A Discourse-Aware Neural Extractive Model for Text Summarization

Jiacheng Xu[*1], **Zhe Gan**[2], **Yu Cheng**[2], **Jingjing Liu**[2]
[1]The 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 for document encoding in state-of-the-art text summarization models. However, sentence-based extractive models often result in redundant or uninformative phrases in the extracted summaries. Also, long-range dependencies throughout a document are not well captured by BERT, which is pre-trained on sentence pairs instead of documents. To address these issues, we present a discourse-aware neural summarization model - DISCOBERT[1]. DISCOBERT extracts sub-sentential discourse units (instead of sentences) as candidates for extractive selection on a finer granularity. To capture the long-range dependencies among discourse units, structural discourse graphs are constructed based on RST trees and coreference mentions, encoded with Graph Convolutional Networks. Experiments show that the proposed model outperforms state-of-the-art methods by a significant margin on popular summarization benchmarks compared to other BERT-base models.

1 Introduction

Neural networks have achieved great success in the task of text summarization (Nenkova et al., 2011; Yao et al., 2017). There are two main lines of research: abstractive and extractive. While the abstractive paradigm (Rush et al., 2015; See et al., 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; Zhou et al., 2018; Narayan et al., 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).

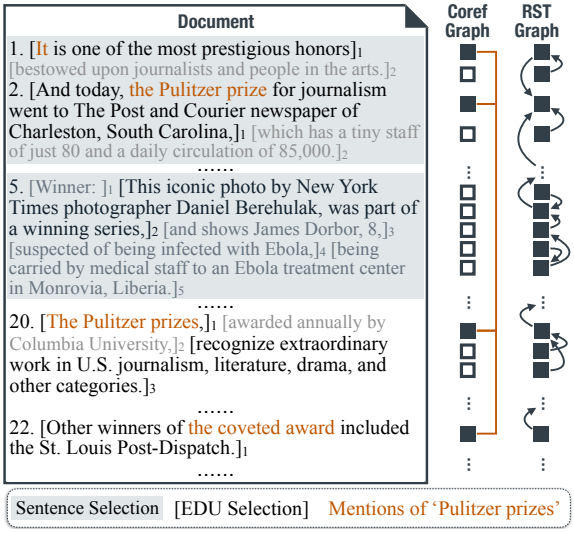

Figure 1: Illustration of DISCOBERT for text summarization. Sentence-based BERT model (baseline) selects whole sentences 1, 2 and 5. The proposed discourse-aware model DISCOBERT selects EDUs {1-1, 2-1, 5-2, 20-1, 20-3, 22-1}. The right side of the figure illustrates the two discourse graphs we use: (i) Coref(erence) Graph (with the mentions of 'Pulitzer prizes' highlighted as examples); and (ii) RST Graph (induced by RST discourse trees).

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 et al., 2018; Zhang et al., 2018; Xu and Durrett, 2019). Compression or rewriting aims to discard uninformative phrases in the selected sentences. However, most of these hybrid systems suffer from the inevitable disconnection between the two stages in the pipeline.

Meanwhile, modeling long-range context for document summarization remains a challenge (Xu

[*] Most of this work was done when the first author was an intern at Microsoft.

[1]Code, illustration and datasets are available at: https://github.com/jiacheng-xu/DiscoBERT.

et al., 2016). Pre-trained language models (Devlin et al., 2019) are designed mostly for sentences or a short paragraph, thus poor at capturing long-range dependencies throughout a document. Empirical observations (Liu and Lapata, 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 paper, we present DISCOBERT, a discourse-aware neural extractive summarization model built upon BERT. To perform compression with extraction simultaneously and reduce redundancy across sentences, we take Elementary Discourse Unit (EDU), a sub-sentence phrase unit originating from RST (Mann and Thompson, 1988; Carlson et al., 2001)[2] as the minimal selection unit (instead of sentence) for extractive summarization. Figure 1 shows an example of discourse segmentation, with sentences broken down into EDUs (annotated with brackets). By operating on the discourse unit level, our model can discard redundant details in sub-sentences, therefore retaining additional capacity to include more concepts or events, leading to more concise and informative summaries.

Furthermore, we finetune the representations of discourse units with the injection of prior knowledge to leverage intra-sentence discourse relations. More specifically, two discourse-oriented graphs are proposed: RST Graph $\mathcal{G}_R$ and Coreference Graph $\mathcal{G}_C$. Over these discourse graphs, Graph Convolutional Network (GCN) (Kipf and Welling, 2017) is imposed to capture long-range interactions among EDUs. *RST Graph* is constructed from RST parse trees over EDUs of the document. On the other hand, *Coreference Graph* connects entities and their coreference clusters/mentions across the document. The path of coreference navigates the model from the core event to other occurrences of that event, and in parallel explores its interactions with other concepts or events.

The main contribution is threefold: (i) We propose a discourse-aware extractive summarization model, DISCOBERT, which operates on a sub-sentential discourse unit level to generate concise and informative summary with low redundancy. (ii) We propose to structurally model inter-sentential context with two types of discourse graph. (iii) DISCOBERT achieves new state of the art on two popular newswire text summarization datasets, outperforming other BERT-base models.

[2]We adopt RST as the discourse framework due to the availability of existing tools, the nature of the RST tree structure for compression, and the observations from Louis et al. (2010). Other alternatives includes Graph Bank (Wolf and Gibson, 2005) and PDTB (Miltsakaki et al., 2004).

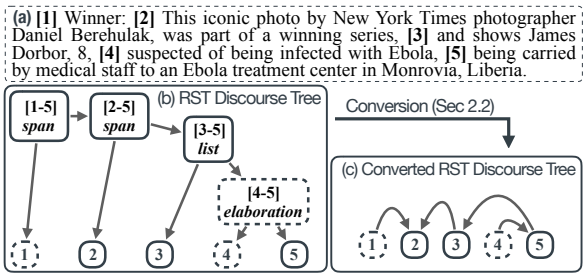

Figure 2: Example of discourse segmentation and RST tree conversion. The original sentence is segmented into 5 EDUs in box (a), and then parsed into an RST discourse tree in box (b). The converted dependency-based RST discourse tree is shown in box (c). Nucleus nodes including [2], [3] and [5], and Satellite nodes including [2] and [4] are denoted by solid lines and dashed lines, respectively. *Relations* are in italic. The EDU [2] is the head of the whole tree (span [1-5]), while the EDU [3] is the head of the span [3-5].

2 Discourse Graph Construction

In this section, we first introduce the Rhetorical Structure Theory (RST) (Mann and Thompson, 1988), a linguistic theory for discourse analysis, and then explain how we construct discourse graphs used in DISCOBERT. Two types of discourse graph are considered: RST Graph and Coreference Graph. All edges are initialized as disconnected, and connections are later added for a subset of nodes based on RST discourse parse tree or coreference mentions.

2.1 Discourse Analysis

Discourse analysis focuses on inter-sentential relations in a document or conversation. In the RST framework, 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 content and grammatical reliance. There are dependencies among EDUs that represent their rhetorical relations.

In this work, we treat EDU as the minimal unit for content selection in text summarization. Fig-

ure 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. As observed in Louis et al. (2010), the RST tree structure already serves as a strong indicator for content selection. On the other hand, the agreement between rhetorical relations tends to be lower and more ambiguous. Thus, we do not encode rhetorical relations explicitly in our model.

In content selection for text summarization, we expect the model to select the most concise and pivotal concept in the document, with low redundancy.[3] However, in traditional extractive summarization methods, the model is required to select a *whole* sentence, even though some parts of the sentence are not necessary. Our proposed approach can select one or several fine-grained EDUs to render the generated summaries less redundant. This serves as the foundation of our DISCOBERT model.

2.2 RST Graph

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. We need to understand the dependencies between EDUs to ensure the grammaticality of the selected combinations. The detail of the derivation of the dependencies could be found in Sec 4.3.

The construction of the RST Graph aims to provide not only local paragraph-level but also long-range document-level connections among 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.*, $\mathcal{G}_R[i][j] = 1$.

2.3 Coreference Graph

Text summarization, especially news summarization, usually suffers from the well-known 'position bias' issue (Kedzie et al., 2018), where most of the key 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 observe 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 poor at modeling such long-range context, especially when there are multiple ambiguous coreferences to resolve.

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, $k = 1$ to m **do**
    $\mathcal{G}_C[j][k] = 1$
  end for
end for
return Constructed Graph $\mathcal{G}_C$.

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 create 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[4]. 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'.

3 DISCOBERT Model

3.1 Overview

Figure 3 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

[3]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.

[4]We intentionally ignore other entities and mentions in this example for simplicity.

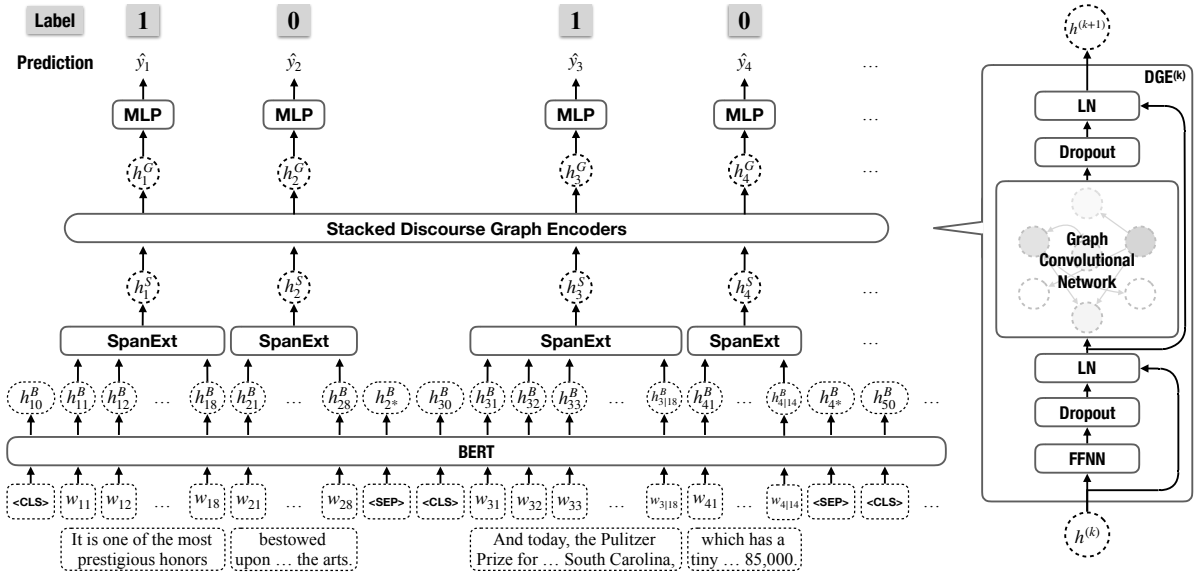

Figure 3: (Left) Model architecture of DISCOBERT. The Stacked Discourse Graph Encoders contain k stacked DGE blocks. (Right) The architecture of each Discourse Graph Encoder (DGE) block.

whole document on the token level. Then, a self-attentive span extractor is used 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 and Lapata (2019), we formulate extractive summarization as a sequential labeling task, where each EDU d_i is scored by 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.

3.2 Document Encoder

BERT is a pre-trained deep bidirectional Transformer encoder (Vaswani et al., 2017; Devlin et al., 2019). Following Liu and Lapata (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.[5] 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 3). The schema of insertion of ⟨CLS⟩ and ⟨SEP⟩ is an approach used in Liu and Lapata (2019). For simplicity, these two tokens are not shown in the equations. BERT model is then used to encode the document:

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

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.

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

[5] 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.

Self-Attentive Span Extractor (SpanExt), proposed in Lee et al. (2017), to learn EDU representation.

For the i-th EDU with ℓ_i words, with the output from the BERT encoder $\{\mathbf{h}^B_{i1}, \mathbf{h}^B_{i2}, \cdots, \mathbf{h}^B_{i\ell_i}\}$, we obtain EDU representation as follows:

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

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

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. $\mathbf{h}^S_i$ 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 $\mathbf{h}^S_i = \text{SpanExt}(\mathbf{h}^B_{i1}, \cdots, \mathbf{h}^B_{i\ell_i})$.

After the span extraction step, the whole document is represented as a sequence of EDU representations: $\mathbf{h}^S = \{\mathbf{h}^S_1, \cdots, \mathbf{h}^S_n\} \in \mathbb{R}^{d_h \times n}$, which will be sent to the graph encoder.

3.3 Graph Encoder

Given the constructed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, nodes $\mathcal{V}$ correspond to the EDUs in a document, and edges $\mathcal{E}$ correspond to either RST discourse relations or 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 architecture design, we present a single Discourse Graph Encoder (DGE) layer. Multiple DGE layers are stacked in our experiments.

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

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

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

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

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

where $\text{LN}(\cdot)$ represents Layer Normalization, $\mathcal{N}_i$ denotes the neighorhood of the i-th EDU node. $\mathbf{h}^{(k+1)}_i$ 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, their output are concatenated: $\mathbf{h}^G = \text{ReLU}(\mathbf{W}_6[\mathbf{h}^G_C; \mathbf{h}^G_R] + \mathbf{b}_6)$.

Dataset	Document # sent.	Document # EDU	Document # tok.	Sum. # tok.	# $\mathcal{E}$ in Graph $\mathcal{G}_R$	# $\mathcal{E}$ in Graph $\mathcal{G}_C$
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.

3.4 Training & Inference

During training, $\mathbf{h}^G$ is used for predicting the oracle labels. Specifically, $\hat{y}_i = \sigma(\mathbf{W}_7 \mathbf{h}^G_i + \mathbf{b}_7)$ 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 binary cross-entropy loss given the predictions and oracles: $\mathcal{L} = -\sum_{i=1}^{n} (y^*_i \log(\hat{y}_i) + (1 - y^*_i) \log(1 - \hat{y}_i))$. 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 R-1 F_1 drops.

During inference, given 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 EDUs are also enforced in prediction to ensure grammacality of generated summaries.

4 Experiments

In this section, we present experimental results on two 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.

4.1 Datasets

We evaluate the models on two datasets: New York Times (NYT) (Sandhaus, 2008), CNN and Dailymail (CNNDM) (Hermann et al., 2015). We use the

script from See et al. (2017) to extract summaries from raw data, and 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 provides statistics of the datasets. The edges in $\mathcal{G}_C$ are undirected, while those 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 un-anonymized version as in previous summarization work. NYT is licensed by LDC[6]. Following previous work (Zhang et al., 2019; Xu and Durrett, 2019), we use 137,778, 17,222 and 17,223 samples for training, validation, and test, respectively.

4.2 State-of-the-art Baselines

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

Extractive Models: **BanditSum** treats extractive summarization as a contextual bandit problem, 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).

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 by pruning constituency parsing tree.

BERT-based Models: BERT-based models have achieved significant improvement on CNNDM and NYT, when compared with LSTM counterparts. **BertSum** is the first BERT-based extractive summarization model (Liu and Lapata, 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 et al., 2019).

4.3 Implementation Details

We use AllenNLP (Gardner et al., 2018) as the code framework. The implementation of graph encoding is based on DGL (Wang et al., 2019). Experiments are conducted on a single NVIDIA P100 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 pre-trained 'bert-base-uncased' model and fine tune it 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-2' 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 et al., 2018)[7]. 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)[8].

The dependencies among EDUs are crucial to the grammaticality of selected EDUs. Here are the 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

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 et al., 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 and Lapata, 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.

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

[7] https://github.com/PKU-TANGENT/NeuralEDUSeg

[8] https://github.com/jiyfeng/DPLP

head is itself. We determine the head node(s) of non-terminal nodes based on their nuclearity.[9] For example, in Figure 2, the heads of text spans [1-5], [2-5], [3-5] and [4-5] need to be grounded to a single EDU. We propose a simple yet effective schema to convert RST discourse tree to a dependency-based discourse tree.[10] We always consider the dependency restriction such as the reliance of Satellite on Nucleus, when we create oracle during preprocessing and when the model makes the prediction. For the example in Figure 2, if the model selects "[5] being carried ... Liberia." as a candidate span, we will enforce the model to select "[3] and shows ... 8," and "[2] This ... series," as well.

The number of chosen EDUs depends on the average length of the reference summaries, dependencies across EDUs as mentioned above, and the length of the existing content. The optimal average number of EDUs selected is tuned on the development set.

4.4 Experimental Results

Results on CNNDM Table 2 shows results on CNNDM. The first section includes Lead3 baseline, sentence-based oracle, and 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 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 helpful than the Coreference Graph, while the combination of both achieves better performance overall.

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 et al., 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.

Results on NYT Results 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 minimal 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.

4.5 Grammaticality

Due to segmentation and partial selection of sentence, the output of our model might not be as grammatical as the original sentence. We manually examined and automatically evaluated 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.

Automatic Grammar Checking We followed Xu and Durrett (2019) to perform automatic grammar checking using Grammarly. Table 4 shows the grammar checking results, where the average number of errors in every 10,000 characters on CNNDM and NYT datasets is reported. We compare DISCOBERT with sentence-based BERT model. 'All' shows the summation of the number of errors in all categories. As shown in the table, the

[9] 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.

[10] 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.

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 with Grammarly on CNNDM and NYT. 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 results. We ask Turkers to grade the overall preference, coherence and grammaticality from 1 to 5. Mean values along with standard deviations are reported.

summaries generated by our model have retained the quality of the original text.

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

Examples & Analysis We show some examples of model output in Table 6. We notice that a decent amount of irrelevant details are removed from the extracted summary.

Despite the success, we further conducted error analysis and found that the errors mostly originated from the RST dependency resolution and the upstream parsing error of the discourse parser. The misclassification of RST dependencies and the hand-crafted rules for dependency resolution hurted the grammaticality and coherence of the 'generated' outputs. Common punctuation issues include extra or missing commas, as well as missing quotation marks. Some of the coherence issue

Clare Hines ~~, who lives in Brisbane,~~ was diagnosed with a brain tumour after suffering epileptic seizures. ~~After a number of tests doctors discovered~~ she had a benign tumour ~~that had wrapped itself around her acoustic, facial and balance nerve and told her she had have it surgically removed or she risked the tumour turning malignant.~~ One week before brain surgery she found out she was pregnant.
Jordan Henderson~~, in action against Aston Villa at Wembley on Sunday,~~ has agreed a new Liverpool deal. The club's vice captain puts pen to paper on a deal which will keep him at Liverpool until 2020. Rodgers will consider Henderson for the role of club captain ~~after Steven Gerrard moves to LA Galaxy at the end of the campaign but, for now, the England international is delighted to have agreed terms on a contract that will take him through the peak years of his career~~.

Table 6: Example outputs from CNNDM by DISCOBERT. Strikethrough indicates discarded EDUs.

originates from missing or improper or missing anaphora resolution. In this example "['Johnny is believed to have drowned,]$_1$ [but actually *he* is fine,']$_2$ [the police say.]$_3$", only selecting the second EDU yields a sentence "actually he is fine", which is not clear who is 'he' mentioned here.

5 Related Work

Neural Extractive Summarization Neural networks have been widely used in extractive summarization. Various decoding approaches, including ranking (Narayan et al., 2018), index prediction (Zhou et al., 2018) and sequential labelling (Nallapati et al., 2017; Zhang et al., 2018; Dong et al., 2018), have been applied to content selection. Our model uses a similar configuration to encode the document with BERT as Liu and Lapata (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.

Discourse & Summarization The use of discourse theory for text summarization has been explored before. Louis et al. (2010) examined the

benefit of graph structure provided by discourse relations for text summarization. Hirao et al. (2013); Yoshida et al. (2014) formulated the summarization problem as the trimming of the document discourse tree. Durrett et al. (2016) presented a system of sentence extraction and compression with ILP methods using discourse structure. Li et al. (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 the selection basis.

Graph-based Summarization Graph approach has been explored in text summarization over decades. LexRank introduced 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 et al. (2017) also proposed graph-based attention in abstractive summarization model. Fernandes et al. (2018) developed a framework to reason long-distance relationships for text summarization.

6 Conclusion

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

Acknowledgement

Thanks to Junyi Jessy Li, Greg Durrett, Yen-Chun Chen, and to the other members of the Microsoft Dynamics 365 AI Research team for the proofreading, feedback and suggestions.

References

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the Original: Fact Aware Neural Abstractive Summarization. In *AAAI Conference on Artificial Intelligence*.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1662–1675, New Orleans, Louisiana. Association for Computational Linguistics.

Yen-Chun Chen and Mohit Bansal. 2018. Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686. Association for Computational Linguistics.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402, Florence, Italy. Association for Computational Linguistics.

Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. 2018. BanditSum: Extractive Summarization as a Contextual Bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3739–3748. Association for Computational Linguistics.

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-Based Single-Document Summarization with Compression and Anaphoricity Constraints. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1998–2008. Association for Computational Linguistics.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text

summarization. *Journal of artificial intelligence research*, 22:457–479.

Patrick Fernandes, Miltiadis Allamanis, and Marc Brockschmidt. 2018. Structured neural summarization. *arXiv preprint arXiv:1811.01824*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-Up Abstractive Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc.

Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. 2013. Single-document summarization as a tree knapsack problem. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA.

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)*, pages 13–24, Baltimore, Maryland. Association for Computational Linguistics.

Chris Kedzie, Kathleen McKeown, and Hal Daume III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of ICLR*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Junyi Jessy Li, Kapil Thadani, and Amanda Stent. 2016. The role of discourse units in near-extractive summarization. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 137–147. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3728–3738, Hong Kong, China. Association for Computational Linguistics.

Annie Louis, Aravind Joshi, and Ani Nenkova. 2010. Discourse indicators for content selection in summarization. In *Proceedings of the SIGDIAL 2010 Conference*, pages 147–156, Tokyo, Japan. Association for Computational Linguistics.

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.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60. Association for Computational Linguistics.

Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2004. The penn discourse treebank. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC04)*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In *AAAI Conference on Artificial Intelligence*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759. Association for Computational Linguistics.

Ani Nenkova, Kathleen McKeown, et al. 2011. Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2–3):103–233.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389. Association for Computational Linguistics.

Evan Sandhaus. 2008. The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083. Association for Computational Linguistics.

Eva Sharma, Luyang Huang, Zhe Hu, and Lu Wang. 2019. An entity-driven framework for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3271–3282.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1181, Vancouver, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, Ziyue Huang, Qipeng Guo, Hao Zhang, Haibin Lin, Junbo Zhao, Jinyang Li, Alexander J Smola, and Zheng Zhang. 2019. Deep graph library: Towards efficient and scalable deep learning on graphs. *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Yizhong Wang, Sujian Li, and Jingfeng Yang. 2018. Toward fast and accurate neural discourse segmentation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 962–967, Brussels, Belgium. Association for Computational Linguistics.

Florian Wolf and Edward Gibson. 2005. Representing discourse coherence: A corpus-based study. *Computational linguistics*, 31(2):249–287.

Jiacheng Xu, Danlu Chen, Xipeng Qiu, and Xuanjing Huang. 2016. Cached long short-term memory neural networks for document-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1660–1669, Austin, Texas. Association for Computational Linguistics.

Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Hong Kong, China. Association for Computational Linguistics.

Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. 2017. Recent advances in document summarization. *Knowledge and Information Systems*, 53(2):297–336.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, Vancouver, Canada. Association for Computational Linguistics.

Yasuhisa Yoshida, Jun Suzuki, Tsutomu Hirao, and Masaaki Nagata. 2014. Dependency-based discourse parser for single-document summarization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1834–1839, Doha, Qatar. Association for Computational Linguistics.

Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. 2018. Neural Latent Extractive Document Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 779–784. Association for Computational Linguistics.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058, Florence, Italy. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural Document Summarization by Jointly Learning to Score and Select Sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663. Association for Computational Linguistics.

A Appendix

Figure 4 provides three sets of examples of the constructed graphs from CNNDM. Specifically, $\mathcal{G}_C$ is strictly symmetric and self-loop is added to all the nodes to prevent the graph from growing too sparse. On the other hand, all of the on-diagonal entries in $\mathcal{G}_R$ are zero because the node from RST graph never points to itself.

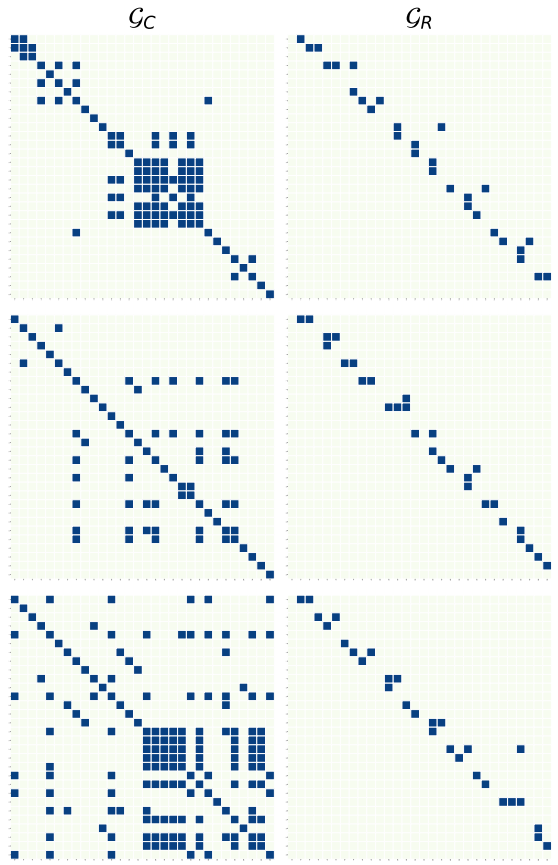

Figure 4: Examples of the adjacent matrix of Coreference Graphs $\mathcal{G}_C$ and RST Graphs $\mathcal{G}_R$.