Extractive Elementary Discourse Units for Improving Abstractive Summarization

Ye Xiong
s2010187@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

Teeradaj Racharak
racharak@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

Minh Le Nguyen
nguyenml@jaist.ac.jp
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa, Japan

ABSTRACT

Abstractive summarization focuses on generating concise and fluent text from an original document while maintaining the original intent and containing the new words that do not appear in the original document. Recent studies point out that rewriting extractive summaries help improve the performance with a more concise and comprehensible output summary, which uses a sentence as a textual unit. However, a single document sentence normally cannot supply sufficient information. In this paper, we apply *elementary discourse unit* (EDU) as textual unit of content selection. In order to utilize EDU for generating a high quality summary, we propose a novel summarization model that first designs an EDU selector to choose salient content. Then, the generator model rewrites the selected EDUs as the final summary. To determine the relevancy of each EDU on the entire document, we choose to apply group tag embedding, which can establish the connection between summary sentences and relevant EDUs, so that our generator does not only focus on selected EDUs, but also ingest the entire original document. Extensive experiments on the *CNN/Daily Mail* dataset have demonstrated the effectiveness of our model.

CCS CONCEPTS

• **Applied computing** → **Document management and text processing**; • **Information systems** → **Summarization**.

KEYWORDS

Abstractive summarization, text generation, two-stage summarization

ACM Reference Format:
Ye Xiong, Teeradaj Racharak, and Minh Le Nguyen. 2022. Extractive Elementary Discourse Units for Improving Abstractive Summarization. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3477495.3531916

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531916

1 INTRODUCTION

Summarization is the task of condensing a document's main points into a shorter document [2, 15, 17]. There are two broad approaches to summarization: *extractive* and *abstractive*. Extractive approaches congregate summary sentences merely from text segments taken straightly from the input text (original text) [4, 23], while abstractive approaches focus on generating novel word and phrases not appeared in the original text [16–18]. Recently, the popular and effective paradigm [4, 9, 11] are to design a two-step pipeline to first select (or extract) and then rewrite (or compress) each pre-extracted sentence. Existing results [6] show that sentences extracted from the original document are prone to contain irrelevant and redundant phrases, and such rewriting systems effectively reduce redundancy, and improve the conciseness and comprehensible.

However, a single sentence in document cannot accommodate enough information to express a summary sentence. Recent studies [9] demonstrate that a high percentage of summary sentences are made up by not only one document sentence; and also, composing a summary through only compressing sentences can cause performance degradation. Concurrently, the pre-extracted sentences usually provide some duplicate information, or trivial details and expresses a comparatively independent meaning, causing difficulty for rewriting multiple sentences into an individual summary sentence, and the degradation of conciseness in the final summary. Besides, rewrtier only rewrites (or compresses) pre-selected content that is likely to lose contextual information in the original document, so that the final summary may lose some information.

To alleviate the aforementioned problems, in this paper, we propose an EDU-based extractor-rewriter summarization paradigm bulit upon BERT [5]. To reduce redundancy and concurrently concentrate on critical information in the document, we choose to use Elementary Discourse Unit, which is a sub-sentence phrase unit proposed from Rhetorical Structure Theory (RST) [14] as the minimal selection unit (instead of a sentence) for our extractor (selector) model. We automatically obtain EDUs based on the BiLSTM CRT framework [20] which is a discourse segmenter, achieved a high F1 score of 94.3 on the RST-DT test set.

Extractor aims to extract informative EDUs. We choose to encode the whole document with BERT and fine tune the BERT model for critical EDU selection. Specifically, we obtain the representation of each token after BERT encoding; then we adopt a Self-Attentive Span Extractor [10] to learn EDU representation. Finally, we can get the extract probability of each EDUs based on the standard transformer encoder. During rewriting, there are also some extra information in original document, which can promote surmise faithful details. We use the entire input document as a context for

rewriting extracted EDUs. To connect this hybrid system (extractor and rewriter), we utilise the group tag embedding [1], which can apprise rewriter of the current critical part (EDUs) being rewritten through grouping the summary sentence and its highly correlated EDUs into one group. Intuitively, rewriter can write a summary based on some primary and critical parts like human being, and the role of the tag embedding is to mark the critical parts and prompt it to the rewriter.

We summarize our main contributions as follows. (1) We propose a two-step paradigm (i.e., first select and then rewrite) for abstractive summarization. Our model consists of EDU-extractor and EDU-rewriter that rewrite the primary and critical information from the entire document meanwhile it uses the entire document as background knowledge to generate informative and abstractive summary with high quality. (2) We utilize the group tag embedding as a connection between the extractor and the rewriter, enabling that the rewriter does not only focus on the output of extractor to rewrite, but also ingest the entire original document to enhance informativeness and abstraction. (3) Our extensive experiments on *CNN/Daily Mail* dataset demonstrate that the proposed method outperforms the strong competitors [11, 13], and is even comparable with the extractive summarization method [13, 22], in terms of both automatic metrics and the human evaluation. The results also show that our method is an effective and practicable way to take advantage of EDUs in the abstractive summarization task.

2 PROPOSED MODEL

We propose a two-step model for abstractive summarization. Our model is inspired by the existing extract-rewrite systems introduced by [3, 11]. First, an extractive summarization model is used to select central content in original document for rewriting. In this work, we consider using **E**lementary **D**iscourse **U**nits (**EDU**) as the selection unit which is different from most of previous models that use sentence. Second, the selected EDUs are emphasized by adding group tag embedding in document. Then, the processed document after emphasizing is used in the rewriter as an input to generate the final summary.

To define the problem, we denote $\mathbf{D} = \{w_i\}_{i=1}^{|\mathbf{D}|}$ to represent document $\mathbf{D}$, which contains $|\mathbf{D}|$ tokens, and $\mathbf{S} = \{w_j\}_{j=1}^{|\mathbf{S}|}$ to represent the final summary $\mathbf{S}$, which contains $|\mathbf{S}|$ tokens.

2.1 EDU-Extractor

Follow Liu and Lapata [13], we apply BERT for document encoding. Specifically, we insert $\langle CLS \rangle$ and $\langle SEP \rangle$ tokens at the beginning and the end of each sentence, respectively. In order to better preserve contextual content and adapt to long articles such as news articles, the maximum sequence length in BERT is extended form 512 to 768 in our expriments.

We assume that $\mathbf{D} = \{\mathrm{E}_1, \mathrm{E}_2, \ldots, \mathrm{E}_n\}$ and $\mathrm{E}_i = \{w_{i1}, w_{i2}, \ldots, w_{i\ell_i}\}$ represent the input document $\mathbf{D}$ after segmenting and the i-th Elementary Discourse Unit, respectively, where n is the number of EDUs in a document and ℓ_i is the number of BPE tokens in the i-th EDU. Then, the BERT model is used to encode the document:

$$\{\mathbf{R}_{11}, \ldots, \mathbf{R}_{n\ell_n}\} = \mathbf{BERT}(\{w_{11}, \ldots, w_{n\ell_n}\}) \tag{1}$$

where $\{\mathbf{R}_{11}, \ldots, \mathbf{R}_{n\ell_n}\}$ denotes the BERT output of the whole document in the same length as the input. Note that, if a selected EDU is the first EDU in a sentence, we need to prepend the $\langle CLS \rangle$ to it. Furthermore, if a selected EDU is the last EDU in a sentence, we also need to append the $\langle SEP \rangle$ to it.

In Liu and Lapata's work, the representation of the $\langle CLS \rangle$ token is used as sentence representation. In contrast, we do not use this setting in this work since we need to obtain the representation of each EDUs in document. Hence, we adopt a Self-Attentive Span Extractor (**SpanExt**), proposed by Lee [10], to learn EDU representaion.

$$\mathbf{R}_i = \mathbf{SpanExt}(\{\mathbf{R}_{i1}, \ldots, \mathbf{R}_{i\ell_i}\}) \tag{2}$$

Here, for the i-th EDU with ℓ_i words, the output from the BERT encoder $\{\mathbf{R}_{i1}, \ldots, \mathbf{R}_{i\ell_i}\}$, each EDU representaion for Eq. 2 can be computed as follows:

$$\alpha_{ij} = \mathbf{W}_2 \cdot ReLU(\mathbf{W}_1 \mathbf{h}_{ij} + \mathbf{b}_1) + \mathbf{b}_2 \tag{3}$$

$$\mathbf{a}_{ij} = \frac{\exp(\alpha_{ij})}{\sum_{k=1}^{\ell_i} \exp(\alpha_{ij})} \tag{4}$$

$$\mathbf{R}_i = \sum_{j=1}^{\ell_i} \mathbf{a}_{ij} \cdot \mathbf{R}_{ij} \tag{5}$$

where α_{ij} is the score of the j-th word in the EDU. This score represents how important the word is to the entire EDU, $\mathbf{a}_{ij}$ is the normalized attention score of the j-th word w.r.t. all the words in EDU, and $\mathbf{R}_i$ is the EDU representation obtained by weighting the sum of the BERT output hidden states. All $\mathbf{W}$ and $\mathbf{b}$ are learning parameters.

After obtaining the representation of EDUs, the whole document is represented as a sequence of EDU representations: $\mathbf{R}_D = \{\mathbf{R}_1, \ldots, \mathbf{R}_n\}$, which will be sent to the EDU selector.

To develop our EDU selector, we use Transformer Encoder [19] (denoted by **TransEnc**) to convert each EDU representation $\mathbf{R}_i$ into the final contextual EDU representation $\mathbf{R}_i^{\mathbf{final}}$, which will be sent to the final output sigmoid classifier that calculates the extraction probability on each EDU according to $\mathbf{R}_i^{\mathbf{final}}$:

$$\mathbf{R}_i^{\mathbf{final}} = \mathbf{TransEnc}(\mathbf{R}_i) \tag{6}$$

$$P(\mathbf{EDU}_i \mid \mathbf{D}) = \sigma(\mathbf{W} \cdot \mathbf{R}_i^{\mathbf{final}} + \mathbf{b}) \tag{7}$$

where $P(\mathbf{EDU}_i \mid \mathbf{D})$ represents the probability that i-th EDU is selected, and $\mathbf{W}$ and $\mathbf{b}$ are trainable paremeters. After knowing selection probabilities of each EDU in the document, we select the five most probable EDUs as the extractor output. Afterwards, the entire document is sent to the summary generator, where the selected EDUs will be emphasized.

2.2 EDU-Based-Rewriter

As shown in Figure 1, the EDU-Based-Rewriter is a summary generator that uses a standard Transformer sequence to sequence model with BERT as the encoder. We expect that generator does not only focus on the selected content (i.e., the EDUs in this work), but also does not lose the rest of text information around. For this purpose, we employ the tag embedding setting [1].

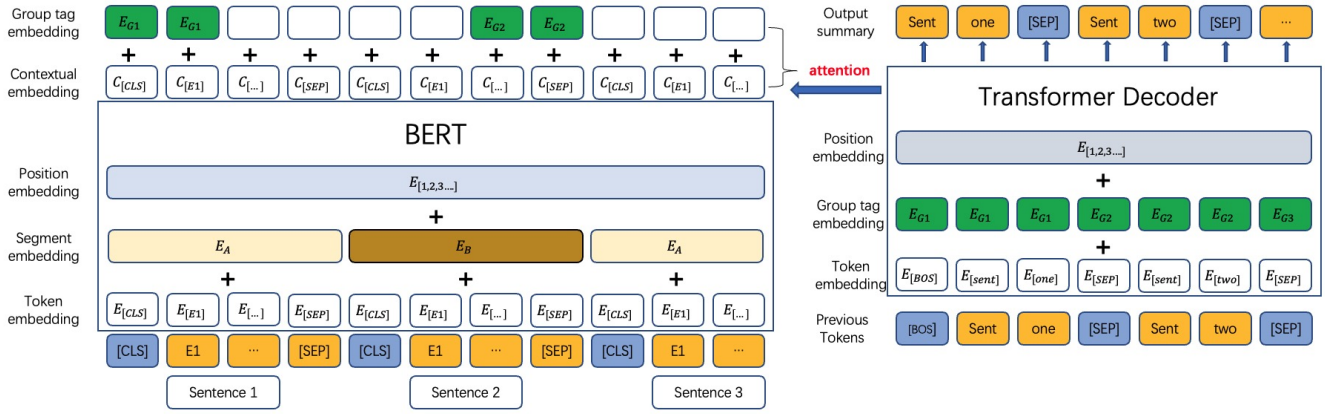

Figure 1: Architecture of the EDU rewriter. The group tag embedding build connection between encoder and decoder.

Specifically, we match the extracted summary with the original entire document using a specific method described in the next section. Each summary sentence corresponds to either single or multiple EDUs. Each sentence and its corresponding EDU will constitute a group and be marked with a tag in the ascending order. For instance, the first summary sentence and its corresponding EDU are marked with tag one, the second summary sentence and its corresponding EDU with tag two, and so on. We define the tag embedding as follows:

For group:

$$G = \{G_k\}_{k=1}^{|G|} \tag{8}$$

where $|G|$ indicates how many groups there are. In this work, it is the same as the number of summary sentences.

For tags:

$$G_D = \{g_i = k \textbf{ if } w_i \in G_k \textbf{ else } 0\}_{i=1}^{|D|} \tag{9}$$

$$G_S = \{g_j = k \textbf{ if } w_j \in G_k \textbf{ else } 0\}_{j=1}^{|S|} \tag{10}$$

where $|D|$ and $|S|$ are the number of tokens in a source document and the summary, respectively. In order to emphasize on the selected content, the embedding of the group tags is shared by the encoder and the decoder. After that, the summary sentence obtains the same tag embedding as its correlative EDUs, so that the k-th summary sentence can correspond to its highly correlated EDUs during decoding.

The original document is sent to the BERT encoder as the input, we also insert $\langle CLS \rangle$ and $\langle SEP \rangle$ tokens at the beginning and the end of each sentence, respectively. After the BERT encoder processing, we obtain the final representation of each token by adding their corresponding tag embedding (**TAGEmb**), as shown in Eq. 11. Note that the detail of the tag embedding is described in Section 4.

$$\mathbf{R}_{D+Tag} = \textbf{BERT}(D) + \textbf{TAGEmb}(G_D) \tag{11}$$

The decoder of EDU-Based-Rewriter (denoted by **TransDec**) follows a standard Transformer decoder architecture [13, 19]. Specifically, special tokens $\langle BOS \rangle$ and $\langle EOS \rangle$ are placed at the beginning and the end of final summary sentences, respectively; and also, token $\langle SEP \rangle$ is used between sentences as usual.

For each decoder beam search step, the group tag is generated, starting with 1 after the special token $\langle BOS \rangle$ and increasing by one after each special token $\langle SEP \rangle$. The decoder's input at each step needs to add the tag embedding (**TAGEmb**), as shown in Eq. 12. We describe the experiments of tag embedding in Section 4.

$$\mathbf{R}_{S+Tag} = \textbf{EMB}(S) + \textbf{TAGEmb}(G_S) \tag{12}$$

$$P(w_j \mid w_{<j}, D, G_D) = \textbf{TransDec}(\mathbf{R}_{S+Tag}^{(<j)}, \mathbf{R}_{D+Tag}) \tag{13}$$

where $\mathbf{R}_{D+Tag}$ is encoder output after adding tag embedding, used as the memory keys and values for multi-head attention. The transformer decoder predicts tokens' probability at position j based on the tagged token embedding before j and the encoder output $\mathbf{R}_{D+Tag}$, as defined in Eq. 13.

3 TRAINING

We train our EDU-extractor and EDU-rewriter separately on a preprocessed dateset labeled with gold standard extractions. To obatin gold-standard EDU extraction, we match each sentence in human summary to each EDU in the original document. In contrast with some empirical methods [1] which use the average recall of ROUGE-1/2/L score, we greedily pick up EDUs until a ROUGE-1 score drops, then, we obtain the best match (a set of EDUs) for each summary sentence. Hence, each summary corresponds to multiple or singe EDU.

After obtaining a gold-standard EDU extraction set, for our EDU-extractor, each gold-standard EDU extraction is labeled 1 and the rest of EDUs in document is labeled 0. Then, we train the model with a binary cross-entropy loss function. For the EDU-rewriter, we convert gold-standard extraction set into group tags using Eq. 9 for each summary sentence and its associated gold-standard extraction belonged to each group, and train the model with a negative log-likelihood loss.

4 EXPERIMENTAL SETUP

We evaluate our model on the non-anonymized version of *CNN/Daily Mail* dataset [7], and follow the standard splitting [17]. The dataset contains 287,226 training pairs, 13,368 validation pairs, and 11,490

test pairs. To segment the documents into EDUs, we use the BiLSTM CRF framework which achieves 94.3 F1 score in EDU-segmentation. To evaluate summarization performance, we use ROUGE metric (R-1,R-2 and R-L) [12].

For our extractor, the EDU-selector is set to 2 layers transformer encoder with an embedding size of 768 being randomly initialized. The optimizer is Adam [8] with $\beta 1 = 0.9$ and $\beta 2 = 0.999$, and total training steps of 50,000 including 10,000 warm up steps [13].

For our rewriter, we initialize the 6 layers transformer decoder randomly with an embedding size of 768, and tie input-output embedding for implementing group tag embedding. The optimizer is the same as the extractor's, and total training steps of 220,000 including 10,000 steps for warm up [13]. Both extractor and rewriter are initialized with pre-trained uncased BERT-base.

5 EXPERIMENTAL RESULTS

Main results are shown in Table 1. We do the comparison with several state-of-the-art on extractive and abstractive summarization methods. Lead-3 uses the first three sentences of the article as its summary according to characteristics of news articles. DISCOBERT [22] is an EDU-based extractive method which selects critical EDUs and utilizes a dependency-based discourse tree to keep the grammaticality of the final summary. BERTSUMEXT [13] is a strong extractive model that uses a linear classifier upon BERT and represents sentence embeddings by introducing special tokens to begin each sentence. Four abstractive methods for comparison include: Fast-Abs [3], a sentence-based extract and rewrite system by using reinforcement learning as connection. BERTSUMEXTABS [13], which rewrites the output of BERTSUMEXT as abstractive summary. BERT+C/W[21], which uses a copy-and-rewriter mechanism to choose sentences for copying or rewriting. EDUSum [11] is a hybrid summarization system that includes an EDU selection module and a standard pointer generator based EDU fusion module, which adopts reinforcement learning to leverage.

As we can see in Table 1, our EDU-extractor (EDUEXT) compared to strong extractive methods (i.e., DISCOBERT and BERTSUMEXT), it is seen that EDUEXT achieves lower results due to the differences in extraction goal, which can just find what is important and ignores the grammar and continuity of the sentence. Nevertheless, after adding EDU-Rewriter to EDUEXT, compared to BERTSUMEXTABS and BERT+C/W which are both similar to EDUEXT+RW in the model architecture, it is seen that EDUEXT+RW achieves better performance with regard to three ROUGE metrics, proving that EDU is more appropriate than sentence to be a basic selection in hybrid (extract and rewrite) method for abstractive summarization. Compared with EDUSum, it demonstrates that our model can make good use of EDU for abstractive summarization.

Compared EDUEXT+RW to EDUEXT, EDUEXT-RW achieves a great improvement in terms of ROUGE-1/2/L score. This shows that EDU-rewriter effectively improves summary quality. To observe the validity of the entire document as an input, we apply EDUEXT to BERTAbs which is the same as the decoder of BERTSUMEXTABS, the resulting gap between EDUEXT+RW and EDUEXT+BERTAbs shows the effectiveness and necessity of using entire document as input since adding background knowledge for rewriting EDUs. Compared with Oracle+RW, our model still has a big gap, perhaps because the pre-selected EDUs do not fully represent the central idea of the entire article, leading to the wrong rewriter direction for rewriter and generating fake information. We need further improvement on EDU selector which will be our future work.

Table 1: Model comparison on CNN/Dalily Mail.

Model	R-1	R-2	R-L
extractive			
Lead-3	40.34	17.70	36.57
DISCOBERT	43.38	20.44	40.21
BERTSUMEXT	43.25	20.24	39.63
abstractive			
Fast-abs	40.88	17.80	38.54
BERT+C/W	42.92	19.43	39.35
BERTSUMEXTABS	42.13	19.60	39.18
EDUSum	41.40	18.03	38.79
EDUEXT	39.28	18.80	37.08
EDUEXT+BERTAbs	42.11	19.69	39.74
EDUEXT+RW	**43.09**	**20.24**	**40.52**
Oracle+RW	54.49	31.76	51.79

Table 2: Test performance with different numbers of candidate EDUs in EDUEXT+RW

Number of candidate EDUs	R-1	R-2	R-3
3	42.67	20.01	40.09
5	**43.09**	**20.24**	**40.52**
6	42.39	20.09	39.88
8	40.83	19.52	38.64

Table 3: Example of a generated summary

Gold summary:twenty years ago, on april 19, 1995, timothy mcveigh set off a massive bomb in oklahoma city,deborah lauter and mark pitcavage: right-wing extremism should still be taken seriously.
Summary generated by ours: 20 years ago , 168 men , women and children were killed in oklahoma city bombing, peter bergen : the bombing exposed the danger of the extreme right, he says the bombing remains the deadliest act of domestic terrorism in u.s. history, bennett : the u.s. citizens targeted their own government with a deadliness

Furthermore, we design experiments on the effect of the number of pre-selected EDUs (candidate EDUs). As Table 2 shows, it performs best when the number of candidate EUDs is five. This illustrates that redundant and sparse content produced by excessive pre-selected content can confuse the rewriter. In order to more intuitively reflect the performance of the model. We randomly choose one generated summary(due to the limited space) by our model from test data for observation. As shown in Table 3, we observe that our model captures the salient information in the source article and generates a more specific and informative summary.

6 CONCLUSIONS

In this paper, we propose a novel EDU-based extract-rewrite abstractive summarization model that improves the informativeness and conciseness of the summary. In our model, EDU-selector is designed to extract critical EDUs, and EDU-rewriter rewrites the

selected EDUs based on the entire document. The group tag embedding builds connection between these two modules. Experimental results have demonstrated that our method outperforms the compared methods, and are able to conclude that: (1) EDU is an appropriate selection unit for summarization and (2) our model can make fully good use of EDU for summarizition.

ACKNOWLEDGMENTS

This work was supported partly by JSPS Kakenhi Grant Number 20H04295. The authors would like to thank helpful comments from reviewers.

REFERENCES

[1] Guangsheng Bao and Yue Zhang. 2021. Contextualized Rewriting for Text Summarization. *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)* (2021).

[2] Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, , and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. *arXiv preprint arXiv:1803.10357.* (2018).

[3] 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.* (2018), 675–686.

[4] 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* . (2016), 484–494.

[5] 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.* (2019), 4171–4186.

[6] 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.* (2018), 4098–4109.

[7] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems* 28 (2015), 1693–1701.

[8] Diederik P. Kingma and Jimmy Ba. 2015. Adam:A method for stochastic optimization. *CoRR,abs/1412.6980.* (2015).

[9] Logan Lebanoff, Kaiqiang Song, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. Scoring sentence singletons and pairs for abstractive summarization. *arXiv preprint arXiv:1906.00077.*. (2019).

[10] 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.* (2017), 188–197.

[11] Zhenwen Li, Wenhao Wu, and Sujian Li. 2020. Composing Elementary Discourse Units in Abstractive Summarization. *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* . (2020), 6191–6196.

[12] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *In Proc. ACL workshop on Text Summarization Branches Out.* (2004).

[13] 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).* (2019), 3728–3738.

[14] 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* (1988).

[15] Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304.* (2017).

[16] 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.* (2015), 379–389.

[17] Abigail See, Peter J.Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointergenerator networks. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (2017), 1073–1083.

[18] 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 (EMNLPIJCNLP).* (2019), 3271–3282.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[20] 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* (2018), 962–967.

[21] Liqiang Xiao, Lu Wang, Hao He, and Yaohui Jin. 2020. Copy or rewrite: Hybrid summarization with hierarchical reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9306–9313.

[22] Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-Aware Neural Extractive Text Summarization. *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* (2020).

[23] 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.* (2018), 654–663.