Abstractive Text Summarization Based on Neural Fusion

Yllias Chali, Wenzhao Zhu
University of Lethbridge
Lethbridge, Alberta
Canada

Abstract

Abstractive text summarization, in comparison to extractive text summarization, offers the potential to generate more accurate summaries. In our work, we present a stage-wise abstractive text summarization model that incorporates Elementary Discourse Unit (EDU) segmentation, EDU selection, and EDU fusion. We first segment the articles into a fine-grained form, EDUs, and build a Rhetorical Structure Theory (RST) graph for each article in order to represent the dependencies among EDUs. Those EDUs are encoded in a Graph Attention Networks (GATs), and those with higher importance will be selected as candidates to be fused. The fusing stage is done by BART which merges the selected EDUs into summaries. Our model outperforms the baseline of BART (large) on the CNN/Daily Mail dataset, showing its effectiveness in abstractive text summarization.

Introduction

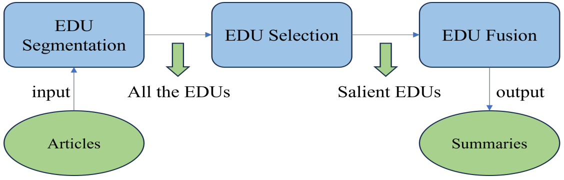
Text summarization can be broadly categorized into two types: extractive and abstractive, depending on whether they reproduce content directly from the source text or produce novel content. The abstractive summarization approaches exhibit reduced reliance on copied content, leading to the generation of summaries that bear closer resemblance to human-annotated ones, so it is more likely to establish a coherent context.

Nevertheless, a majority of the existing methodologies either treat the entire text sequences (See, Liu, and Manning, 2017; Lewis et al., 2020) or important sentences (Chen and Bansal, 2018) as inputs, introducing excessive noise during the summarization process. In order to solve the problem of the unexpected noise that the input sequences may bring to the predicted abstracts, we propose employing fine-grained units, EDUs, as the foundamental units to generate summaries, and we only take the concatenations of those informative EDUs as the inputs to be fused to avoid most of the redundant content. Therefore, how to truncate the whole text sequences into fine-grained units and selecting those important ones are the two necessary steps before starting to fuse them.

Motivated by the problems we come across above, our proposed work can be generally summarized in the following three stages (Figure 1):

1. Segmenting the original long text sequences into fine-grained units and preserving the dependencies among those units in order to be selected,
2. Selecting the informative units as the candidate units to be fused according to the semantic relations among all the discourse units,
3. Fusing the selected units by taking their concatenations as the inputs of the fusing model to ensure the coherence of the final outputs in an abstractive way.

Figure 1: The three stages of our work



We propose to use this procedure to maximize the semantic coherence of the summaries while ensuring a high density of information on the abstract content. Specifically, our objective is to enhance the summarization accuracy of the comprehensive model applied to the source text by identifying an appropriate method for EDU selection and subsequent EDU fusion. We intend to exceed the BART (large) baseline (Devlin et al., 2019) by using the method above.

Due to the copyrights and the qualities of some of the text summarization datasets such as New York Times (NYT) (Consortium and Company, 2008) and XSum (Narayan, Cohen, and Lapata, 2018), our experiments are mostly conducted on CNN/Daily Mail (Hermann et al., 2015) dataset. Also, the time and the resource investment required for the creation of a reliable, high-quality dataset further exceed our ability, Additionally, our work only focuses on short documents (sequences that are less than 2048 tokens) since long document processing requires higher hardware capacity.

To transform the original articles into groups of EDUs, we first followed the guidelines provided by Fairseq (Ott et al., 2019) to convert the raw text files into formatted documents, facilitating subsequent processing steps. We then deployed the segmentation tool proposed by Wang, Li, and Yang (2018) to further convert the preprocessed documents of articles into EDU lists. In the selecting stage, we first used the method provided by Ji and Eisenstein (2014) to establish the graphs that illustrate the relations between every two EDUs or EDU spans. Then, we designed a graph neural network-based (Veličković et al., 2017) model, taking the graphs as inputs to classify the EDUs as 0 (unimportant) and 1 (important). Finally, we used BART (Lewis et al., 2020) to fuse those EDUs that are labeled with 1 to increase the coherence of the outputs. The details of the implementations will be elaborated in the following sections.

Essentially, our method optimizes the data for fusing. Specifically, it converts the original long sequences into lists of informative units, greatly reducing the hardware requirements of the training device, and saving the memory expenses of training. Also, the performance of the fusing model reaches a higher level by optimizing the datasets. Additionally, our model can be used in other types of text that have close contextualizations, such as academic papers and novel chapters. In this way, our method can help people have a quick and accurate understanding of the text within a certain length.

Our main contribution lies in the utilization of graph attention networks to select important EDUs and fuse them. This approach substantially reduces input redundancy for BART, resulting in an improved overall segmentation-selection-fusion process that outperforms the baseline of BART.

Related Works

RST-based (Mann and Thompson, 1988) datasets are used in many of the works. Recently, Wang, Li, and Yang (2018) employed a neural network-based methodology in conjunction with the RST Discourse Treebank (RST-DT) (Carlson, Marcu, and Okurovsky, 2001) dataset to segment sentences into EDUs, subsequently utilized as inputs for downstream tasks. This method was used in the work by Li, Wu, and Li (2020) as well as our work. Moreover, diverse techniques exist for sentence segmentation, including methods rooted in Text Cohesion Theory (Lebanoff et al., 2020) and phrase division (Li et al., 2017; Bing et al., 2015).

DiscoBERT proposed by Xu et al. (2020a) stands out as a prominent EDU selecting model with BERT (Devlin et al., 2019) as its encoder. It used Discourse Parsing from Linear Projection (DPLP) (Ji and Eisenstein, 2014) as the method to build RST trees for each dataset it was using. Also, it adeptly processed the converted RST graphs and coreference graphs derived from articles, identifying the most coherent concatenation of EDUs, to integrate into summaries. These concatenated sequences then directly assume the role of summaries. Inspired by this method, we propose to use RST graphs as inputs for our Graph Attention Network (Veličković et al., 2017) which serves as the selector in our framework. Also, we employ BERT as the encoder to enhance the extraction of dependencies among EDUs. For the convenience of deploying the model, we leverage the repository proposed by Fey and Lenssen (2019) to build the architecture of our model.

An array of diverse architectures characterizes the state-of-the-art approaches. Prominently, the traditional RNN-based neural network, Pointer Generator Network (PGNet) See, Liu, and Manning (2017) emerges as a favorable choice for sentence fusion, as evidenced by Li, Wu, and Li (2020); Lebanoff et al. (2019); Xu et al. (2020b). The integration of Pointer Network (Ptr-Net) (Vinyals, Fortunato, and Jaitly, 2015) augments the efficacy of the selection phase by facilitating the generation of output completely from input, a strategy notably adopted by Chen and Bansal (2018). Furthermore, transformer-based (Vaswani et al., 2017) models (Lewis et al., 2020; Radford et al., 2018; Devlin et al., 2019), constitute valuable alternatives in instances where neural networks are not employed (Rush, Chopra, and Weston, 2015). Other architectural enhancements are often realized through structural modifications (Li et al., 2017) or the incorporation of supplementary sub-structures (Xu et al., 2020b).

Various mechanisms are harnessed to enhance the model's precision. Notably, Xu et al. (2020b); See, Liu, and Manning (2017); Chen and Bansal (2018) employed the copy mechanism, facilitating not only the generation of novel terms but also the direct replication of words from the source text, effectively addressing the out-of-vocabulary (OOV) challenge. The efficacy of the coverage mechanism (Tu et al., 2016), extensively employed in Neural Machine Translation (NMT) (Luong, Pham, and Manning, 2015; Cohan et al., 2018), extends to the domain of text summarization. Moreover, reinforcement learning leveraged by Li, Wu, and Li (2020); Chen and Bansal (2018), integration of window size (Luong, Pham, and Manning, 2015; Wang, Li, and Yang, 2018; Song et al., 2019) to attenuate noise stemming from distant information, and the adoption of minimum-error rate translation (Rush, Chopra, and Weston, 2015) stand as prevalent optimization strategies. Post-fusion refinement is achievable through the imposition of linguistically-motivated constraints (Li et al., 2017; Bing et al., 2015; Thadani and McKeown, 2013) or the application of Conditional Random Field (CRF) (Wang, Li, and Yang, 2018).

EDU Segmentation

Segmentation Model

The RNN-based model provided by Wang, Li, and Yang (2018) can segment sentences into EDUs with an accuracy of around 94.5% on the RST-DT dataset. Therefore, we consider the method to be relatively highly accurate. This tool leverages a pre-trained word encoder model ELMo (Peters et al., 2018) to counter the limitation of the dataset size.

RST tree building

Discourse Parsing from Linear Projection (DPLP) was proposed by Ji and Eisenstein (2014) to conduct RST parsing and make predictions for relations between each two EDUs and EDU spans, and nuclearities of EDUs. Essentially, DPLP is a shift-reduce method based on RST, if two

EDUs meet the condition for establishing a relation, *reduce* is done, otherwise *shift* is done. When *reduce* is taken, the relation between the top two EDUs or EDU spans in the stack is also determined, along with their nuclearities. Consequently, the built RST trees have both the relations between each two EDUs or EDU spans and the nuclearities of EDUs.

EDU Selection

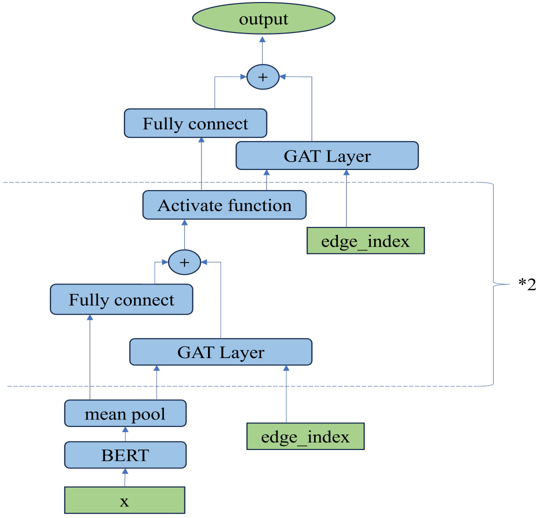
The consideration of nuclearity in an RST tree provides a general indication of the hierarchical importance of different units, but it might not capture the full spectrum of significance or relevance in all contexts. Moreover, nuclearity is just a valuable concept in RST for understanding the general structure and hierarchy of discourse units, selecting important EDUs provides a more refined and task-specific perspective on the significance of individual elements within the discourse. Therefore, selecting EDUs based on their nuclearities and making determinations under global conditions is necessary.

Selection Model

We designed a simple 3-layer GAT model for predicting the labels of EDUs. The task, is therefore a node-level classification task. The first two layers of GAT are for information aggregation and the last GAT layer is designed for normalizing the features into the dimension of the number of classes. To solve the possible vanishing gradient problem, we used Exponential Linear Unit (ELU) Clevert, Unterthiner, and Hochreiter (2016) as the activating function.

Additionally, we leveraged element-wise summation to combine the outputs of different GAT layers with the original inputs, allowing the model to capture various levels of abstraction and information from different sources. Figure 2 shows the overall architecture of our model for EDU selection.

Figure 2: The architecture of EDU selecting model



In all three layers of GAT, we used the multi-head mechanism to reduce the randomness of the predicted results. The first two layers of GAT concatenate the outputs of the 4 heads while the last layer averages the 4 heads and aggregates the information along with the last fully connecting layer and predicts the classes of EDUs. In Figure 2, x indicates the index sequences of EDUs and $edge\_index$ represents the edges that connect EDUs, for example, $(3, 5)$ shows the edge from EDU_3 to EDU_5.

In the training stage, the logit loss function is utilized because our task is essentially a classification task. Specifically, the function can be shown as formula 1, where N indicates the number of nodes in one batch. We finally use the average loss in one batch. $Adam$ algorithm is employed to update the parameters of the model.

$$l_n = -[y_n \log \sigma(x_n) + (1 - y_n) \log(1 - \sigma(x_n))] \quad (1)$$

$$L = \{l_1, ..., l_N\}^T \quad (2)$$

$$l(x, y) = mean(L) \quad (3)$$

The trained model is then used for generating the sequences of labels for the EDUs from all the articles, where only those EDUs that are labeled with '1' will be chosen and concatenated.

Converted RST Graphs

The dataset we used for selecting is the converted RST graphs of CNN/Daily Mail and it was uploaded on the repository [1] by Xu et al. (2020a). The conversion from the originally built RST trees to converted homogeneous graphs can be generally shown in figure 3:

Figure 3: The converting process of an RST tree

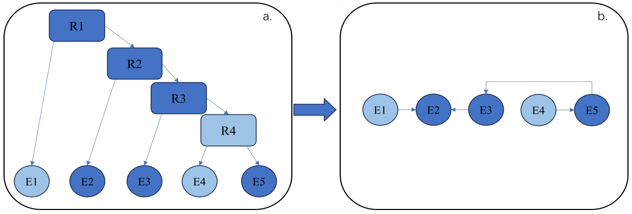
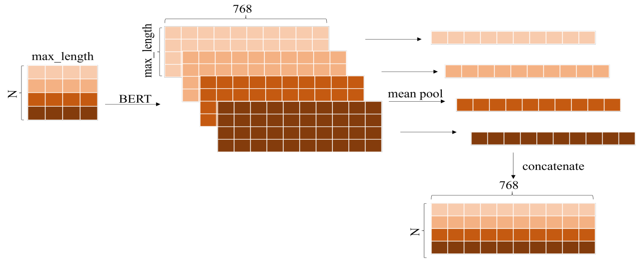


Figure 3.a shows the original RST tree and 3.b shows the converted RST discourse tree, the dark blue nodes are nucleus nodes and the light blue nodes are satellite ones. $R_1 - R_4$ are relations between each two nodes and $E_1 - E_5$ indicate the EDUs. The converting principles can be concluded as two rules: *In terms of the intermediate nodes, (a) if the two children are both nucleus nodes or both satellite nodes, then the right node is pointing at the left node, (b) if there is one satellite node and one nucleus node, then the satellite note is pointing at the nucleus node* Xu et al. (2020a). In this way, the converted RST graphs always have *N* nodes and *N-1* edges.

[1] https://utexas.app.box.com/v/DiscoBERT-ACL2020

EDU encoding We utilized BERT as the encoder to enrich the features of the embeddings. Focusing on aligning the input dimensions of GAT, we convert the 3-dimensional vectors into 2-dimensional ones by using the mean pooling strategy. Finally, we get the articles in the form of 2-dimensional vectors $(N, 768)$, where N is the number of EDUs in an article. The process can be depicted in figure 4.

Figure 4: The overall process of encoding articles



Greedy Method

An alternative approach we utilized in our study is the application of a greedy method. The article-summary pairs are first obtained and segmented as previously shown, followed by the concatenation of 1, 2 and 3 EDUs respectively for each sentence. Finally, the combinations that maximize the $ROUGE - L_{recall}$ scores in terms of the highlights presented will be designated as the important EDUs.

Figure 5: Greedy Method

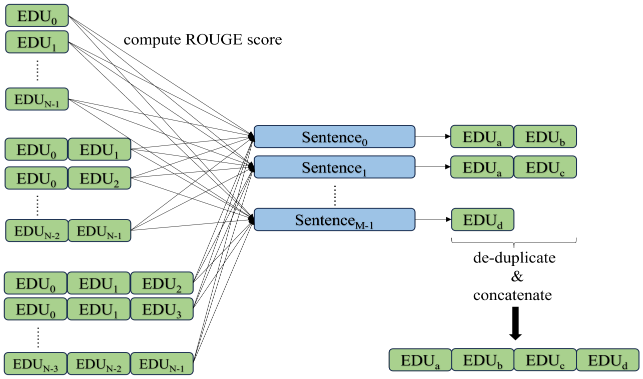

EDU_0
EDU_1
EDU_{N-1}
EDU_0 EDU_1
EDU_0 EDU_2
EDU_{N-2} EDU_{N-1}
EDU_0 EDU_1 EDU_2
EDU_0 EDU_1 EDU_3
EDU_{N-3} EDU_{N-2} EDU_{N-1}
compute ROUGE score
Sentence_0
Sentence_1
Sentence_{M-1}
EDU_a EDU_b
EDU_a EDU_c
EDU_d
de-duplicate
&
concatenate
EDU_a EDU_b EDU_c EDU_d

Figure 5 illustrates the procedure of selecting important EDUs using the greedy method. Within the diagram, N represents the count of EDUs within an article, while M signifies the number of sentences in a referenced summary. The indices a, b, c, and d denote potential candidate EDU positions. The procedure can also be represented using the following equation:

$$EDU_C = \sum_{k=0}^{M-1} \arg\max(ROUGE - L_{recall}(EDU_{comb}, sentence_k)) \quad (4)$$

where EDU_{comb} is the set encompassing all combinations of 1, 2 and 3 EDUs, and EDU_C is the outcome achieved by concatenating the selected EDUs. With the objective of diminishing redundancy within concatenated EDUs, in cases where duplicate EDUs are present in the candidate lists, we eliminate the subsequent occurrences of these EDUs and retain only those EDUs that make their initial appearance in the candidate list.

Also, to counter the problem of the potential empty EDUs, we set up all the ROUGE scores of an empty EDU to zero so those empty EDUs will not be selected in this method.

EDU Fusion

Due to the state-of-the-art performance of BART in natural language processing, it is employed to fuse the selected EDUs in our work.

For the fusion of the converted RST graphs, we gathered all the labels assigned to the EDUs within those graphs. This leads us to focus solely on those EDUs marked with a '1' label, which are then concatenated to serve as the input for our fusing model. For the article-summary pairs, the outputs are the indices of important EDUs. Essentially, both datasets are the combined sequences of the selected EDUs.

For the evaluation of the converted RST graphs, we encounter a situation where the associated graphs lack referenced summaries, to tackle this issue, we need to match each graph with the summaries from the article-summary pair dataset by using the hash table.

To align the RST discourse tree dataset with the CNN/Daily Mail article-summary pair dataset, we established a hash table. In this table, the document IDs serve as the strings to generate the hash values, and the values in the table are the corresponding summaries from the article-summary pair dataset. The function we used for generating hash value is *SHA-256* hash function. The summaries are stored as dictionaries within JSON files, with file names corresponding to the respective RST graph files.

Owing to the pre-training tasks of BART, it is capable of capturing the rich semantic information from the given text and making predictions regressively. We fine-tune it to make BART fit more to the given downstream task and the corresponding dataset. The fine-tuning object for our work is making BART have a better performance when taking the concatenated important EDUs as inputs, especially when the whole sequence is not coherent, the model can be trained to be capable of outputting coherent sequences with correct grammar.

With the filtering of the selecting model, the input sequences are much shorter than the original inputs, which reduces the requirements of the device. We modified the max length of the input from 2048 to 1024, in order to boost the encoding process.

Experiments

Implementation Details

EDU segmentation: We use the pre-trained model proposed by Wang, Li, and Yang (2018), the environment is set according to the repository[2] uploaded by Li, Wu, and Li (2020).

[2] https://github.com/PKU-TANGENT/EDUSum

Package	Version
pytorch	1.12.0-gpu
cuda	11.7
torch_geometric	2.3.1
transformers	4.30.2

Table 1: EDU selection environment

Package	Version
pytorch	1.10.0-gpu
cuda	10.2
hydra-core	1.0.7
omegaconf	2.0.6
bitarray	2.6.2
sacrebleu	2.3.1
regex	2022.10.31

Table 2: EDU fusion environment

EDU Selection: GNN structures are utilized from PyG[3]. Specifically, we use the environment configured as shown in Table 1.

In the data processing stage, the data structure *Data* is employed for storing the graphs. *DataLoader* sews up a certain number of graphs in a mini-batch into a large graph. The batch size of the training stage is 24 and the max length of each EDU is 20. We leverage the multi-head mechanism in each GAT layer, the number of heads in each layer is set to 4.

The training process is done on a 4-GPU server, where each GPU has 32 gigabytes of memory. We use the multi-GPU mechanism to boost the training speed.

For the greedy selecting method, we employ the code from Wang, Li, and Yang (2018) and make modifications to the calculation of the ROUGE scores. The datasets may contain some empty EDUs or EDUs that only contain the invalid characters. When meeting those EDUs, we set the rouge scores for all sentences to 0.

EDU Fusion: We use the pre-trained BART (large) model as our baseline, and the fine-tuning is also done based on the model. The model parameters can be accessed from a GitHub repository [4]. With two different datasets preprocessed by the GNN and the Greedy Method respectively, the max length of the input sequence is modified from 2048 to 1024 (Due to the redundancy reduction of the original articles, the inputs are far shorter than before). Other hyper-parameters are set the same as the BART model fine-tuning on the original CNN/Daily Mail dataset. The environment is set as in Table 2.

Other than using BART, we also utilized GPT-3 as one of the testing models. GPT-3 is one of the most powerful language models proposed by OpenAI, we use the fastest model, *Ada*, in GPT-3 and since OpenAI has a series of well-defined APIs for users to use their models fast and easily, we do not need to do further configuration to set up the hyper-parameters as all the hyper-parameters were preconfigured.

	Number of Graphs
Train	287,227
Test	11,490
Valid	13,368

Table 3: Article-summay pair dataset size

	Number of Graphs
Train	287,227
Test	11,490
Valid	13,368

Table 4: Converted RST discourse tree dataset size

Evaluation

The evaluation metrics for all the experiments are the ROUGE scores of the precision, recall and F1 scores. We first leverage the Penn Treebank Tokenizer from StanfordNLP[5] to tokenize all the articles and referenced summaries into lists of tokens, subsequently followed by the utilization of *files2rouge* package to automatically calculate the metrics between two files. The command *files2rouge* requires the two files (hypotheses and references) to have the same number of lines.

Results

Our work focuses on the comparison between the performance of BART fine-tunned on the original CNN/Daily Mail dataset and the performance of BART fine-tunned on the concatenated sequences of selected EDUs from the CNN/Daily Mail dataset.

With different pre-processing methods, we have two different datasets processed based on the CNN/Daily Mail dataset, the article-summary pair dataset (Table 3) and the converted RST tree dataset (Table 4).

Our results are presented in Table 5. The validation metrics are the F1 scores of all the ROUGE metrics. The first line introduces the baseline of BART-large, and it is directly referenced from the paper where BART was proposed. The second line shows an alternative method that combines the greedy method and Ada (one of the GPT-3 models). The results from the third line are obtained by replacing the GPT-3 model with BART based on the method of the second line. The last line is the result of the combination of GNN and BART, which shows the best ROUGE-1 and ROUGE-2 scores among all the methods.

All the validations are based on the source files that contain the sequential concatenations of selected EDUs and the target files that contain the highlights of articles. Compared with BART-Large, all the experiments with EDU selection outperform it, among which the GNN method has the best performance in terms of the ROUGE-1 (58.8%) and ROUGE-2 (35.7%), and the greedy method has the best

[3] https://github.com/pyg-team/pytorch_geometric

[4] https://github.com/facebookresearch/fairseq/tree/main/examples/bart

[5] Official website: https://stanfordnlp.github.io/CoreNLP/

	ROUGE-1	ROUGE-2	ROUGE-L
BART-Large	44.16	21.28	40.90
Greedy Method + GPT-3	48.74	23.15	45.35
Greedy Method + BART	58.12	34.68	**55.37**
GNN + BART	**58.80**	**35.70**	55.30

Table 5: Experimental results on CNN/Daily Mail dataset

ROUGE-L score with about 0.07 percent higher than the GNN method, reaching a percentage of 55.37.

The results show that the summarization performance is better after changing the inputs from the whole article to the combinations of EDUs. With the same constituent of the dataset, BART outperforms Ada. Additionally, with more parameters, GNN has a better performance in terms of selecting salient EDUs than the greedy method.

Conclusion

In our work, we choose to optimize the datasets by truncating the sentences in articles into fine-grained units, EDUs, and selecting the most informative ones as the candidates to be fused. We use GNN and the greedy method to select those salient EDUs separately and apply BART to fuse them. In all our stages, the segmentation and the selection stages serve as the encoder to label the important EDUs and concatenate the EDUs that are labeled with '1' which indicates those EDUs as important. The decoder consists of the BART-large pretrained model. When using GNN as the selecting method, the encoder and the decoder can be trained separately, but the inputs of the decoder have to be restricted to the outputs of the encoder. Our stage-wise method outperforms the BERT-large baseline with a 14.64% higher ROUGE-1 score, 14.42% higher ROUGE-2 score and 14.4% higher ROUGE-L score in text summarization.

Transformer-based models outperform most of the other neural models in terms of natural language generation, at the same time, they require more calculation amount which means they take more time to be trained and fine-tuned. Complete articles naturally have rhetorical structures that can assist in comprehending the semantic relations, and this type of structure can always be represented as graphs, where Graph Neural Networks can be used . Graph neural networks, with a good performance of processing graph-based data, although require strict pre-processing of the datasets, can lead to a better performance of the results of the graph-based tasks.

Our model is only proven to work better on the CNN/Daily Mail dataset whose reference summaries are usually more than one sentence, this reduces the difficulty of selecting the salient EDUs. There are a large number of parameters in our model which require a long time to fine-tune them. In our future work, we can try to use a better selection method that has a smaller volume and has a better performance in choosing salient units.

Acknowledgments

The research reported in this paper was conducted at the University of Lethbridge and supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada discovery grant and the NSERC Alliance - Alberta Innovates Advance Program grant. We thank also the anonymous reviewers for their comments.

References

Bing, L.; Li, P.; Liao, Y.; Lam, W.; Guo, W.; and Passonneau, R. 2015. Abstractive multi-document summarization via phrase selection and merging. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1587–1597. Beijing, China: Association for Computational Linguistics.

Carlson, L.; Marcu, D.; and Okurovsky, M. E. 2001. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

Chen, Y.-C., and Bansal, M. 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)*, 675–686. Melbourne, Australia: Association for Computational Linguistics.

Clevert, D.-A.; Unterthiner, T.; and Hochreiter, S. 2016. Fast and accurate deep network learning by exponential linear units (elus).

Cohan, A.; Dernoncourt, F.; Kim, D. S.; Bui, T.; Kim, S.; Chang, W.; and Goharian, N. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 615–621. New Orleans, Louisiana: Association for Computational Linguistics.

Consortium, L. D., and Company, N. Y. T. 2008. *The New York Times Annotated Corpus*. LDC corpora. Linguistic Data Consortium.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Fey, M., and Lenssen, J. E. 2019. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Hermann, K. M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. In Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Ji, Y., and Eisenstein, J. 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)*, 13–24.

Lebanoff, L.; Song, K.; Dernoncourt, F.; Kim, D. S.; Kim, S.; Chang, W.; and Liu, F. 2019. Scoring sentence singletons and pairs for abstractive summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2175–2189. Florence, Italy: Association for Computational Linguistics.

Lebanoff, L.; Muchovej, J.; Dernoncourt, F.; Kim, D. S.; Wang, L.; Chang, W.; and Liu, F. 2020. Understanding points of correspondence between sentences for abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, 191–198. Online: Association for Computational Linguistics.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. Online: Association for Computational Linguistics.

Li, P.; Lam, W.; Bing, L.; Guo, W.; and Li, H. 2017. Cascaded attention based unsupervised information distillation for compressive summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2081–2090. Copenhagen, Denmark: Association for Computational Linguistics.

Li, Z.; Wu, W.; and Li, S. 2020. Composing elementary discourse units in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6191–6196. Online: Association for Computational Linguistics.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1412–1421. Lisbon, Portugal: Association for Computational Linguistics.

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* 8(3):243–281.

Narayan, S.; Cohen, S. B.; and Lapata, M. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization.

Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; and Auli, M. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. New Orleans, Louisiana: Association for Computational Linguistics.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training. *OpenAI*.

Rush, A. M.; Chopra, S.; and Weston, J. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 379–389. Lisbon, Portugal: Association for Computational Linguistics.

See, A.; Liu, P. J.; and Manning, C. D. 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)*, 1073–1083. Vancouver, Canada: Association for Computational Linguistics.

Song, K.; Tan, X.; Qin, T.; Lu, J.; and Liu, T.-Y. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Thadani, K., and McKeown, K. 2013. Supervised sentence fusion with single-stage inference. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 1410–1418. Nagoya, Japan: Asian Federation of Natural Language Processing.

Tu, Z.; Lu, Z.; Liu, Y.; Liu, X.; and Li, H. 2016. Modeling coverage for neural machine translation. *arXiv preprint arXiv:1601.04811*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer networks. *Advances in neural information processing systems* 28.

Wang, Y.; Li, S.; and Yang, J. 2018. Toward fast and accurate neural discourse segmentation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 962–967. Brussels, Belgium: Association for Computational Linguistics.

Xu, J.; Gan, Z.; Cheng, Y.; and Liu, J. 2020a. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5021–5031. Online: Association for Computational Linguistics.

Xu, S.; Li, H.; Yuan, P.; Wu, Y.; He, X.; and Zhou, B. 2020b. Self-attention guided copy mechanism for abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 1355–1362. Online: Association for Computational Linguistics.