A Survey on Text Summarization Using Graph Neural Network

Rifat Shamsuddin
dept. of CSE
BRAC University,
Dhaka, Bangladesh
rifat.shamsuddin@g.bracu.ac.bd

Md. Abrar Fahim
dept.of CSE
BRAC University,
Dhaka Bangladesh
md.abrar.fahim@g.bracu.ac.bd

Md Tafhim Sadman Islam
dept. of CSE
BRAC University,
Dhaka Bangladesh
md.tafhim.sadman.islam@g.bracu.ac.bd

Md Alkawsar Rahman
dept. of CSE
BRAC University,
Dhaka Bangladesh
md.alkawser.rahman@g.bracu.ac.bd

Jahidul Islam
dept. of CSE
BRAC University,
Dhaka Bangladesh
jahidul.hasan@g.bracu.ac.bd

Md Humaion Kabir Mehedi
dept. of CSE
BRAC University,
Dhaka, Bangladesh
humaion.kabir.mehedi@g.bracu.ac.bd

Md Sabbir Hossain
dept. of CSE
BRAC University,
Dhaka Bangladesh
md.sabbir.hossain1@g.bracu.ac.bd

MD. Mustakin Alam
dept. of CSE
BRAC University,
Dhaka, Bangladesh
md.mustakin.alam@g.bracu.ac.bd

Annajiat Alim Rasel
dept. of CSE
BRAC University,
Dhaka, Bangladesh
annajiat@gmail.com

***Abstract*—Graph Neural Networking (GNN) is an important extension of Deep Learning. Recent development has enabled it to be used in practical applications in the various fields that involve graphs. Natural Language Processing or NLP is one such field that great strides have been made in recent years. There are a large number of NLP related problems that can be expressed in a graph for better understanding and better conceptualization of the problem. Considering the sheer amount of text data available, text summarization has become one of the most important fields of NLP. There has also been extensive research on summarization using graph-based approaches to overcome the limitations of traditional sequence-to-sequence approaches. In this paper we aim to survey recent research related to Text Summarization applied by GNN. We also aim to give a brief overview of recent studies in this field.**

***Index Terms*—Natural Language Processing, Graphical Neural Network, Text Summarization, Abstractive Summarization, Extractive Summarization**

I. Introduction

The part of deep learning known as ”Graph Neural Networking,” or GNN, deals with issues involving graph data or structures that resemble graphs. In order to solve problems involving classification, biochemical graphs, computer vision, graph classification, networking, recommendation, etc., GNN are frequently utilized. [1] Although texts are typically treated as sequences in NLP-related situations, they can also be transformed into graphs or structures that resemble graphs to better depict the problem itself and highlight the relationships between phrases. We have already seen the use of GNN for text classification, text summarization, relation extraction, machine translation, and many other NLP tasks have been the subject of numerous studies. Although unlike other types of data, text is non-Euclidean space data, which makes it more complex for processing. Because encoding and representing text data in this format is easier, most available text processing methods are sequence-to-sequence models. However, encoding text data that way may not highlight the relationship between sentences. Therefore, GNN models can help mitigate the complex task of highlighting the sequence-to-sequence relationships among documents. However, most models are constructed at a statistical level which scarcely captures semantic or syntactic information among graphs.

II. Text Summarization

Getting concise and pertinent information out of a text or document is a fundamental NLP challenge known as text summarization. The primary goal of this work is to draw out the text's important points without sacrificing its intended meaning in the process. It has gotten more difficult to extract pertinent information from the vast amount of textual material that is currently available. Taking into account the possibility that the type of textual data displayed may demonstrate various issues considering its structure and size. There are a couple of simple examples of text summarization which we see almost all the time. Such as, search engines providing a small summary information of the entire web document when we look up certain topics. In this case it shows relevant information to our search topics [2]. Furthermore, there are news articles that generate a preview of the news article as its headlines.

Moreover, the path of the investigation of making a text summary model can change based solely on the type of incoming data. For instance, single document summarization concentrates on a single type of document, whereas multiple document summarization concentrates on a collection of papers written from several points of view.

Due to redundancy, the amount of text, encoder limitations, etc., multiple document summarization is more difficult than summarizing a single document. Additionally, the approaches differ based on the summarizing algorithm used (supervised or unsupervised), the language restrictions (monolingual or multilingual), the degree of summary being constructed (sentence, headline, or entire summary), etc. However, all of them fall into one of two main categories depending on how text summarization is approached: extractive, where the most important text from the document is extracted and grouped to give a summary; and abstractive, where the salient information in the document is identified to produce a clear summary of the document. Since, abstractive text summary generates text using only the extracted ideas from the text it is a more complicated approach for text summarization. Moreover, there can be specialized text summarization techniques, depending on the type of input text. For example, we cannot utilize the normal methods for text summarization for texts structured like a conversation. Moreover, the type of topic the text document is about can also affect this.

There are already multiple proposed extractive and abstractive summarization techniques that are available. One of the oldest techniques is the feature based text summarization, which uses features like linguistic and statistical to highlight the importance of a sentence in a document. This can be features like the length of the sentences, frequency term, title-feature, term weight, thematic words, TF-IDF etc. [3], [4]. Relevance Measure based methods uses relevance score of the sentences to rank sentences [5]. Topic Based Methods finds the key sentences for the summary using the distribution of topic words in the input document [6], [7]. Latent Semantic Analysis methods uses latent semantic indexing to analyse the relationship between documents and similar types of terms they that occur in sentences. In this method singular value decomposition is utilized to reduce the dimension of the sentence vector while preserving the similarity structure [5], [8]. Then there are neural networks and machine learning based approaches which uses and encoder-decoder type models to create summaries [9]. Moreover, the recent popular neural networks are, encoder-decoder type RNN models [10], [11], query based summarization system AttSum [12]. And finally, there are graph-based methods, which uses various graph-based algorithms to extract the information from graph-like representation of the data. Depending on the graph type, each graph highlights different features among documents. This can easily help in highlighting the semantic and syntactic relations among sentences.

Graph Based Text Summarization

In graph based text summarization approach, words, sentences, and sometimes even the entire document are treated as nodes. The edges connect the nodes represent their semantic relationship between them. This relationship may also vary depending on what type of relation the graph is trying to portray. This is one of the main reasons why graph based approaches are favored, since it gives a visualization of the nodes and edges that helps in highlighting the relationship among the nodes. This can help in detecting sentences that are important for the overall document.

There have been multiple studies focusing on text summarization using GNN or graph based methods. Most of them are either extractive or abstractive. Moreover, the studies also vary from one another according to the type of graphs that are being utilized. The earliest mentioned graph-based model is PageRank, which is an algorithm for managing interconnectivity between web pages. Further on, methods like TextRank [13] and LexRank were developed which uses cosine similarity to build a connectivity graph of the sentences [14]. For Multi Document Summarization there is DivRank, which is a ranking algorithm for balancing the prestige and diversity of vertices [15]. There is also an abstractive summarization method which uses a graph-based attention mechanism for keeping track of the encoded text [16]. Later on, models like Graph Convolution Network (GCN) [17] and Graph Attention Network (GAT) gained popularity due to its simplicity [18]. Moreover, a GAT based transformer encoder using knowledge graphs [19]. Recent proposed models such as CGSUM, BASS, PLANSUM, SSN-DM, etc. We will go through each of the models in the later portion of our survey.

III. Related Works

We have selected 8 proposed models in total relating to text summarization using GNN. In this section we will give a brief overview of each of their methodology and their results.

There are basically two approaches to text summarization: extractive and abstractive. But these approaches may even differ considering what type of documents the data is being extracted. As for graph-based methods, the process can also differ from one another according to the graph structure that is being used to portray the documents. The proposed models we studied for our survey all share some common structure. Most research concerning single documents is mainly focused on extracting cross-sentence relationships in long-form single documents. This is then extended to apply on multi document summarization as they share somewhat similar methodology. The models we chose for our survey are about single document summarization are all about summarizing the long-form documents.

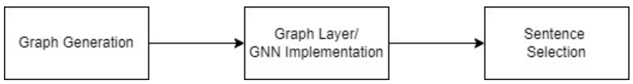

Fig. 1. Extractive Summarization

The methodology for extractive summarization using GNN can be divided into three phases: graph generation, graph layer, and sentence selector. For text encoding, most models use encoders like pre-trained language models, transformers, RNN, LSTM, BERT, or CNN. Most models use RNN, LSTM, or BERT to capture cross-sentence relations utilizing their capability of capturing long range dependencies. [11].

Although RNN and LSTM are used for embedding long distance dependencies, they have their limitations and do not have long distance dependency for something like long-form documents or multi documents. And most models that uses attention based transformers perform better than LSTM [20]. BERT has a limit of 512 tokens. One way of bypassing this limit is truncating the document. However, for summarization this is not feasible as it might cause loss of gold sentences. While segmenting the documents and extracting local summaries might not be accurate with the entire document. Graphs can be used to overcome this problem of modeling long distance dependency among documents. A hierarchical network can be used with discourse graphs into the encoder model to get the global context. Moreover, heterogeneous graphs with discourse and conference relations can help overcome this problem [21].

For long form documents one of the proposed methods utilizes a memory mechanism and a sliding window to process long documents sequentially. To overcome the limitation of BERT a sliding encoder is applied to process the document in multiple windows sequentially while a memory module stores important information it learns from the previous windows. Positional information of each of the windows is used to capture discourse information [22]. In another proposed method, the CNN encoder is used for extracting local n-gram features and BiLSTM for the global feature. The TF-IDF is used as the weight of the edges [23]. Similarly, knowledge graphs on triples (Subject, Verb and Object) can capture various features like position, length, frequency etc. which can be used for extracting important sentences in a document [24]. For MDS, one method takes a multi-document as input and represents all sentences in a graph structure where each of the sub graphs act as potential summaries. Here the document acts as a relation graph of the sentences [25]. Moreover, most of the long document summarization methods can be applied on multi documents as well as both processing both of these types of documents for summarization face the same problems. For example, the SSN-DM model can also be used for MDS by using the relation of document-word [22]. One model used hierarchical transformer encoded documents and used attention based fully connected graph relations and using similarity or discourse relations [26]. Another method uses structure of the scientific text as most related texts have similar citations. Taking that into consideration using a citation graph can help in extracting summaries [27].

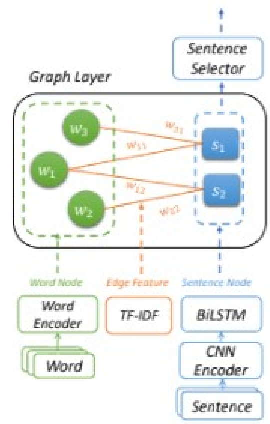

Fig. 2. Heterogenous Graph Model [23]

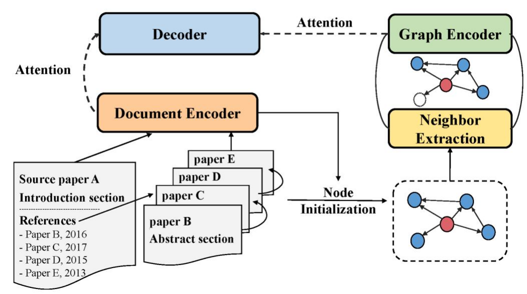

Fig. 3. CGSUM Model [27]

Most models utilize GCN and GAT in their graph layers for embedding graph information in their text documents. Positional information for each of the windows is used to capture discourse information. The memory module with the use of GAT is encoded on the graph [22]. Similarly, GAT can also be used for updating word nodes and sentence nodes in heterogeneous graph models. [23]. In this phase GNN are used for updating the word, sentence or document representation using semantic, discourse, knowledge, action, etc. graphs for enhancing their representation. The SgSum model used a graph-informed attention mechanism for highlighting the pairwise relations [25].

After the graph layer, suitable sentences are then selected for summary. Most section selection is done using summary labeling. In our survey there are few ways of summary labeling. It can be done using Multi Layered Perceptron classification and the sentences are then ranked for choosing the summary. The summary labels are classified using MLP or Multi Layered

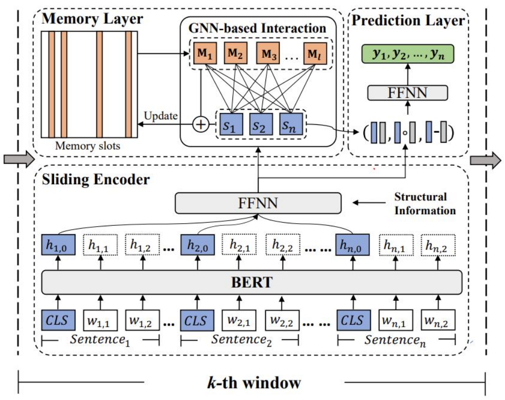

Fig. 4. SSN-DM Model [22]

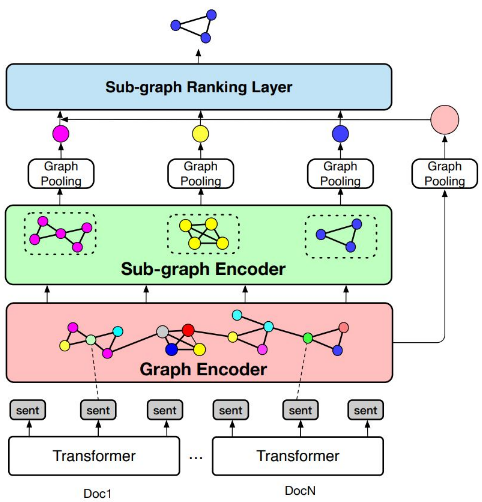

Fig. 5. SgSUM Model [25]

Perceptron classification. The sentences of the entire document are then ranked and the top-ranked sentences are chosen for the final summary [22]. For some model decoders like Trigram blocking used for discard overlapping trigrams [23] which are redundant information [28]. The basic idea is that a gold summary has the highest matching score to the source document, and a good candidate summary should obtain a high score. For the SgSum model, the whole document is turned into a graph. Similar nodes are redundant while the disconnected nodes are not connected. A similar type of decoder has been used for subgraph extraction. Another method for extracting text is the greedy method which is used for SgSum and CGSUM models [25], [27].

As for abstractive summarization, three papers we are going to highlight. All the overall models are somewhat similar to extracting beside the sentence selection phase. Since the main difference between methodologies of the two types of summarization falls in the last step. In extractive summarization, it utilizes prediction or ranking layer to extract the best parts for making the summary. While for abstractive summarization, it has to generate texts so most models utilize a decoder similar to the extractive models. One model, PLANSUM, used encoder BiLSTM and decoder LSTM with an attention mechanism. In its model it utilized a fusion module to aggregate and reduce tokens which is passed down to decoders for the sentence generation [29]. Similar to the previous methods, Global Aware Beam Search also uses attention tokens for attaining global aware inference. Moreover, this model uses beam search based on the predicted global attention distribution. As for the decoder it used pre trained BERT and PEGASUS. [30]. Lastly, one paper proposes a structure-aware model that uses discourse and action graphs. This model is mainly made for texts with complex structures like conversations and for keeping track of utterances and who is doing what. They proposed the method of using utterances and action graphs. For its decoder, it uses a multi-granularity decoder equipped with action, discourse, and utterance graphs, which is a modified BERT transformer decoder [31].

TABLE I
SURVEY TABLE

References	Year	Summary Type	Model
[31]	2021	Abstractive Summary	S-BERT with discourse and utterance graph
[22]	2021	Extractive Summary	SSN-DM
[23]	2020	Extractive Summary	CNN and BiLSTM
[24]	2020	Extractive Summary	T-LSTM, S-LSTM, NN, SVM, SVR, PKG, UKG, FKG
[27]	2021	Extractive Summary	CGSUM
[25]	2021	Extractive Summary	SgSum
[30]	2021	Abstractive Summary	Global Aware Beam Search
[29]	2021	Abstractive Summary	PLANSUM

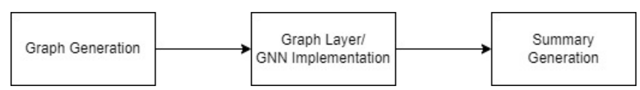

Fig. 6. Abstractive Summarization

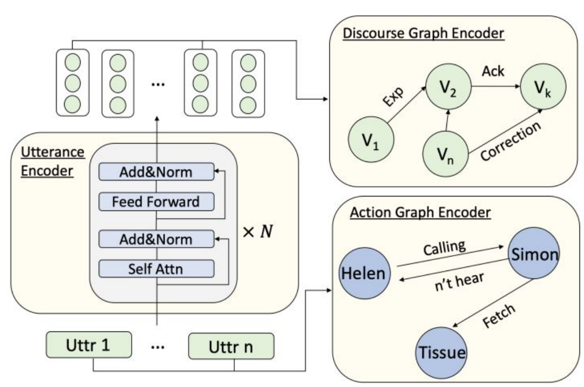

Fig. 7. S-BERT Encoder discourse and utterance graph

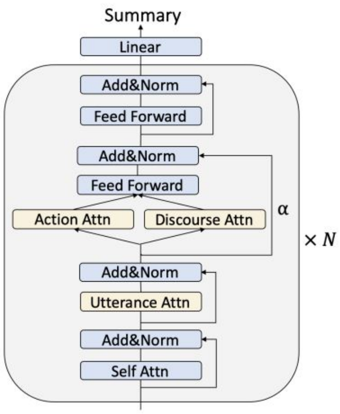

Fig. 8. S-BERT Multi Granularity Decoder using discourse and utterance graph

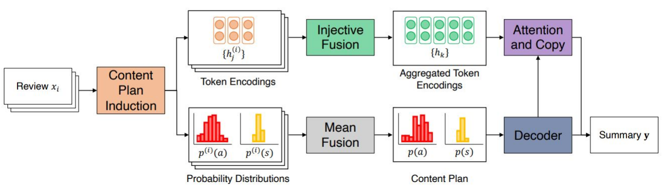

Fig. 9. PLANSUM Model

IV. Results

Most of the models highlighted utilize somewhat the same structure. However, they differ from the type of graphs that are being used for highlighting sentence/document relationships. Most models utilized BERT and LSTM. However, there are more improved versions of BERT that can give better results than the base BERT model. Like Span BERT, which performs at a span level [32] and RoBERTa, which in general performs better than bert. And as for graphs, the ones that are being utilized to highlight global context action and discourse relation are the most widely used. Some approaches used LSTM to resolve the question of slope vanishing. Almost all the proposed papers in our survey utilized GAT to encode the graph using necessary embeddings. And most of these models suffer from long range dependencies which is why they utilized an attention based mechanism to overcome that problem. Each of the methods have their own advantages and disadvantages. From our survey, it is noticeable that from the selected papers abstractive summarization fared better according to their represented ROGUE metrics. Metrics like, ROUGE1, ROUGE2, and ROUGE-L have been used for evaluating the quality of generated summaries. Moreover, the models were compared their results with contemporary GNN models and they outperformed all of them. If we compare the papers among them, it might not give a correct interpretation since each of the methods create certain problems in mind. For extractive HETEROSUMGRAPH yielded better results. However, most of these methods suffer from high computational cost because of the complexity of constructing graphs.

V. Conclusion & Future Scope

A recent increase in the volume of text data and the impossibility of reading it all word for word has increased the significance of the summary process. A variety of conclusions can be drawn from the analysis of the papers examined. The majority of models that produced better outcomes were built on hierarchical heterogeneous graphs. Additionally, they used attention based graphs to emphasize the connections between the documents. Although deep learning algorithms produce greater outcomes, the limits of the encoder still exist. Despite the fact that extractive summarization is easier than abstractive, The appeal of abstraction is more widespread than that of the latter. We were unable to emphasize the full scope of abstractive summarization because we only examined a small number of papers on that topic. Since only a small number of papers on this topic has been covered, extensive and more an in-depth overview was not possible. However, we were able to highlight similar information and methods that have been used among the proposed methods. It is possible to expand the scope of future research and improve the methods of text summarization from this observation.

References

[1] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[2] T. Steiner, R. Troncy, and M. Hausenblas, "How google is using linked data today and vision for tomorrow," in *Proceedings of Linked Data in the Future Internet at the Future Internet Assembly (FIA 2010), Ghent, December 2010*, 2010.

[3] P. B. Baxendale, "Machine-made index for technical literature—an experiment," *IBM Journal of research and development*, vol. 2, no. 4, pp. 354–361, 1958.

[4] P. Gupta, V. S. Pendluri, and I. Vats, "Summarizing text by ranking text units according to shallow linguistic features," in *13th International Conference on Advanced Communication Technology (ICACT2011)*, pp. 1620–1625, IEEE, 2011.

[5] Y. Gong and X. Liu, "Generic text summarization using relevance measure and latent semantic analysis," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 19–25, 2001.

[6] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of research and development*, vol. 2, no. 2, pp. 159–165, 1958.

[7] M.-T. Nguyen, V. C. Tran, X. H. Nguyen, and L.-M. Nguyen, "Web document summarization by exploiting social context with matrix co-factorization," *Information Processing & Management*, vol. 56, no. 3, pp. 495–515, 2019.

[8] J. Steinberger, K. Jezek, *et al.*, "Using latent semantic analysis in text summarization and summary evaluation," *Proc. ISIM*, vol. 4, no. 93-100, p. 8, 2004.

[9] M. Yousefi-Azar and L. Hamey, "Text summarization using unsupervised deep learning," *Expert Systems with Applications*, vol. 68, pp. 93–105, 2017.

[10] L. Wang and W. Ling, "Neural network-based abstract generation for opinions and arguments," *arXiv preprint arXiv:1606.02785*, 2016.

[11] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," *arXiv preprint arXiv:1603.07252*, 2016.

[12] Z. Cao, W. Li, S. Li, F. Wei, and Y. Li, "Attsum: Joint learning of focusing and summarization with neural attention," *arXiv preprint arXiv:1604.00125*, 2016.

[13] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404–411, 2004.

[14] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of artificial intelligence research*, vol. 22, pp. 457–479, 2004.

[15] Q. Mei, J. Guo, and D. Radev, "Divrank: the interplay of prestige and diversity in information networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1009–1018, 2010.

[16] J. Tan, X. Wan, and J. Xiao, "Abstractive document summarization with a graph-based attentional neural model," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1171–1181, 2017.

[17] M. Yasunaga, R. Zhang, K. Meelu, A. Pareek, K. Srinivasan, and D. Radev, "Graph-based neural multi-document summarization," *arXiv preprint arXiv:1706.06681*, 2017.

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

[19] R. Koncel-Kedziorski, D. Bekal, Y. Luan, M. Lapata, and H. Hajishirzi, "Text generation from knowledge graphs with graph transformers," *arXiv preprint arXiv:1904.02342*, 2019.

[20] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.

[21] Y. J. Huang and S. Kurohashi, "Extractive summarization considering discourse and coreference relations based on heterogeneous graph," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 3046–3052, 2021.

[22] P. Cui and L. Hu, "Sliding selector network with dynamic memory for extractive summarization of long documents," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5881–5891, 2021.

[23] D. Wang, P. Liu, Y. Zheng, X. Qiu, and X. Huang, "Heterogeneous graph neural networks for extractive document summarization," *arXiv preprint arXiv:2004.12393*, 2020.

[24] A. Vhatkar, P. Bhattacharyya, and K. Arya, "Knowledge graph and deep neural network for extractive text summarization by utilizing triples," in *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pp. 130–136, 2020.

[25] M. Chen, W. Li, J. Liu, X. Xiao, H. Wu, and H. Wang, "Sgsum: Transforming multi-document summarization into sub-graph selection," *arXiv preprint arXiv:2110.12645*, 2021.

[26] Y. Liu and M. Lapata, "Hierarchical transformers for multi-document summarization," *arXiv preprint arXiv:1905.13164*, 2019.

[27] C. An, M. Zhong, Y. Chen, D. Wang, X. Qiu, and X. Huang, "Enhancing scientific papers summarization with citation graph," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 12498–12506, 2021.

[28] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," *arXiv preprint arXiv:1705.04304*, 2017.

[29] R. K. Amplayo, S. Angelidis, and M. Lapata, "Unsupervised opinion summarization with content planning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 12489–12497, 2021.

[30] Y. Ma, Z. Lan, L. Zong, and K. Huang, "Global-aware beam search for neural abstractive summarization," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16545–16557, 2021.

[31] J. Chen and D. Yang, "Structure-aware abstractive conversation summarization via discourse and action graphs," *arXiv preprint arXiv:2104.08400*, 2021.

[32] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: Improving pre-training by representing and predicting spans," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020.