ITC 2/51
Information Technology and Control
Vol. 51 / No. 2 / 2022
pp. 345-355
DOI 10.5755/j01.itc.51.2.30796

GATSum: Graph-Based Topic-Aware Abstract Text Summarization

Received 2022/02/24 | Accepted after revision 2022/05/11

http://dx.doi.org/10.5755/j01.itc.51.2.30796

HOW TO CITE: Jiang, M., Zou, Y., Xu, J., Zhang, M. (2022). GATSum: Graph-Based Topic-Aware Abstract Text Summarization. *Information Technology and Control*, 51(2), 345-355. http://dx.doi.org/10.5755/j01.itc.51.2.30796

GATSum: Graph-Based Topic-Aware Abstract Text Summarization

Ming Jiang, Yifan Zou, Jian Xu, Min Zhang
Hangzhou Dianzi University, College of Computer, Hangzhou, China, 310000;
emails: jmzju@163.com, hdu_zouyifan@163.com, jian.xu@hdu.edu.cn, hz_andy@163.com

Corresponding author: Yifan Zou (hdu_zouyifan@163.com)

The object of text summarization is to cut down the extent of the text into a summary containing key data. Abstract methods are challenging tasks, it is necessary to devise a machine-processed to pick up the message from the text with advantage, and after that make a summary. However, most of the existing abstract approaches are difficult to capture global semantics, ignoring the impact of global information on obtaining important content. To solve this difficulty, this paper suggests a Graph-Based Topic Aware abstract Text Summarization (GTASum) framework. Specifically, GTASum seamlessly incorporates a neural topic model to find potential topic information, which can maintain document-level characteristics for generating summaries. In addition, the model integrates the graph neural network which can effectively capture the relationship between sentences through the document representation of graph structure, and simultaneously update the local and global information. The further discussion showed that latent topics can help the model capture salient content. We practiced experiments on two datasets, and the result shows that GTASum is superior to many extractive and abstract approaches in terms of ROUGE measurement. The result of the ablation study proves that the model has the ability to capture the original subject and the correct information and improve the factual accuracy of the summarization.

KEYWORDS: Text Summarization, Abstract, BERT, Neural topic model, Graph attention network.

1. Introduction

The purpose of text summarization is to aid people in quickly grasping the key data of the text, and it is a valuable job in NLP (Natural Language Processing) tasks. Currently, summarization approaches can separate into two types: extractive and abstract. Extractive models mainly copy important information from the original text and aggregate them into text [10, 17]. This can usually retain the salient information of the original text and has the correct grammar, but it is easy to generate a lot of redundant messages. In contrast, neural-based abstract models usually use the sequence-to-sequence framework, which can understand text content and generate words that are not appearance in the source document. Abstract models are closer to the essence of summarization and have the potential to generate high-quality summarization. Therefore, the research in this paper is biased towards abstract approaches.

A key point of summarization is topic information modeling. Although current Transformer-based abstract summarization models have achieved great success which because they can effectively capture contextual features and obtain local semantic information between sentences and paragraphs, they ignore higher-level semantic information. To better capture the global semantics of input documents, researchers try to introduce topic information to conduct the process of making a summary. Topic models such as LDA [3] (Latent Dirichlet Allocation), PFA [41] (Poisson Factor Analysis), NVDM (Neural Variational Document Model) [22], and NTM [7] (Neural Topic Model) can provide additional information for document understanding. The distribution of all tokens in the vocabulary is described by taking topic information as a global variable. For the text summarization domain, by integrating document-level characteristics into the summarization model, we trust it will enhance the representation of the model.

However, despite the extensive literature incorporating topic modeling into the text summarization task [31, 39], we found that quite a few previous studies used topic models as a single source of information rather than unified and jointly improved the text summarization task and topic models. This attracted our attention, and GNN (Graph Neural Networks) was considered to solve this problem. Lately, GNN has been universally used for cross-sentence relation modeling for summarization tasks [15]. Several studies [35, 37] set up document graphs according to discourse analysis. But, this method relies on external tools, which possibly bring out semantically fragmented outputs [20]. To sum up, GAT (Graph Attention Network) is constructed using sentence context representation and topic information, and the document context vector and topic information are updated simultaneously. This can not only jointly update local semantics and global semantics, but also reduce the problem of semantic fragmentation.

In this work, we suggest a novel GTASum model (Graph-Based Topic-Aware abstract text summarization). First, the document is encoded with BERT [6] (Bidirectional Encoder Representation from Transformers) to obtain contextual sentence representations; meanwhile, NTM [7] is used to learn the potential topic of the document. Second, construct a heterogeneous document graph that contains sentence representation nodes and potential topic nodes, then revise them using a adapted GAT [30]. Finally, the sentence representations containing topic information are fed into a Transformer-based decoder to generate summaries. Overall, the primary contributions of us are as follows:

1. A novel Graph-Based Topic-Aware abstract text summarization model is proposed, which helps to capture global semantic information and provides guidance in the procedure of making a summary. This solves the problem of text summarization lacking global semantic information.
2. A heterogeneous document graph is designed to jointly update local semantics and global semantics and reduce the problem of semantic fragmentation.
3. GTASum is evaluated on two standard datasets (CNN/DailyMail and XSum) and outperforms many existing extractive and abstract models in terms of ROUGE (Recall-Oriented Understudy for Gisting Evaluation) measure.

2. Related Work

2.1. Abstract Text Summarization

In abstract text summarization, sequence-to-sequence [28] is the most mainstream framework. The early sequence-to-sequence framework is mainly based on RNN [5] (Recurrent Neural Network). This traditional model first uses some LSTM (Long Short-Term Memory) units to obtain the input sequence, encodes it into a fixed-length vector representation, and then uses some LSTM units to read the vector and decoded into the output sequence [4]. However, researchers found when the input sequence is longer, it is more difficult for the model to acquire a reasonable vector representation, so the attention mechanism [2] was introduced into the field of text summarization. The problem derived from this is that there are a large number of OOV (Out-of-Vocabulary) problems and generation repetition problems, and the PGN+-Cov [27] (Pointer Generator Network and Coverage mechanism) model cleverly optimizes the cover words in the generation process in a penalized manner, and achieves good results. The traditional RNN structure is suitable for natural language modeling because it can handle variable-length inputs, but it has the disadvantages of being difficult to parallelize, difficult to train, and difficult to capture long-distance and hierarchical dependencies. Transformer [29] proposed by Google in 2017 abandoned RNN. It used self-attention to make the model parallelizable, and the improvement effect was significant, which leads more and more researchers using Transformer to replace RNN. In 2018, the Transformer-based pre-trained language model BERT was born, providing a large number of performance gains for summarization tasks [8, 19]. The current state-of-the-art models of text summarization, including BART [16] (Bidirectional and Auto-Regressive Transformers), PEGASUS [38] (Pre-training with Extracted Gap-sentences for Abstractive Summarization), and ProphetNet [11], all use Transformer-based architectures. Therefore, this paper adopts BERT as the document encoder to learn sentence context representation and obtain local information.

2.2. Topic Model

Topic modeling is a strong method to learn document-level characteristics, which can explore the hidden semantic structure of text [33]. However, it has not been applied to the field of text summarization until recent years. A core idea of topic models is that documents are mixtures of topics, each distributed over words in the corpus vocabulary. In order to obtain all distributions, the traditional approach is to utilize the LDA algorithm. Recent studies have introduced neural networks, and topic models have been improved by general auto-encoders or neural VAEs (Variational Auto-encoders), which have also derived variants such as NVDM, GSM [21] (Gaussian Softmax Distribution), and NTM. The researchers conducted many experiments to explore the effect of topic modeling. Ailem et al. [1] build a topic-augmented decoder which can generate summaries based on input documents and document latent topics. Narayan et al. [25] suggested a topic-conditioned sequence-to-sequence model which is using CNN (Convolutional Neural Networks) framework. Wang et al. [32] introduced topic helpers for Transformer-based summarization models, along with topic data explored by a separate topic component.

The various experiments above show that the use of topic models to capture global semantics is very effective, and topic models can also be used as components alone to provide additional information [42]. Topic-features have been applied to the language generation process to guide text generation with specified topics

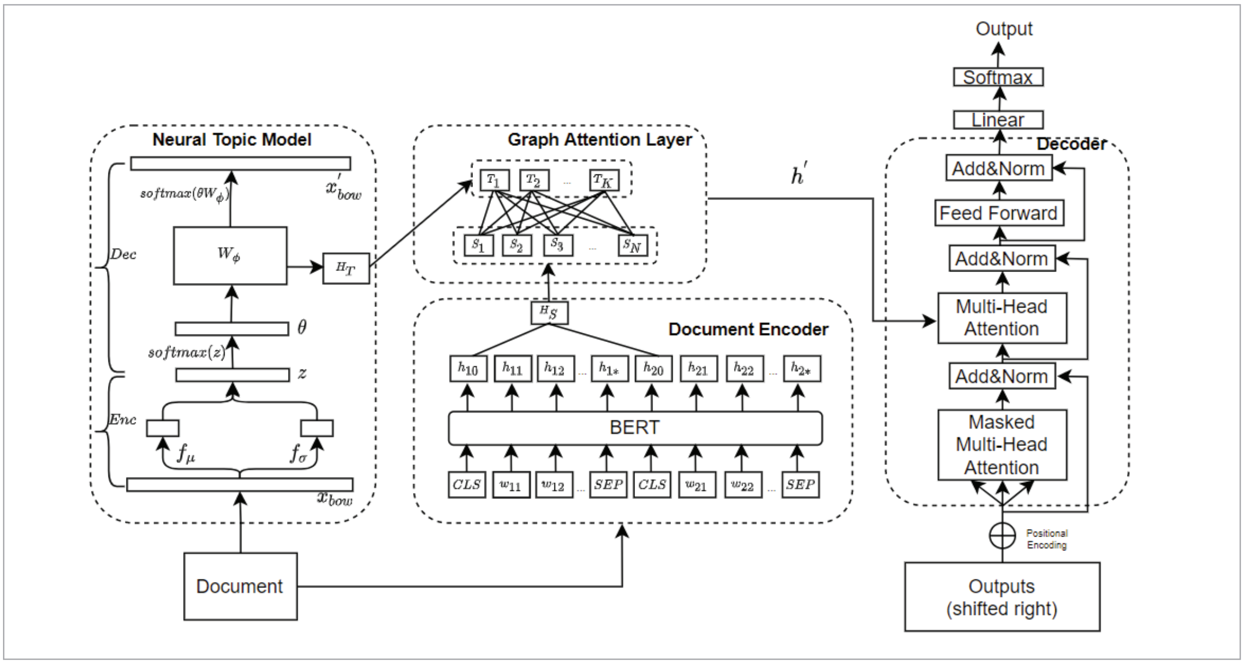
2.3. Graph-based Summarization

In earlier studies, there are mainly two algorithms, TextRank [23] and LexRank [9], which build document graphs based on sentence similarity and extract summaries without supervision. Thereafter, Wei et al. [34] suggested constructing a document graph that contains words, topics, and sentences, and utilize Markov chains to learn the graph and generate summaries. Recently, GNN networks have attracted much research in text summarization tasks [11, 36, 37]. And these text summarization models which are under the GNN framework only constructed document graphs based on words or sentences. On the contrary, we introduce sentence nodes and topic information nodes to construct document graph at the same time, which can not only solve the joint updating of sentence context representation and topic information but also reduce the problem of semantic fragmentation. In addition, since the traditional GAT is designed for homogeneous graphs and is not suitable for this model, it is necessary to modify the traditional GAT to design a new heterogeneous document graph.

2. Model

This section will describe our model structure, according to Figure 1. The GTASum model contains three parts: First, Document encoder. Second, Neural topic models. Third, Graph Attention Layer. Given a document containing N sentences $D = \{s_1, s_2, ..., s_N\}$,

Figure 1
Our model



given a document containing M reference summaries $Y = \{y_1, y_2, ..., y_m\}$. The goal of GTASum is to generate abstract summaries from documents D and reference summaries Y. First given a document D, the NTM will try to obtain the distribution of the topic of the document and a set of topic representations. Simultaneously, the document encoder with the pre-trained language model BERT will get contextual representations for every sentence. The GAT uses the topic representations and the contextual representations to construct a heterogeneous document graph, and revise their node representations at the same time. Then the graph is encoded, the sentence representation and topic information will be combined and sent to the Transformer-based decoder to generate a summary. Each part will be explained in detail below.

3.1. Document Encoder

The purpose of the document encoder is to get a sentence contextual representation. Based on powerful pre-trained language models, self-supervised learning methods can be run on large-scale corpora to learn better feature representations for each vocabulary. It uses Transformer as the main framework, combined with the Attention mechanism, to better capture the bidirectional relationship in the sentence and solve the long dependency problem, so this paper chooses to obtain the feature representation of the document based on BERT. For every sentence, we insert <CLS> at the start and <SEP> at the end, after that in order to get the hidden representations, all tokens will be put into the BERT layer. By the following formula:

$$\{h_{10}, h_{11}, ..., h_{n*}\} = \text{BERT}(\{w_{10}, w_{11}, ..., w_{n*}\}), \quad (1)$$

where w_{ij} represents the j-th word of the i-th sentence; w_{i0} and w_{i*} represent <CLS> and <SEP>, respectively. After BERT encoding, we treat the hidden states of <CLS> $H_s=\{h_{10},...,h_{n0}\}$ as contextual representations of corresponding sentences, and they will be further enriched by topic information in later steps.

3.2. Neural Topic Model

Our topic information learning is based on the extraction of latent topic information via a NTM. NTM is implemented based on VAE [14] framework. It learns document latent topics through a sequence-to-sequence encoder. As shown in Figure 2, it is a representation of the NTM model. The model is given an input document and obtains a bag of words, where V is the number of words x_{box}. The NTM is defined as follows:

- In the encoder: Given an input bag of words x, $\mu = f_\mu(x)$ and $log\sigma=f_\sigma(x)$ are generated, where the μ and the σ are learnable prior parameter. Both functions f_μ and f_σ are linear transformation functions with ReLU activations.
- In the decoder: There are three main steps in the decoding step. First, Gaussian Softmax [21] is adopted to describe the topic distribution z~$N(\mu,\sigma)$ and $\theta=softmax(z)$, where z is the latent topic variable, $\theta\in R^K$ is the result of z normalization, and its dimension is the predefined topic number K. Second, by learning $p_w=softmax(W_\varphi\theta)$ to predict the occurrence probability of words $p_w\in R^V$, where $W_\varphi\in R^{V\times K}$ is the topic-word distribution matrix similar to the LDA topic model. Third, extract each word from p to reconstruct the input word bag x_{box}.

In this model, the intermediate parameters W_φ is used, and they are all encoded into topic information. So we can further construct topic representations by them, as shown below:

$$H_T = f_\phi(W_\phi^T) = \{t_1,\ldots,t_K\}, \tag{2}$$

where $H_T\in R^{K\times dt}$ is a set of subject representations with a predefined dimension of d_t.

The GAT will put to use H_T to enhance the representations of all sentences. Unlike others [40], GTASum does not take topic information as a fixed feature of an external model, but learns it through neural learning methods and will be dynamically revised through the entire framework network.

3.3. Graph Attention Layer

First, a graph needs to be constructed. The definition of the graph is given as follows: Give $G=\{V,E\}$ indicate an undirected graph, where $V=V_S\cup V_T$ is the set of nodes. $V_S=\{S_1,S_2,\ldots,S_N\}$ represents N sentence nodes, $V_T=\{T_1,T_2,\ldots,T_K\}$ represents K topic nodes. E is the set of edges, and $E=\{e_{11},\ldots,e_{NK}\}$ represents the weight of the edge between the i-th sentence node and the j-th topic node.

We initialize sentence node and topic node vectors with hidden representations learned from BERT H_S and topic representations learned from NTM H_T. The final representation of each node is then obtained through a graph attention network (GAT). GAT first performs self-attention processing, learns the attention value between nodes, and normalizes it:

$$e_{ij} = Leaky\,\mathrm{Re}\,LU(\vec{a}[W_bS_i \parallel W_bT_j]), \tag{3}$$

$$\alpha_{ij} = soft\max(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{l\in N_i}\exp(e_{il})}, \tag{4}$$

where S_i represents the i-th sentence node, T_j represents the j-th topic node, W_b is a learnable parameter, N_i represents a neighbor node, // is a vector splicing operation, a is a feed-forward neural network, can map the splicing vector to a real number, and *LeakyRELU()* is an activation function. However, considering the use of two semantic units, sentence and topic, we need to modify the ordinary GAT graph, Hu et al. [18] inspired us to consider mapping topics and sentences into an implicit common space, and then compute the attention value. So modify Equation(3) to Equation(5):

$$e_{ij} = Leaky\,\mathrm{Re}\,LU(\vec{a}[f_S(S_i) \parallel f_T(T_j)]), \tag{5}$$

among them, the parameters W_b are removed, and a nonlinear transformation function f_S and f_T are used to project topic nodes and sentence nodes into the common space vector.

After the attention value α_{ij} is obtained, the neighbor nodes $h_i^{`}$ can be weighted and summed to obtain the new node feature representation that the context of the article is the contextual knowledge representation:

$$S_i^{'} = \sigma(\sum_{j\in N_i}\alpha_{ij}W_cT_j), \tag{6}$$

where N_i represents the neighbor node, W_c is a learnable parameter, T_j represents the j-th topic node, $S_i^{`}$ is the i-th sentence node weighted by the topic node, and $\sigma()$ is a sigmoid function.

Finally, a sentence feature representation with topic information is generated and output to the decoder:

$$h^{'} = \{S_1^{'},\ldots,S_n^{'}\}. \tag{7}$$

When using GAT, it is not necessary to pay attention and calculate on all nodes, instead, pay attention to the neighbor nodes. That is, assigning weights of different values to the neighbor nodes of each node, the attention mechanism can be used to focus on the nodes that are more closely related to the text while ignoring the nodes that are less related to the text. The calculation process can better grasp the global information and knowledge of the text without losing local information, and the calculated attention is used to represent the global information.

3.4. Enhanced Learning

The decoder of the framework adopts the Transformer framework. The Transformer-based decoder itself has its own encoding-decoding attention layer, which can effectively gather key information and capture the information that needs to be mined for text generation. In each decoding step, the components of a sequence are output according to the input sequence, and the generation is repeated until a termination symbol is encountered. Each time step is doing input for the next time step, and then outputs the final result through the decoder side. This paper stacks a 6-layer Transformer, each with a multi-head attention layer and a feed-forward layer. The final output of the encoder $h`$ is the contextual embedding with topic information, which can be input into the decoder. Finally, we will jointly train the NTM and decoder to reduce the loss.

For NTM, we define the objective function as the negative evidence lower bound, as follows:

$$L_{NTM} = D_{KL}(p(z) \| q(z|x)) - E_{q(z|x)}[p(x|z)], \tag{8}$$

where the first term represents the Kullback-Leibler divergence loss, and the second term represents the reconstruction loss. $q(z|x)$ and $p(x|z)$ indicate the encoder and decoder network.

For the decoder, during training, the maximum likelihood function objective is used to minimize, as follows:

$$L_{Trans} = -\frac{1}{D}\sum_{y,x\in D}\log p(y|x;\theta), \tag{9}$$

where x is the text, y is the reference summary, D is the training set, and θ is the model parameter.

The final model loss function is a linear combination of the above two losses and hyperparameters to balance their weights:

$$L = L_{Trans} + \lambda L_{NTM}, \tag{10}$$

where $\lambda \in [0,1]$.

4. Experimental Setup

4.1. Datasets

To better train the model, this paper evaluates GTA-Sum on two standard datasets: CNN/DailyMail [12] and XSum [25]. Both datasets are public, commonly used text summarization datasets. Both of them have been widely used in automatic text summarization tasks in the past two years. The data set is pre-divided into a training set Train, a validation set Valid and a test set Test. The data volume of the two is shown in the Table 1.

Table 1

Datasets structure

Datasets	Train	Valid	Test
CNN/DM	287188	13367	11490
XSum	204045	11332	11334

4.2. Evaluation Index

This paper adopts ROUGE-1 to measure the unigram recall between abstracts and documents, ROUGE-2 to measure similar bigram recalls, and ROUGE-L to measure the longest common sub-sequence between abstracts and documents. These three indicators are automatic evaluations. ROUGE compares auto-generated summaries and hand-crafted standard summaries by counting the overlapping vocabulary between the two. The larger the ROUGE value, the better the generation effect. This approach has become a metric for evaluating the generated summary model. The calculation method is as follows:

$$ROUGE-N = \frac{Count_{match}(ref, pred)}{Count(ref)}, \tag{11}$$

where $Count(ref)$ is the length of the manual standard abstract and $Count_{match}(ref, pred)$ is the number of words that appear in both the manual standard abstract and the automatically generated abstract. They can be calculated by the Pyrouge package

4.3. Experimental Settings

GTASum is trained end-to-end, using a base version of the BERT model to extract lexical features for all experiments. By preprocessing the corpus articles, the input articles are truncated into 512 words. The decoder uses a 6-layer Transformer. This paper uses Adam [13] as the optimizer with a learning rate of 2e-4. And set the epoch to 30 and the batch size to 8. For NTM, we select the most frequent 2000 words as the topic vocabulary, which comes from the training set, and then set the number of topics as K=128. Set λ=0.75 to equipoise the loss of information selection and topic modeling. In GAT, take GAT Layer=2, set the number of topic node attention heads to 4, set the number of sentence node attention heads to 6, and set the hidden size to 128. In each abstract, replace all entities in the relevant reference chain in the abstract with the canonical entity used in the diagram.

For all input and output content, words that appear less than 3 times are replaced by <UNK>. The post-processing step removes duplicate sentences and duplicate compound sentences. During training, if the model performance drops, halve the learning rate and use the halved learning rate for fine-tuning.

4.4. Overall Performance

This section evaluates the proposed model, testing the accuracy and consistency of the proposed GTASum model on the task of generating abstract summary sentences. To better compare the models, we compared GTASum with the following powerful summarization models, both extractive and abstract:

1 Lead-3 [27]: is a rules-based approach. Take 3 sentences from the article as its summarization.
2 SummaRuNNer [24]: is an abstract model summary based on two-layer bidirectional GRU-RNN (Gated Recurrent Unit and Recurrent Neural Network). It defines the summarization problem as a sequence classification problem, and for each sentence, a binary classifier is learned to decide whether to include it or not.
3 BERTSUM [19]: is the first extractive summarization model which under BERT-based. Insert multiple segmentation tokens in the document to get a representation of all sentence.
4 PGN+Cov [27]: flexibly copy words from the source text through pointers, so as to automatically choose to generate new words or copy words according to the probability, not only retaining the original text information but also consolidating the model generation ability. In addition, the replication mechanism ensures that duplicate words are not generated.
5 BART [16]: uses a bidirectional encoder to enrich sequence understanding, then use a left-to-right decoder to make a summary.

Tables 2-3 shows the automatic evaluation results of the proposed model GTASum and comparative models on the CNN/DailyMail datasets and the XSum datasets, and Table.3 shows examples of the generated summaries.

- **Result on CNN/DailyMail:** The Table.2 shows the evaluation results on the CNN/DailyMail datasets. We compare GTASum with kinds of strong baseline models, including strong abstraction. GTASum achieves better performance in most cases and achieves the best scores on ROUGE-1 and ROUGE-2, proving that global semantic information plays a guiding role in model generation. The rule-based Lead-3 and the extractive model BERTSUM perform better on the CNN/DailyMail dataset which is biased towards extractive summarization. BART has the best performance on ROUGE-L, which shows its language representation ability based on a powerful pre-trained corpus. However, on ROUGE-1 and ROUGE-2, GTASum improves by 1.1% and 0.2%, respectively. The excellent performance of GTASum underlines the importance of introducing topic model components and graph attention network components to improve the generation effect.

Table 2
Results on CNN/DailyMail

Models	Rouge1	Rouge2	RougeL
Lead-3	40.29	17.68	36.47
SummaRuNNer	39.60	16.20	35.30
BERTSUM	42.13	19.60	39.18
PGN+Cov	39.53	17.28	36.38
BART	**44.16**	**21.28**	**40.90**
GTASum(our's)	**44.46**	**21.32**	**39.84**

- **Result on XSum:** The Table.3 shows the evaluation results on the XSum dataset. The result of the XSum datasets retains just a short text, and the models need to condense the information and generate sentences containing key information, so the dataset is more inclined to generate segmented text summaries. BART absorbs the specific characteristics of BERT's bidirectional encoder and GPT's left-to-right decoder and is based on the standard sequence-to-sequence Transformer model, which makes it more appropriate for text generation tasks. While BART has made progress on generation tasks, it has also achieved state-of-the-art on some text understanding tasks. The performance of GTASum on XSum is better than that of the extraction model, and it has achieved results comparable to BART, which also shows that the topic model has played a certain role in understanding and generating text.

Figure 2 shows examples of our model.

4.5. Ablation Study

This section discusses the impact of each part of the GTASum model on the final performance. As shown in Table.3, to evaluate the effectiveness of the GTA-Sum module, we compared the full GTASum model with two ablation variants. First, without NTM, remove the NTM module. We can build a document graph that contains fully connected sentence nodes. It is also will be considered as computing self-attention on the top of BERT. Second, without GAT, remove the GAT module. Then splice the overall topic vector and each sentence representation, and send them into the decoder.

Table 3
Result on XSum

Models	Rouge1	Rouge2	RougeL
Lead-3	16.30	1.60	11.90
SummaRuNNer	-	-	-
BERTSUM	38.79	16.40	31.16
PGN+Cov	28.10	8.02	21.22
BART	**45.4**	**22.27**	**37.25**
GTASum(our's)	**44.60**	**21.53**	**36.66**

Figure 3
Ablation study results

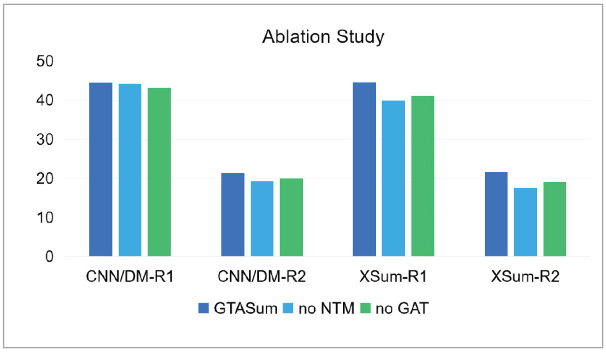


Figure 2
Examples of our model

Dataset	Source Article	Target	Summary by our model
CNN/ DialyMail	*Despite the hype surrounding its first watch, the iPhone is still the engine behind Apple's phenomenal success, its latest figures have revealed. The results far surpassed most analysts' expectations for the first three months of the year, when sales traditionally fall from their holiday-season peak. Apple sold more than 61 million...*	*Apple sold more than 61 million iPhones in the quarter . Apple didn't report any results for the new Apple Watch . Believed around 2 million watches have been sold, according to estimates .*	*Apple sold more than 61 million iPhones in the quarter, accounting for more than two-thirds of its $58 billion in revenue. iPad revenue fell 29% to $5.4 billion, continuing a steady decline in tablet sales. Apple also announced an expansion of its effort to return more of its cash war chest to investors.*
XSum	*The London trio are up for best UK act and best album, as well as getting two nominations in the best song category. "We got told like this morning 'Oh I think you're nominated'", said Dappy...*	*N-Dubz have told Newsbeat they are socked to have picked up four nominations for the Mobo awards*	*N-Dubz have revealed they were surprised to be nominated for four Mobo Awards*

As shown in Figure 3, the results of ablation studies on 2 datasets are presented, from that, the following conclusions will be observed: the full GTASum model gets the best outperforms, it shows that all module is needful and combine them for the best performance.

4.6. Influence of Potential Topics

This section we desire to comprehend how underlying topics can guide the process of making a summary. For this reason, we set forth the local weight of a sentence as the weighted sum of the attention scores between the sentence and every topic, namely:

$$Topic-Weight_i = \sum_{j=1}^{K} \theta^j \alpha_{j,i}, \quad (12)$$

where *Topic-Weight*$_i$ is the weighted topic weight of the i-th sentence; θ_j is the weight of the j-th topic in the document, which is the normalized document topic distribution learned by NTM in Section 3.2; α_{ij} is from the j-th topic node to the i-th sentence Attention scores of nodes, learned from Equation (4).

5. Conclusion and Future Work

In this study, we integrate the global semantic structure of the text by proposing a novel graph-based topic-aware abstract text summarization model GTASum and investigate the abstract text summarization problem. In particular, neural topic models, BERT, and graph neural networks are combined for summarization. Extensive experiments are conducted on two real-world datasets to compare GTASum with several methods. It turns out that GTASum outperforms the vast majority of classical models, with performance approaching industry-leading methods. In future work, we will continue to explore incorporating more types of high-level semantic units into the model to improve the performance and robustness of the model.

Acknowledgement

This work is supported by Xiaoshan District Technical Plan Project (No. 2020102).

References

1. Ailem, M., Zhang, B., Sha, F. Topic Augmented Generator for Abstractive Summarization. Computing Research Repository (CoRR abs), 2019. https://doi.org/10.48550/arXiv.1908.07026
2. Bahdanau, D., Cho, K., Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. International Conference on Learning Representations (ICLR), 2015.
3. Blei, D. M., Ng, A. Y., Jordan, M. I. Latent Dirichlet Allocation. The Journal of machine Learning Research (JMLR), 2003, 3, 993-1022.
4. Cho, K., Merriënboer, B. V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. Empirical Methods in Natural Language Processing (EMNLP), 2014, 1724-1734. https://doi.org/10.3115/v1/D14-1179
5. Chung, J., Gülçehre, C., Cho, K. H., Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. Computing Research Repository (CoRR abs), 2014. https://doi.org/10.48550/arXiv.1412.3555
6. Devlin, J., Chang, M. V., Lee, K., Toutanova, K. Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2019, 4171-4186. https://doi.org/10.18653/v1/N19-1423
7. Ding, R., Nallapati, R., Xiang, B. Coherence-aware Neural Topic Modeling. Empirical Methods in Natural Language Processing (EMNLP), 2018, 830-836. https://doi.org/10.18653/v1/D18-1096
8. Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., Hon, H. W. Unified Language Model Pre-training for Natural Language Understanding and Generation. Neural Information Processing Systems (NeurIPS), 2019, 32, 13042-13054. https://doi.org/10.48550/arXiv.1905.03197
9. Erkan, G., Radev, D. R. Lexrank: Graph-based Lexical Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research, 2004, 22, 457-479. https://doi.org/10.1613/jair.1523
10. Fattah, M. A. A Hybrid Machine Learning Model for Multi-document Summarization. Applied Intelligence,

2014, 40(4), 592-600. https://doi.org/10.1007/s10489-013-0490-0

11. Fernandes, P., Allamanis, M., Brockschmidt, M. Structured Neural Summarization. International Conference on Learning Representations (ICLR), 2019. https://doi.org/10.48550/arXiv.1811.01824
12. Hermann, K. M., Kociský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P. Teaching Machines to Read and Comprehend. Neural Information Processing Systems (NeurIPS), 2015, 28, 1693-1701.
13. Kingma, D. P., Ba, J. Adam: A Method for Stochastic Optimization. International Conference for Learning Representations, 2015. https://doi.org/10.48550/arXiv.1412.6980
14. Kingma, D. P., Welling, M. Auto-encoding Variational Bayes. International Conference on Learning Representations (ICLR), 2014.
15. Koncel-Kedziorski, R., Bekal, D., Luan, Y., Lapata, M., Hajishirzi, H. Text Generation from Knowledge Graphs with Graph Transformers. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2019, (1), 2284-2293. https://doi.org/10.18653/v1/N19-1238
16. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L. Bart: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. Association for Computational Linguistics (ACL), 2020, 58, 7871-7880. https://doi.org/10.18653/v1/2020.acl-main.703
17. Lin, C. Y. Training a Selection Function for Extraction. Conference on Information and knowledge management (CIKM), 1999, 55-62. https://doi.org/10.1145/319950.319957
18. Linmei, H., Yang, T., Shi, C., Ji, H., Li, X. Heterogeneous Graph Attention Networks for Semi-supervised Short Text Classification. Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, (1), 4821-4830. https://doi.org/10.18653/v1/D19-1488
19. Liu, Y., Lapata, M. Text Summarization with Pretrained Encoders. Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, (1), 3728-3738. https://doi.org/10.18653/v1/D19-1387
20. Liu, Z., Chen, N. Exploiting Discourse-level Segmentation for Extractive Summarization. Workshop on New Frontiers in Summarization, 2019, 2, 116-121. https://doi.org/10.18653/v1/D19-5415
21. Miao, Y., Grefenstette, E., Blunsom, P. Discovering Discrete Latent Topics with Neural Variational Inference. International Conference on Machine Learning (ICML), 2017, 2410-2419. https://doi.org/10.48550/arXiv.1706.00359
22. Miao, Y., Yu, L., Blunsom, P. Neural Variational Inference for Text Processing. International Conference on Machine Learning (ICML), 2016, 1727-1736. https://doi.org/10.48550/arXiv.1511.06038
23. Mihalcea, R., Tarau, P. Textrank: Bringing Order into Text. Empirical Methods in Natural Language Processing (EMNLP), 2004, 404-411.
24. Nallapati, R., Zhai, F., Zhou, B. Summarunner: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. AAAI Conference on Artificial Intelligence, 2017, 3075-3081
25. Narayan, S., Cohen, S. B., Lapata, M. Don't give me the Details, Just the Summary! Topic-aware Convolutional Neural Networks for Extreme Summarization. Empirical Methods in Natural Language Processing (EMNLP), 2018, 1797-1807. https://doi.org/10.18653/v1/D18-1206
26. Qi, W., Yan, Y., Gong, Y., Liu, D., Duan, N., Chen, J., Zhang, R., Zhou, M. Prophetnet: Predicting Future n-gram for Sequence-to-Sequence Pre-training. Empirical Methods in Natural Language Processing (EMNLP), 2020, 2401-2410. https://doi.org/10.18653/v1/2020.findings-emnlp.217
27. See, A., Liu, P. J., Manning, C. D. Get to the Point: Summarization with Pointer-Generator Networks. Association for Computational Linguistics, 2017, 1, 1073-1083. https://doi.org/10.18653/v1/P17-1099
28. Sutskever, I., Vinyals, O., Le, Q. V. Sequence to Sequence Learning with Neural Networks. Neural Information Processing Systems (NeurIPS), 2014, 27, 3104-3112.
29. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. Attention is All You Need. Neural Information Processing Systems (NeurIPS), 2017, 30, 5998-6008. https://doi.org/10.48550/arXiv.1706.03762
30. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y. Graph Attention Networks. Computing Research Repository (CoRR abs), 2017. https://doi.org/10.48550/arXiv.1710.10903
31. Wang, L., Yao, J., Tao, Y., Zhong, L., Liu, W., Du, Q. A Reinforced Topic-aware Convolutional Sequence-to-Sequ-

ence Model for Abstractive Text Summarization. International Joint Conferences on Artificial Intelligence (IJCAI), 2018, 4453-4460. https://doi.org/10.24963/ijcai.2018/619

32. Wang, Z., Duan, Z., Zhang, H., Wang, C., Tian, L., Chen, B., Zhou, M. Friendly Topic Assistant for Transformer Based Abstractive Summarization. Empirical Methods in Natural Language Processing (EMNLP), 2020, 485-497. https://doi.org/10.18653/v1/2020.emnlp-main.35
33. Wang, Z., Wang, C., Zhang, H., Duan, Z., Zhou, M., Chen, B. Learning Dynamic Hierarchical Topic Graph with Graph Convolutional Network for Document Classification. International Conference on Artificial Intelligence and Statistics, 2020, 108, 3959-3969
34. Wei, Y. Document Summarization Method Based on Heterogeneous Graph. Fuzzy Systems and Knowledge Discovery (FSKD), 2012, 1285-1289. https://doi.org/10.1109/FSKD.2012.6234047
35. Xu, J., Durrett, G. Neural Extractive Text Summarization with Syntactic Compression. Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, 3292-3303. https://doi.org/10.18653/v1/D19-1324
36. Xu, J., Gan, Z., Cheng, Y., Liu, J. Discourse-aware Neural Extractive Text Summarization. Association for Computational Linguistics (ACL), 2020, 5021-5031. https://doi.org/10.18653/v1/2020.acl-main.451
37. Yasunaga, M., Zhang, R., Meelu, K., Pareek, A., Srinivasan, K., Radev, D. R. Graph-based Neural Multi-document Summarization. Computational Natural Language Learning (CoNLL), 2017, 21, 452-462. https://doi.org/10.18653/v1/K17-1045
38. Zhang, J., Zhao, Y., Saleh, M., Liu, P. J. Pegasus: Pre-training with Extracted Gap-sentences for Abstractive Summarization. International Conference on Machine Learning (ICML), 2020, 37, 11328-11339.
39. Zheng, C., Zhang, K., Wang, H. J., Fan, L. Topic-aware Abstractive Text Summarization. Computing Research Repository (CoRR abs), 2020. https://doi.org/10.48550/arXiv.2010.10323
40. Zheng, X., Sun, A., Li, J., Muthuswamy, K. Subtopic-driven Multi-document Summarization. Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, (1), 3151-3160. https://doi.org/10.18653/v1/D19-1311
41. Zhou, M, Hannah, L., Dunson, D. B., Carin, L. Beta-Negative Binomial Process and Poisson Factor Analysis. International Conference on Artificial Intelligence and Statistics, 2012, 1462-1471.
42. Zou, Y., Zhao, L., Kang, Y., Lin, J., Peng, M., Jiang, Z, Sun, C., Zhang, Q., Huang, X., Liu, X. Topic-Oriented Spoken Dialogue Summarization for Customer Service with Saliency-aware Topic Modeling. AAAI Conference on Artificial Intelligence, 2021, 35(16), 14665-14673. https://doi.org/10.48550/arXiv.2012.07311