1*]**, Zhe Gan**[2]**, Yu Cheng**[2]**, Jingjing Liu**[2]

[1]University of Texas at Austin   [2]Microsoft Dynamics 365 AI Research
jcxu@cs.utexas.edu; {zhe.gan, yu.cheng, jingjl}@microsoft.com

Abstract

Recently BERT has been adopted in state-of-the-art text summarization models for document encoding. However, such BERT-based extractive models use the sentence as the minimal selection unit, which often results in redundant or uninformative phrases in the generated summaries. As BERT is pre-trained on sentence pairs, not documents, the long-range dependencies between sentences are not well captured. To address these issues, we present a graph-based discourse-aware neural summarization model - DISCOBERT. By utilizing discourse segmentation to extract discourse units (instead of sentences) as candidates, DISCOBERT provides a fine-grained granularity for extractive selection, which helps reduce redundancy in extracted summaries. Based on this, two discourse graphs are further proposed: (i) RST Graph based on RST discourse trees; and (ii) Coreference Graph based on coreference mentions in the document. DISCOBERT first encodes the extracted discourse units with BERT, and then uses a graph convolutional network to capture the long-range dependencies among discourse units through the constructed graphs. Experimental results on two popular summarization datasets demonstrate that DISCOBERT outperforms state-of-the-art methods by a significant margin.

Introduction

Neural networks have achieved great success in the task of text summarization (Yao, Wan, and Xiao 2017). There are two main lines of research: abstractive and extractive. While the abstractive paradigm (Rush, Chopra, and Weston 2015; See, Liu, and Manning 2017; Celikyilmaz et al. 2018; Sharma et al. 2019) focuses on generating a summary word-by-word after encoding the full document, the extractive approach (Cheng and Lapata 2016; Cao et al. 2016; Zhou et al. 2018; Narayan, Cohen, and Lapata 2018) directly selects sentences from the document to assemble into a summary. The abstractive approach is more flexible and generally produces less redundant summaries, while the extractive approach enjoys better factuality and efficiency (Cao et al. 2018).

Recently, some hybrid methods have been proposed to take advantage of both, by designing a two-stage pipeline to first select and then rewrite (or compress) candidate sentences (Chen and Bansal 2018; Gehrmann, Deng, and Rush
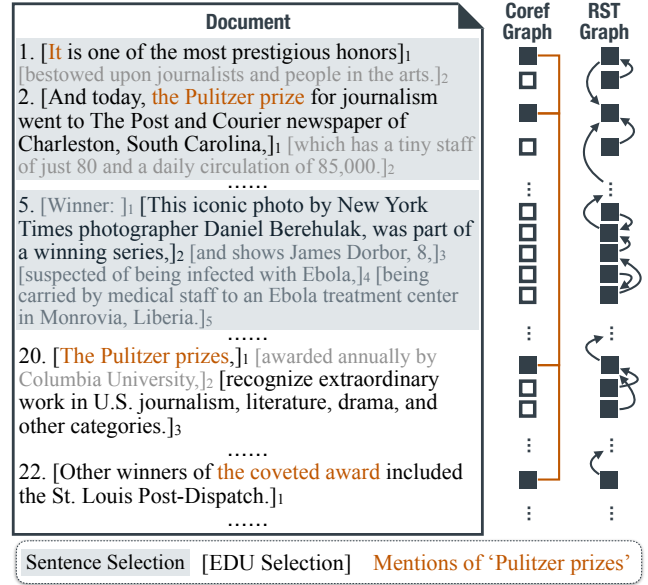
Figure 1: Illustration of DISCOBERT for text summarization. Sentence-based BERT model (baseline) selects whole sentences 1, 2 and 5. The proposed discourse-aware model DISCOBERT selects EDUs {1-1, 2-1, 5-2, 20-1, 20-3, 22-1}, which avoids unnecessary details and generates a more concise summary. The right side of the figure shows the two discourse graphs we use: (i) Coreference Graph (with the mentions of 'Pulitzer prizes' annotated as example); and (ii) RST Graph (induced by RST discourse trees).

2018; Zhang et al. 2018; Xu and Durrett 2019). Compression or rewriting aims to discard uninformative phrases from the selected sentences, but most of such systems suffer from the disconnection of the two stages in the pipeline, which results in limited performance improvement.

Meanwhile, modeling long-range context for document summarization still remains a challenging task. With the recent success of pre-trained language models (LMs) (Devlin et al. 2019), the encoding of input document has been greatly improved. However, since pre-trained LMs are mostly designed for target sentence pairs or short paragraphs, they perform poorly at capturing long-range dependencies among sentences. Empirical observations (Liu 2019) show that adding standard encoders such as LSTM or Transformer

(Vaswani et al. 2017) on top of BERT to model inter-sentential relations does not bring in much performance gain.

In this work, we present DISCOBERT, a discourse-aware neural extractive summarization model built upon BERT. To perform compression simultaneously with extraction and reduce the redundancy across sentences, we take the Elementary Discourse Unit (EDU), a sub-sentence phrase unit originated from RST (Mann and Thompson 1988; Carlson, Marcu, and Okurovsky 2001) as the minimal selection unit (instead of the sentence unit) for extractive summarization. Figure 1 shows an example of discourse segmentation, with sentences broken down into EDUs (annotated with brackets). The Baseline Selection is realized by sentence-based BERT model, and the EDU Selection is achieved by our model. After discarding some redundant details in the sentences, our model has the capacity for including additional concepts or events, therefore generating more concise and informative summaries.

Furthermore, to better capture document-level long-distance dependency, we also propose a graph-based approach to leverage intra-sentence discourse relations among EDUs. Two types of discourse graph are proposed: (*i*) a directed RST Graph, and (*ii*) an undirected Coreference Graph. The RST Graph is constructed from the parse tree over the EDUs of the document. Rhetorical relations of EDUs, such as contradiction, elaboration, and attribution, are addressed in the RST Graph. On the other hand, the Coreference Graph connects the entities and their coreference clusters/mentions across the document. In Figure 1, we show part of the coreference mention cluster of 'Pulitzer prize'. The path of coreference navigates the model from the core event to other occurrences of the event, as well as further exploring its interactions with other concepts or events. After constructing the graphs, Graph Convolutional Network (GCN) (Kipf and Welling 2017) is employed to capture the long-range interactions among EDUs.

The main contributions of this paper are summarized as follows. (*i*) We propose a discourse-aware extractive summarization model, DISCOBERT, treating EDUs (instead of sentences) as the minimal selection unit to provide a fine-grained granularity for extractive selection, while preserving the grammaticality and fluency of generated summaries. (*ii*) We propose two discourse graphs, and use a graph-based approach to model the inter-sentential context based on discourse relations among EDUs. (*iii*) Experimental results show that DISCOBERT achieves new state-of-the-art performance on two popular newswire text summarization datasets.

Discourse Graph Construction

We first introduce the concept of Rhetorical Structure Theory (RST) (Mann and Thompson 1988), a linguistic theory for discourse analysis, and then explain the methods to construct discourse graphs, which will be used in DISCOBERT. Two discourse graphs are considered: RST Graph and Coreference Graph. For initialization of both graphs, all edges are disconnected. Connections are then added for a subset
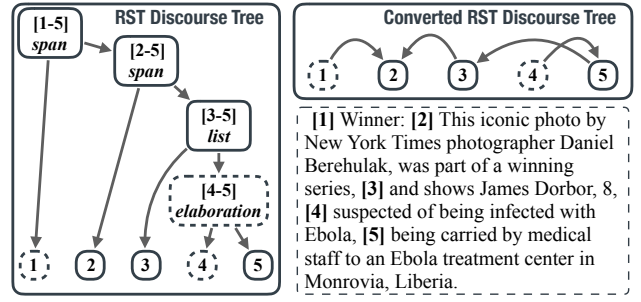

Figure 2: Example of discourse segmentation and RST tree conversion. The original sentence is segmented into 5 EDUs in the lower right box, and then parsed into an RST discourse tree in the left box. The converted dependency-based RST discourse tree is shown in the upper right box. Nucleus and Satellite nodes are denoted in solid lines and dashed lines, respectively. Non-terminal relation nodes with the start and the end of the span are shown on the right side. *Relations* are in italic. The second EDU [2] is the head of the whole tree (span [1-5]), while the third EDU [3] is the head of the span [3-5].

of nodes based on RST discourse parse tree or coreference mentions.

Discourse Analysis

Discourse analysis focuses on inter-sentential relations of text in a document or conversation. In RST, the discourse structure of text can be represented in a tree format. The whole document can be segmented into contiguous, adjacent and non-overlapping text spans called Elementary Discourse Units (EDUs). Each EDU is tagged as either Nucleus or Satellite, which characterizes its nuclearity or saliency. Nucleus nodes are generally more central, and Satellite nodes are more peripheral and less important in terms of its content and grammatical reliance. There are dependencies among the EDUs that represent their rhetorical relations. In this work, we treat EDU as the minimal unit for content selection in text summarization. Figure 2 shows an example of discourse segmentation and the parse tree of a sentence. Among these EDUs, rhetorical relations represent the functions of different discourse units. For example, 'elaboration' shows that one EDU describes the detail of the other EDU. 'Contrast' means that two EDUs are rhetorically different or opposed to each other.

In text summarization, we expect a good model to select the most concise, relevant and central concept of the document with low redundancy. For example, in Figure 2, details such as the name of the suspected child in [3], the exact location of the photo in [5], and who was carrying the child in [4], are unlikely to be reflected in the final summary. However, in traditional extractive summarization methods, the model is required to select the *whole* sentence, even though some parts of the sentence are not necessary. In our approach, the model is allowed to select one or several fine-grained EDUs to make the generated summaries less redundant. This serves as the basis of our proposed DISCOBERT model.

Algorithm 1 Construction of the Coreference Graph $\mathcal{G}_C$.

Require: Coreference clusters $C = \{C_1, C_2, \cdots, C_n\}$; mentions for each cluster $C_i = \{E_{i1}, \cdots, E_{im}\}$.
    Initialize the Graph $\mathcal{G}_C$ without any edge $\mathcal{G}_C[*][*] = 0$.
    for $i = 0$ to n **do**
        Collect the location of all occurences $\{E_{i1}, \cdots, E_{im}\}$ to $L = \{l_1, \cdots, l_m\}$.
        for $j = 1$ to m **do**
            $\mathcal{G}_C[j][j] = 1$
            for $k = 1$ to m **do**
                $\mathcal{G}_C[j][k] = 1$
            end for
        end for
    end for
    return Constructed Graph $\mathcal{G}_C$.

Figure 3: Example of the adjacent matrix of Coreference Graph $\mathcal{G}_C$ and RST Graph $\mathcal{G}_R$.

RST Graph

As aforementioned, we adopt EDU as the minimal selection unit for our summarization model. Here, we further utilize the discourse trees of the document to capture the rhetorical structure of the document, and build a discourse-aware model for text summarization.

When selecting sentences as candidates for extractive summarization, we assume each sentence is grammatically self-contained. But for EDUs, some restrictions need to be considered to ensure grammaticality. For example, Figure 2 illustrates an RST discourse parse tree of a sentence, where "[2] This iconic ... series" is a grammatical sentence but "[3] and shows ... 8" is not by itself. We need to understand the dependencies between EDUs to ensure the grammaticality of the selected combinations. There are two steps to learn the derivation of dependencies: *head inheritance* and *tree conversion*.

Head inheritance defines the head node for each valid non-terminal tree node. For each leaf node, the head is itself. We determine the head node(s) of non-terminal nodes based on their nuclearity.[1] For example, in Figure 2, the head of text spans [1-5], [2-5], [3-5] and [4-5] need to be grounded to a single EDU.

In this way, we propose a simple yet effective schema to convert the RST discourse tree to a dependency-based discourse tree.[2] We always consider the dependency restriction such as the reliance of Satellite on Nucleus, when we create oracle during pre-processing and when the model makes the prediction. For the example shown in Figure 2, if the model selects "[5] being carried ... Liberia.", we will enforce the model to select "[3] and shows ... 8," and "[2] This ... series,". The dependencies in this example are $\{(4 \rightarrow 5), (5 \rightarrow 3), (3 \rightarrow 2), (1 \rightarrow 2)\}$.

---

[1]If both children are N(ucleus), then the head of the current node inherits the head of the left child. Otherwise, when one child is N and the other is S, the head of the current node inherits the head of the N child.

[2]The proposed schema is summarized as follows: If one child node is N and the other is S, the head of the S node depends on the head of the N node. If both children are N and the right child does not contain a subject in the discourse, the head of the right N node depends on the head of the left N node.
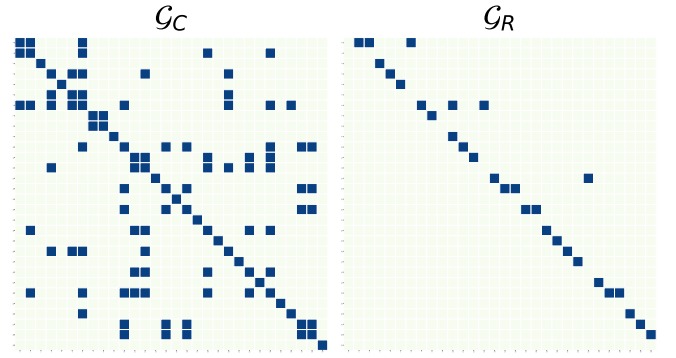
The construction of the RST Graph aims to provide not only local paragraph-level but also long-range document-level connections among the EDUs. We use the converted dependency version of the tree to build the RST Graph $\mathcal{G}_R$, by initializing an empty graph and treating every discourse dependency from the i-th EDU to the j-th EDU as a directed edge, i.e., $\mathcal{G}_R[i][j] = 1$.

Coreference Graph

Text summarization, especially news summarization, usually suffers from the well-known 'position bias' (Kedzie, McKeown, and Daume III 2018), where most of the crucial information is described at the very beginning of the document. However, there is still a decent amount of information spread in the middle or at the end of the document, which is often ignored by summarization models. We find that around 25% of oracle sentences appear after the first 10 sentences in the CNNDM dataset. Besides, in long news articles, there are often multiple core characters and events throughout the whole document. However, existing neural models are not good at modeling such long-range context, especially when there are multiple ambiguous coreferences to resolve.

To encourage and guide the model to capture the long-range context in the document, we propose a Coreference Graph built upon discourse units. Algorithm 1 describes how to construct the Coreference Graph. We first use Stanford CoreNLP (Manning et al. 2014) to detect all the coreference clusters in an article. For each coreference cluster, all the discourse units containing the mention of the same cluster will be connected. This process is iterated over all the coreference mention clusters to generate the final Coreference Graph.

Figure 1 provides an example, where 'Pulitzer prizes' is an important entity and has occurred multiple times in multiple discourse units. The constructed Coreference Graph is shown on the right side of the document. We intentionally ignore other entities and mentions in this example for simplicity. When graph $\mathcal{G}_C$ is constructed, edges among 1-1, 2-1, 20-1 and 22-1 are all connected due to the mentions of 'Pulitzer prizes'. Figure 3 shows an example of the two constructed graphs. $\mathcal{G}_C$ is symmetric and self-loop is added to all the nodes to prevent the graph from being too sparse.
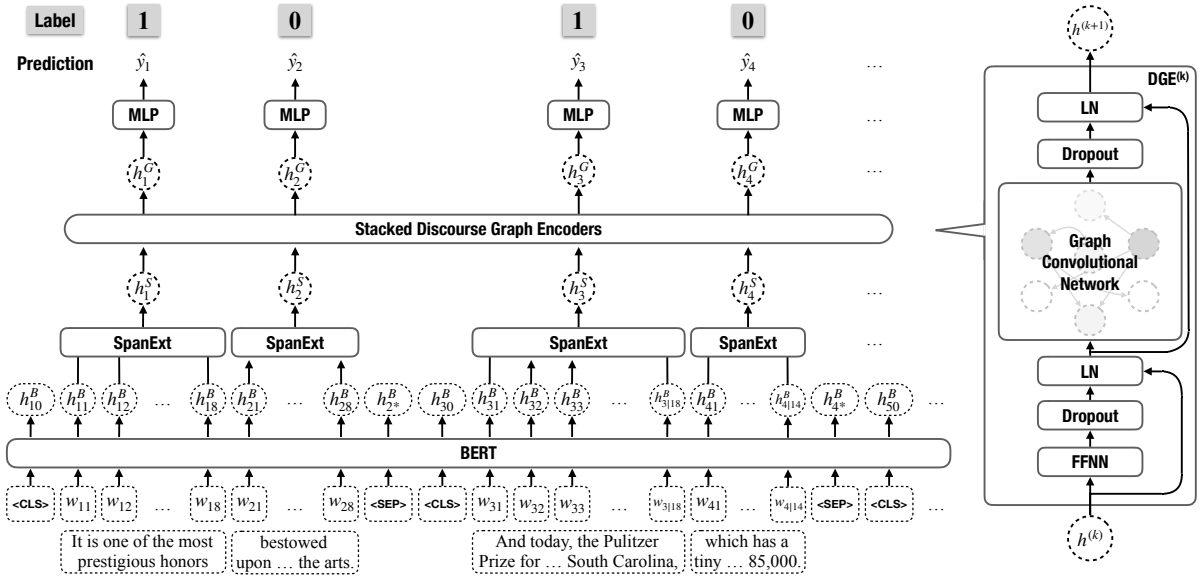
Figure 4: (Left) Model architecture of DISCOBERT. The Stacked Discourse Graph Encoders contain k stacked DGE blocks. (Right) The architecture of each Discourse Graph Encoder (DGE) block.

DiscoBERT Model

In this section, we present DISCOBERT, a BERT-based extractive summarization model, which takes EDUs as the minimal selection unit for redundancy reduction and uses discourse graphs to capture long-range dependencies between EDUs.

Model Overview

Figure 4 provides an overview of the proposed model, consisting of a Document Encoder, and a Graph Encoder. For the Document Encoder, a pre-trained BERT model is first used to encode the whole document on token level, and then a self-attentive span extractor is designed to obtain the EDU representations from the corresponding text spans. The Graph Encoder takes the output of the Document Encoder as input and updates the EDU representations with Graph Convolutional Network based on the constructed discourse graphs, which are then used to predict the oracle labels.

Assume that document D is segmented into n EDUs in total, i.e., $D = \{d_1, d_2, \cdots, d_n\}$, where d_i denotes the i-th EDU. Following Liu (2019), we formulate extractive summarization as a sequential labelling task, where each EDU d_i is scored by the neural networks, and decisions are made based on the scores of all EDUs. The oracle labels are a sequence of binary labels, where 1 stands for being selected and 0 for not. We denote the labels as $Y = \{y_1^*, y_2^*, \cdots, y_n^*\}$. During training, we aim to predict the sequence of labels Y given the document D. During inference, we need to further consider discourse dependency to ensure the coherence and grammaticality of the output summary.

Document Encoder

BERT is a pre-trained deep bidirectional Transformer encoder (Vaswani et al. 2017; Devlin et al. 2019). Following Liu (2019), we encode the whole document with BERT, and finetune the BERT model for summarization.

BERT is originally trained to encode a single sentence or sentence pair. However, a news article typically contains more than 500 words, hence we need to make some adaptation to apply BERT for document encoding. Specifically, we insert ⟨CLS⟩ and ⟨SEP⟩ tokens at the beginning and the end of each sentence, respectively.[3] In order to encode long documents such as news articles, we also extend the maximum sequence length that BERT can take from 512 to 768 in all our experiments.

The input document after tokenization is denoted as $D = \{d_1, \cdots, d_n\}$, and $d_i = \{w_{i1}, \cdots, w_{i\ell_i}\}$, where ℓ_i is the number of BPE tokens in the i-th EDU. If d_i is the first EDU in a sentence, there is also a ⟨CLS⟩ token prepended to d_i; if d_j is the last EDU in a sentence, there is a ⟨SEP⟩ token appended to d_j (see Figure 4 for illustration). These two tokens are not shown in the equations for simplicity. The BERT model is then used to encode the document as:

$$\{\mathbf{h}_{11}^B, \cdots, \mathbf{h}_{n\ell_n}^B\} = \text{BERT}(\{w_{11}, \cdots, w_{n\ell_n}\}), \quad (1)$$

where $\{\mathbf{h}_{11}^B, \cdots, \mathbf{h}_{n\ell_n}^B\}$ is the BERT output of the whole document in the same length as the input sequence.

After the BERT encoder, the representation of the ⟨CLS⟩ token can be used as sentence representation. However, this approach does not work in our setting, since we need to extract the representation for EDUs instead. Therefore, we adopt a Self-Attentive Span Extractor (SpanExt), proposed in Lee et al. (2017), to learn the representation of EDUs.

For the i-th EDU with ℓ_i words, with the output from the BERT encoder $\{\mathbf{h}_{i1}^B, \mathbf{h}_{i2}^B, \cdots, \mathbf{h}_{i\ell_i}^B\}$, we obtain the EDU

---

[3] We also tried inserting ⟨CLS⟩ and ⟨SEP⟩ at the beginning and the end of every EDU, and treating the corresponding ⟨CLS⟩ representation as the representation for each EDU, but the performance drops drastically.

representation as follows:

$$\alpha_{ij} = \mathbf{W}_2 \cdot \text{ReLU}(\mathbf{W}_1 \mathbf{h}_{ij}^B + \mathbf{b}_1) + \mathbf{b}_2 \quad (2)$$

$$\mathbf{a}_{ij} = \frac{\exp(\alpha_{ij})}{\sum_{k=1}^{\ell_i} \exp(\alpha_{ik})}, \quad \mathbf{h}_i^S = \sum_{j=1}^{\ell_i} \mathbf{a}_{ij} \cdot \mathbf{h}_{ij}^B, \quad (3)$$

where α_{ij} is the score of the j-th word in the EDU, $\mathbf{a}_{ij}$ is the normalized attention of the j-th word w.r.t. all the words in the span. $\mathbf{h}_i^S$ is a weighted sum of the BERT output hidden states. Throughout the paper, all the $\mathbf{W}$ matrices and $\mathbf{b}$ vectors are parameters to learn. We abstract the above Self-Attentive Span Extractor as $\mathbf{h}_i^S = \text{SpanExt}(\mathbf{h}_{i1}^B, \cdots, \mathbf{h}_{i\ell_i}^B)$.

After the span extraction step, the whole document is represented as a sequence of EDU representations: $\mathbf{h}^S = \{\mathbf{h}_1^S, \cdots, \mathbf{h}_n^S\} \in \mathbb{R}^{d_h \times n}$, which will be sent into the graph encoder.

Graph Encoder

Given the constructed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the nodes $\mathcal{V}$ correspond to the EDUs in a document, and the edges $\mathcal{E}$ correspond to either the RST discourse relations or the coreference mentions. We then use Graph Convolutional Network to update the representations of all the EDUs, to capture long-range dependencies missed by BERT for better summarization. To modularize the architecture design, we present a single Discourse Graph Encoder (DGE) layer here. Multiple DGE layers are stacked in our experiments.

Assume that the input for the k-th DGE layer is denoted as $\mathbf{h}^{(k)} = \{\mathbf{h}_1^{(k)}, \ldots, \mathbf{h}_n^{(k)}\} \in \mathbb{R}^{d_h \times n}$, and the corresponding output is denoted as $\mathbf{h}^{(k+1)} = \{\mathbf{h}_1^{(k+1)}, \ldots, \mathbf{h}_n^{(k+1)}\} \in \mathbb{R}^{d_h \times n}$. The k-th DGE layer is designed as follows:

$$\mathbf{u}_i^{(k)} = \mathbf{W}_4^{(k)} \text{ReLU}(\mathbf{W}_3^{(k)} \mathbf{h}_i^{(k)} + \mathbf{b}_3^{(k)}) + \mathbf{b}_4^{(k)} \quad (4)$$

$$\mathbf{v}_i^{(k)} = \text{LN}(\mathbf{h}_i^{(k)} + \text{Dropout}(\mathbf{u}_i^{(k)})) \quad (5)$$

$$\mathbf{w}_i^{(k)} = \text{ReLU}\Big(\sum_{j \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_i|} \mathbf{W}_5^{(k)} \mathbf{v}_j^{(k)} + \mathbf{b}_5^{(k)}\Big) \quad (6)$$

$$\mathbf{h}_i^{(k+1)} = \text{LN}(\text{Dropout}(\mathbf{w}_i^{(k)}) + \mathbf{v}_i^{(k)}), \quad (7)$$

where $\text{LN}(\cdot)$ represents Layer Normalization, $\mathcal{N}_i$ denotes the neighborhood of the i-th EDU node. $\mathbf{h}_i^{(k+1)}$ is the output of the i-th EDU in the k-th DGE layer, and $\mathbf{h}^{(1)} = \mathbf{h}^S$, which is the output from the Document Encoder. After K layers of graph propagation, we obtain $\mathbf{h}^G = \mathbf{h}^{(K+1)} \in \mathbb{R}^{d_h \times n}$, which is the final representation of all the EDUs after the stacked DGE layers. For different graphs the parameter of DGEs are not shared. If we use both graphs, we concatenate the output of two graphs:

$$\mathbf{h}^G = \text{ReLU}(\mathbf{W}_6[\mathbf{h}_C^G; \mathbf{h}_R^G] + \mathbf{b}_6) \quad (8)$$

Training & Inference

During training, $\mathbf{h}^G$ is used for predicting the oracle labels. Specifically,

$$\hat{y}_i = \sigma(\mathbf{W}_7 \mathbf{h}_i^G + \mathbf{b}_7), \quad (9)$$

| Dataset | Document | | | Sum. | # $\mathcal{E}$ in Graph | |
	# sent.	# EDU	# tok.	# tok.	$\mathcal{G}_R$	$\mathcal{G}_C$
CNNDM	24	67	541	54	66	233
NYT	22	66	591	87	65	143

Table 1: Statistics of the datasets. The first block shows the average number of sentences, EDUs and tokens in the documents. The second block shows the average number of tokens in the reference summaries. The third block shows the average number of edges in the constructed RST Graphs ($\mathcal{G}_R$) and Coreference Graphs ($\mathcal{G}_C$), respectively.

where $\sigma(\cdot)$ represents the logistic function, and $\hat{y}_i$ is the prediction probability ranging from 0 to 1. The training loss of the model is the binary cross-entropy loss given the predictions and oracles:

$$\mathcal{L} = -\sum_{i=1}^{n} (y_i^* \log(\hat{y}_i) + (1 - y_i^*) \log(1 - \hat{y}_i)), \quad (10)$$

The above loss is summed over all the training samples. For DISCOBERT without graphs, the output from Document Encoder $\mathbf{h}^S$ is used for prediction instead. The creation of oracle is operated on EDU level. We greedily pick up EDUs with their necessary dependencies until the R-1 F_1 drops.

During inference, for an input document, after obtaining the prediction probabilities of all the EDUs, i.e., $\hat{\mathbf{y}} = \{\hat{y}_1, \cdots, \hat{y}_n\}$, we sort $\hat{\mathbf{y}}$ in descending order, and select EDUs accordingly. Note that the dependencies between the EDUs are also enforced in prediction to ensure grammacality of generated summaries.

Experiments

In this section, we present experimental results on three popular news summarization datasets. We compare our proposed model with state-of-the-art baselines, and conduct detailed analysis to validate the effectiveness of DiscoBERT.

Datasets

We evaluate the proposed models on three datasets: New York Times (NYT) (Sandhaus 2008), CNN and Dailymail (CNNDM) (Hermann et al. 2015). We use the script from See, Liu, and Manning (2017) to extract summaries from raw data. We use Stanford CoreNLP for sentence boundary detection, tokenization and parsing (Manning et al. 2014). Due to the limitation of BERT, we only encode up to 768 BERT BPEs.

Table 1 shows the statistics of the datasets. The edges in $\mathcal{G}_C$ are undirected, while the ones in $\mathcal{G}_R$ are directional. For CNNDM, there are 287,226, 13,368 and 11,490 samples for training, validation and test, respectively. We use the unanonymized version as in previous summarization work. For NYT, it is licensed by LDC[4], and following previous work (Zhang, Wei, and Zhou 2019; Xu and Durrett 2019), there are 137,778, 17,222 and 17,223 samples for training, validation and test, respectively.

---

[4]https://catalog.ldc.upenn.edu/LDC2008T19

Model	R-1	R-2	R-L
Lead3	40.42	17.62	36.67
Oracle (Sentence)	55.61	32.84	51.88
Oracle (Discourse)	61.61	37.82	59.27
NeuSum (Zhou et al. 2018)	41.59	19.01	37.98
BanditSum (Dong et al. 2018)	41.50	18.70	37.60
JECS (Xu and Durrett 2019)	41.70	18.50	37.90
PNBERT (Zhong et al. 2019)	42.39	19.51	38.69
PNBERT w. RL	42.69	19.60	38.85
BERT (Zhang, Wei, and Zhou 2019)	41.82	19.48	38.30
HIBERT$_S$	42.10	19.70	38.53
HIBERT$_S^*$	42.31	19.87	38.78
HIBERT$_M^*$	42.37	19.95	38.83
BERTSUM (Liu 2019)	**43.25**	20.24	**39.63**
T5-Base (Raffel et al. 2019)	42.05	**20.34**	39.40
BERT	43.07	19.94	39.44
DISCOBERT	43.38	20.44	40.21
DISCOBERT w. $\mathcal{G}_C$	43.58	20.64	40.42
DISCOBERT w. $\mathcal{G}_R$	43.68	20.71	40.54
DISCOBERT w. $\mathcal{G}_R$ & $\mathcal{G}_C$	**43.77**	**20.85**	**40.67**

Table 2: Results on the test set of the CNNDM dataset. ROUGE-1, -2 and -L F_1 are reported. Models with the asterisk symbol (*) used extra data for pre-training. R-1 and R-2 are shorthands for unigram and bigram overlap; R-L is the longest common subsequence.

State-of-the-art Baselines

We compare the proposed model with the following state-of-the-art neural text summarization models.

Extractive Models BanditSum treats extractive summarization as a contextual bandit problem, and is trained with policy gradient methods (Dong et al. 2018). **NeuSum** is an extractive model with seq2seq architecture, where the attention mechanism scores the document and emits the index as the selection (Zhou et al. 2018). **DeepChannel** is an extractive model with salience estimation and contrastive training strategy (Shi et al. 2019).

Compressive Models JECS is a neural text-compression-based summarization model using BLSTM as the encoder (Xu and Durrett 2019). The first stage is selecting sentences, and the second stage is sentence compression with pruning constituency parsing tree.

BERT-based Models BERT-based models have achieved significant improvement on CNNDM and NYT, when compared with their LSTM counterparts. Specifically, **BertSum** is the first BERT-based extractive summarization model (Liu 2019). Our baseline model BERT is the re-implementation of BertSum. **PNBert** proposed a BERT-based model with various training strategies, including reinforcement learning and Pointer Networks (Zhong et al. 2019). **HiBert** is a hierarchical BERT-based model for document encoding, which is further pretrained with unlabeled data (Zhang, Wei, and Zhou 2019).

Implementation Details

We use AllenNLP (Gardner et al. 2018) as the code framework. Experiments are conducted on a single NVIDIA P100

Model	R-1	R-2	R-L
Lead3	41.80	22.60	35.00
Oracle (Sentence)	64.22	44.57	57.27
Oracle (Discourse)	67.76	48.05	62.40
JECS (Xu and Durrett 2019)	45.50	25.30	38.20
BERT (Zhang, Wei, and Zhou 2019)	48.38	29.04	40.53
HIBERT$_S$	48.92	29.58	41.10
HIBERT$_M$	49.06	29.70	41.23
HIBERT$_S^*$	49.25	29.92	41.43
HIBERT$_M^*$	**49.47**	**30.11**	**41.63**
BERT	48.48	29.01	40.62
DISCOBERT	49.78	30.30	42.44
DISCOBERT w. $\mathcal{G}_C$	49.79	30.18	42.48
DISCOBERT w. $\mathcal{G}_R$	49.86	30.25	42.55
DISCOBERT w. $\mathcal{G}_R$ & $\mathcal{G}_C$	**50.00**	**30.38**	**42.70**

Table 3: Results on the test set of the NYT dataset. Models with the asterisk symbol (*) used extra data for pre-training.

card, and the mini-batch size is set to 6 due to GPU memory capacity. The length of each document is truncated to 768 BPEs. We use the 'bert-base-uncased' model for all experiments. We train all our models for up to 80,000 steps. ROUGE (Lin 2004) is used as the evaluation metrics, and 'R-1' is used as the validation criteria.

The realization of discourse units and structure is a critical part of EDU pre-processing, which requires two steps: discourse segmentation and RST parsing. In the segmentation phase, we use a neural discourse segmenter based on the BiLSTM CRF framework (Wang, Li, and Yang 2018)[5]. The segmenter achieved 94.3 F_1 score on the RST-DT test set, in which the human performance is 98.3. In the parsing phase, we use a shift-reduce discourse parser to extract relations and identify neuclrity (Ji and Eisenstein 2014)[6].

Experimental Results

Results on CNNDM Table 2 shows the results on CNNDM. The first section includes the Lead3 baseline, the sentence-based oracle, and the discourse-based oracle. The second section lists the performance of baseline models, including non-BERT-based and BERT-based variants. The performance of our proposed model is listed in the third section. BERT is our implementation of the sentence-based BERT model. DISCOBERT is our discourse-based BERT model without Discourse Graph Encoder. DISCOBERT w. $\mathcal{G}_C$ and DISCOBERT w. $\mathcal{G}_R$ are the discourse-based BERT model with Coreference Graph and RST Graph, respectively. DISCOBERT w. $\mathcal{G}_R$ & $\mathcal{G}_C$ is the fusion model encoding both graphs.

The proposed DISCOBERT beats the sentence-based counterpart and all the competitor models. With the help of Discourse Graph Encoder, the graph-based DISCOBERT beats the state-of-the-art BERT model by a significant margin (0.52/0.61/1.04 on R-1/-2/-L on F_1). Ablation study with individual graphs shows that the RST Graph is slightly more

[5]https://github.com/PKU-TANGENT/NeuralEDUSeg
[6]https://github.com/jiyfeng/DPLP

Source	M	All	CR	PV	PT	O
CNNDM	Sent	33.0	18.7	9.0	2.3	3.0
	Disco	34.0	18.3	8.4	2.6	4.7
NYT	Sent	23.3	13.5	5.9	0.8	3.1
	Disco	23.8	13.9	5.7	0.8	3.4

Table 4: Number of errors per 10,000 characters based on automatic grammaticality checking on CNNDM and NYT with Grammarly, where *lower values are better*. Detailed error categories, including correctness (CR), passive voice (PV) misuse, punctuation (PT) in compound/complex sentences and others (O), are listed from left to right.

Model	All	Coherence	Grammaticality
Sent	3.45 ± 0.87	3.30 ± 0.90	3.45 ± 1.06
Disco	3.24 ± 0.84	3.15 ± 0.95	3.25 ± 1.02
Ref	3.28 ± 0.99	3.12 ± 0.94	3.29 ± 1.06

Table 5: Human evaluation on model outputs. We ask Turkers to grade the overall preference, coherence and grammaticality based on 1 to 5 scale. The mean value along with the standard deviation is shown.

helpful than the Coreference Graph, while the combination of both achieves better performance overall.

Results on NYT Results on the NYT dataset are summarized in Table 3. The proposed model surpasses previous state-of-the-art BERT-based model by a significant margin. HIBERT$_S^*$ and HIBERT$_M^*$ used extra data for pre-training the model. We notice that in the NYT dataset, most of the improvement comes from the use of EDUs as minimial selection units. DISCOBERT provides 1.30/1.29/1.82 gain on R-1/-2/-L over the BERT baseline. However, the use of discourse graphs does not help much in this case.

Grammaticality Due to the segmentation and partial selection of the sentence, the output of our model might not be as grammatical as the original sentence. We manually examined and automatically evaluated the model output, and observed that overall, the generated summaries are still grammatical, given the RST dependency tree constraining the rhetorical relations among EDUs. A set of simple yet effective post-processing rules helps to complete the EDUs in some cases.

Table 4 shows automatic grammatical checking results using Grammarly, where the average number of errors in every 10,000 characters on CNNDM and NYT datasets is reported. We compare DISCOBERT with the sentence-based BERT model. 'All' shows the summation of the number of errors in all categories. As shown in the table, the summaries generated by our model have retained the quality of the original text.

We also conduct human evaluation on the model outputs. We sampled 200 documents from the test set of CNNDM, and for each sample we asked two Turkers to compare three summaries and grade them on the basis of 1 to 5 scale. Results are shown in Table 5. The Sent-BERT model (the original BERTSum model) selects sentences from the document, hence it provides the best overall readability, coherence and grammaticality. In some cases reference summaries are just long phrases, so the scores are slightly lower than those of the sentence model. Our DISCOBERT model is slightly worse than the Sent-BERT model, but is fully comparable to the other two variants.

Error Analysis Despite the sucess, we furhter conducted error analysis, and found that the errors mostly originated from punctuation and coherence. Common punctuation issues in-

clude extra or missing commas, as well as missing quotation marks. For example, if we only select the first EDU of the sentence "['Johnny is believed to have drowned,]$_1$ [but actually he is fine,']$_2$ [the police say.]$_3$", the output "'Johnny is believed to have drowned." does not look like a grammatical sentence due to the punctuation. The coherence issue originates from the missing or improper pronoun resolution. As shown in the above example, only selecting the second EDU yields a sentence "actually he is fine", which is not clear who is 'he' mentioned here.

Related Work

Neural Extractive Summarization Neural networks have been widely used in extractive summarization. Various decoding approaches, including ranking (Narayan, Cohen, and Lapata 2018), index prediction (Zhou et al. 2018) and sequential labelling (Nallapati, Zhai, and Zhou 2017; Zhang et al. 2018; Dong et al. 2018), have been applied to content selection. Our model uses similar configuration to encode the document with BERT as Liu (2019) did, but we use discourse graph structure and graph encoder to handle the long-range dependency issue.

Neural Compressive Summarization Text summarization with compression and deletion has been explored in some recent work. Xu and Durrett (2019) presented a two-stage neural model for selection and compression based on constituency tree pruning. Dong et al. (2019) presented a neural sentence compression model with discrete operations including deletion and addition. Different from these studies, as we use EDUs as minimal selection basis, sentence compression is achieved automatically in our model.

EDU for Summarization The use of discourse theory for text summarization has been explored before. Louis, Joshi, and Nenkova (2010) examined the benefit of graph structure provided by discourse relations for text summarization. Hirao et al.; Yoshida et al. (2013; 2014) formulated the summarization problem as the trimming of the document discourse tree. Durrett, Berg-Kirkpatrick, and Klein (2016) presented a system of sentence extraction and compression with ILP methods using discourse structure. Li, Thadani, and Stent (2016) demonstrated that using EDUs as units of content selection leads to stronger summarization performance. Compared with them, our proposed method is the first neural end-to-end summarization model using EDUs as selection basis.

Graph-based Summarization Graph approach has been explored in text summarization over decades. LexRank in-

troduced a stochastic graph-based method for computing relative importance of textual units (Erkan and Radev 2004). Yasunaga et al. (2017) employed a GCN on the relation graphs with sentence embeddings obtained from RNN. Tan, Wan, and Xiao (2017) also proposed a graph-based attention mechanism in abstractive summarization model.

Conclusions

In this paper, we present DISCOBERT for text summarization. DISCOBERT uses discourse unit as the minimal selection basis to reduce summarization redundancy, and leverages two constructed discourse graphs as inductive bias to capture long-range dependencies among discourse units for better summarization. We validate our proposed approach on two popular datasets, and observe consistent improvement over baseline methods. For future work, we will explore better graph encoding methods, and apply discourse graphs to other tasks that require long document encoding.

References

[2016] Cao, Z.; Li, W.; Li, S.; Wei, F.; and Li, Y. 2016. AttSum: Joint learning of focusing and summarization with neural attention. In *COLING*.

[2018] Cao, Z.; Wei, F.; Li, W.; and Li, S. 2018. Faithful to the Original: Fact Aware Neural Abstractive Summarization. In *AAAI*.

[2001] Carlson, L.; Marcu, D.; and Okurovsky, M. E. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *SIGDIAL*.

[2018] Celikyilmaz, A.; Bosselut, A.; He, X.; and Choi, Y. 2018. Deep communicating agents for abstractive summarization. In *NAACL*.

[2018] Chen, Y.-C., and Bansal, M. 2018. Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. In *ACL*.

[2016] Cheng, J., and Lapata, M. 2016. Neural summarization by extracting sentences and words. In *ACL*.

[2019] Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

[2018] Dong, Y.; Shen, Y.; Crawford, E.; van Hoof, H.; and Cheung, J. C. K. 2018. BanditSum: Extractive Summarization as a Contextual Bandit. In *EMNLP*.

[2019] Dong, Y.; Li, Z.; Rezagholizadeh, M.; and Cheung, J. C. K. 2019. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In *ACL*.

[2016] Durrett, G.; Berg-Kirkpatrick, T.; and Klein, D. 2016. Learning-Based Single-Document Summarization with Compression and Anaphoricity Constraints. In *ACL*.

[2004] Erkan, G., and Radev, D. R. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *JAIR*.

[2018] Gardner, M.; Grus, J.; Neumann, M.; Tafjord, O.; Dasigi, P.; Liu, N. F.; Peters, M.; Schmitz, M.; and Zettle-moyer, L. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*.

[2018] Gehrmann, S.; Deng, Y.; and Rush, A. 2018. Bottom-Up Abstractive Summarization. In *EMNLP*.

[2015] Hermann, K. M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching Machines to Read and Comprehend. In *NeurIPS*.

[2013] Hirao, T.; Yoshida, Y.; Nishino, M.; Yasuda, N.; and Nagata, M. 2013. Single-document summarization as a tree knapsack problem. In *EMNLP*.

[2014] Ji, Y., and Eisenstein, J. 2014. Representation learning for text-level discourse parsing. In *ACL*.

[2018] Kedzie, C.; McKeown, K.; and Daume III, H. 2018. Content selection in deep learning models of summarization. In *EMNLP*.

[2017] Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

[2017] Lee, K.; He, L.; Lewis, M.; and Zettlemoyer, L. 2017. End-to-end neural coreference resolution. In *EMNLP*.

[2016] Li, J. J.; Thadani, K.; and Stent, A. 2016. The role of discourse units in near-extractive summarization. In *SIGDIAL*.

[2004] Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*.

[2019] Liu, Y. 2019. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*.

[2010] Louis, A.; Joshi, A.; and Nenkova, A. 2010. Discourse indicators for content selection in summarization. In *SIGDIAL*.

[1988] Mann, W. C., and Thompson, S. A. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*.

[2014] Manning, C.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.; and McClosky, D. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *ACL*.

[2017] Nallapati, R.; Zhai, F.; and Zhou, B. 2017. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In *AAAI*.

[2018] Narayan, S.; Cohen, S. B.; and Lapata, M. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *NAACL*.

[2019] Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

[2015] Rush, A. M.; Chopra, S.; and Weston, J. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *EMNLP*.

[2008] Sandhaus, E. 2008. The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia*.

[2017] See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *ACL*.

[2019] Sharma, E.; Huang, L.; Hu, Z.; and Wang, L. 2019. An entity-driven framework for abstractive summarization. In *EMNLP*.

[2019] Shi, J.; Liang, C.; Hou, L.; Li, J.; Liu, Z.; and Zhang, H. 2019. Deepchannel: Salience estimation by contrastive learning for extractive document summarization. In *AAAI*.

[2017] Tan, J.; Wan, X.; and Xiao, J. 2017. Abstractive document summarization with a graph-based attentional neural model. In *ACL*.

[2017] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

[2018] Wang, Y.; Li, S.; and Yang, J. 2018. Toward fast and accurate neural discourse segmentation. In *EMNLP*.

[2019] Xu, J., and Durrett, G. 2019. Neural extractive text summarization with syntactic compression. In *EMNLP*.

[2017] Yao, J.-g.; Wan, X.; and Xiao, J. 2017. Recent advances in document summarization. *Knowledge and Information Systems* 53(2):297–336.

[2017] Yasunaga, M.; Zhang, R.; Meelu, K.; Pareek, A.; Srinivasan, K.; and Radev, D. 2017. Graph-based neural multi-document summarization. In *CoNLL*.

[2014] Yoshida, Y.; Suzuki, J.; Hirao, T.; and Nagata, M. 2014. Dependency-based discourse parser for single-document summarization. In *EMNLP*.

[2018] Zhang, X.; Lapata, M.; Wei, F.; and Zhou, M. 2018. Neural Latent Extractive Document Summarization. In *EMNLP*.

[2019] Zhang, X.; Wei, F.; and Zhou, M. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *ACL*.

[2019] Zhong, M.; Liu, P.; Wang, D.; Qiu, X.; and Huang, X. 2019. Searching for effective neural extractive summarization: What works and what's next. In *ACL*.

[2018] Zhou, Q.; Yang, N.; Wei, F.; Huang, S.; Zhou, M.; and Zhao, T. 2018. Neural Document Summarization by Jointly Learning to Score and Select Sentences. In *ACL*.