Multiplex Graph Neural Network for Extractive Text Summarization

Baoyu Jing[†], **Zeyu You**[‡], **Tao Yang**[‡], **Wei Fan**[‡] and **Hanghang Tong**[†]
[†]University of Illinois at Urbana-Champaign
[‡]Tencent America
{baoyuj2, htong}@illinois.edu
{zeyuyou, tytaoyang, davidwfan}@tencent.com

Abstract

Extractive text summarization aims at extracting the most representative sentences from a given document as its summary. To extract a good summary from a long text document, sentence embedding plays an important role. Recent studies have leveraged graph neural networks to capture the inter-sentential relationship (e.g., the discourse graph) to learn contextual sentence embedding. However, those approaches neither consider multiple types of inter-sentential relationships (e.g., semantic similarity & natural connection), nor model intra-sentential relationships (e.g, semantic & syntactic relationship among words). To address these problems, we propose a novel Multiplex Graph Convolutional Network (Multi-GCN) to jointly model different types of relationships among sentences and words. Based on Multi-GCN, we propose a Multiplex Graph Summarization (Multi-GraS) model for extractive text summarization. Finally, we evaluate the proposed models on the CNN/DailyMail benchmark dataset to demonstrate the effectiveness of our method.

1 Introduction

Numerous documents from a variety of sources are uploaded to the Internet or database everyday, such as news articles (Hermann et al., 2015), scientific papers (Qazvinian and Radev, 2008) and electronic health records (Jing et al., 2019). How to effectively digest the overwhelming information has always been a fundamental question in natural language processing (Nenkova and McKeown, 2011). This question has sparked the research interests in the task of extractive text summarization, which aims to generate a short summary of a document by extracting the most representative sentences from it.

Most of the recent methods (Cheng and Lapata, 2016; Narayan et al., 2018; Luo et al., 2019; Wang et al., 2020a; Mendes et al., 2019; Zhou et al., 2018) formulate the task of extractive text summarization as a sequence labeling task, where the labels indicate whether a sentence should be included in the summary. To extract sentence features, existing approaches generally use Recurrent Neural Networks (RNN) (Yasunaga et al., 2017; Nallapati et al., 2017; Zhou et al., 2018; Mendes et al., 2019; Luo et al., 2019; Cheng and Lapata, 2016), Convolutional Neural Networks (CNN) (Cheng and Lapata, 2016; Luo et al., 2019; Narayan et al., 2018) or Transformers (Zhong et al., 2019; Liu and Lapata, 2019a). Endeavors have been made to develop models to capture various sentence-level relations. Early studies, such as LexRank (Radev, 2004) and TextRank (Mihalcea and Tarau, 2004), built similarity graphs among sentences and leverage PageRank (Page et al., 1999) to score them. Later, graph neural networks e.g., Graph Convolutional Network (GCN) (Kipf and Welling, 2016) have been adopted on various inter-sentential graphs, such as the approximate discourse graph (Yasunaga et al., 2017), the discourse graph (Xu et al., 2020) and the bipartite graph between sentences and words (Wang et al., 2020a; Jia et al., 2020).

Albeit the effectiveness of the existing methods, there are still two under-explored problems. Firstly, the constructed graphs of the existing studies only involve one type of edges, while sentences are often associated with each other via multiple types of relationships (referred to as the *multiplex graph* in the literature (De Domenico et al., 2013; Jing et al., 2021a)). Two sentences with some common keywords are considered to be naturally connected (we refer this type of graph as the *natural connection graph*). For example, in Figure 1, the first and the last sentence exhibit a natural connection (green) via the shared keyword "City". Although the two sentences are far away from each other, they can be jointly considered as part of the summary since the entire document is about the keyword "City". However, such a relation can barely

be captured by traditional encoders such as RNN and CNN. Two sentences sharing similar meanings are also considered to be connected (we refer this type of graph as the *semantic graph*). In Figure 1, the second and the third sentence are semantically similar since they express a similar meaning (yellow). The semantic similarity graph maps the semantically similar sentences into the same cluster and thus helps the model to select sentences from different clusters, which could improve the coverage of the summary. Different relationships provide relational information from different aspects, and jointly modeling different types of edges will improve model's performance (Wang et al., 2019; Park et al., 2020; Jing et al., 2021b; Yan et al., 2021; Jing et al., 2021c). Secondly, the aforementioned methods fall short in taking advantage of the valuable relational information among words. Both of the syntactic relationship (Tai et al., 2015; He et al., 2017) and the semantic relationship among words (Kenter and De Rijke, 2015; Wang et al., 2020b; Varelas et al., 2005; Wang et al., 2021; Radev et al., 2004) have been proven to be useful for the downstream tasks, such as text classification (Kenter and De Rijke, 2015; Jing et al., 2018), information retrieval (Varelas et al., 2005) and text summarization (Radev et al., 2004).

We summarize our contributions as follows:

- To exploit multiple types of relationships among sentences and words, we propose a novel Multiplex Graph Convolutional Network (Multi-GCN).
- Based on Multi-GCN, we propose a Multiplex Graph based Summarization (Multi-GraS) framework for extractive text summarization.
- We evaluate our approach and competing methods on the CNN/DailyMail benchmark dataset and the results demonstrate our models' effectiveness and superiority.

2 Methodology

We first present Multi-GCN to jointly model different relations, and then present the Multi-GraS approach for extractive text summarization.

2.1 Multiplex Graph Convolutional Network

Figure 2c illustrates Multi-GCN over a multiplex graph with initial node embedding $\mathbf{X}$ and a set of relations $\mathcal{R}$. Firstly, Multi-GCN learns node embeddings $\mathbf{H}_r$ of different relations $r \in \mathcal{R}$ separately, and then combines them to produce the final

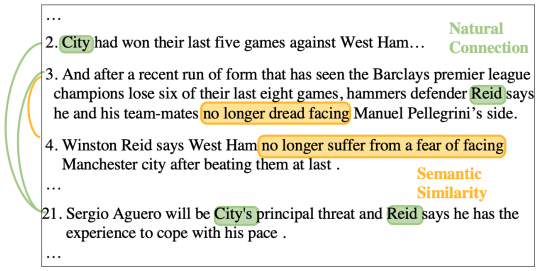


Figure 1: An example document: There are two different relationships among sentences: the semantic similarity (yellow) and the natural connection (green). Sentences 2, 3, 21 are the oracle sentences.

embedding $\mathbf{H}$. Secondly, Multi-GCN employs two types of skip connections, the inner and the outer skip-connections, to mitigate the over-smoothing (Li et al., 2018) and the vanishing gradient problems of the original GCN (Kipf and Welling, 2016).

More specifically, we propose a Skip-GCN with an inner skip connection to extract the embeddings $\mathbf{H}_r$ for each relation. The updating functions for the l-th layer of the Skip-GCN are defined as:

$$\hat{\mathbf{H}}_r^{(l)} = \text{GCN}_r^{(l)}(\mathbf{A}_r, \mathbf{H}_r^{(l-1)}) + \mathbf{H}_r^{(l-1)} \quad (1)$$

$$\mathbf{H}_r^{(l)} = \text{ReLU}(\hat{\mathbf{H}}_r^{(l)}\mathbf{W}_r^{(l)} + \mathbf{b}_r^{(l)}) \quad (2)$$

where $\mathbf{A}_r$ is the adjacency matrix for the relation r; $\mathbf{W}_r^{(l)}$ and $\mathbf{b}_r^{(l)}$ denote the weight and bias. Note that $\mathbf{H}_r^{(0)} = \mathbf{X}$ is the initial embedding, and $\mathbf{H}_r$ is the output after all Skip-GCN layers.

Next, we combine the embedding of different relations $\{\mathbf{H}_r\}_{r\in\mathcal{R}}$ by the following equations:

$$\mathbf{H} = \tanh(\text{cat}(\{\mathbf{H}_r\}_{r\in\mathcal{R}})\mathbf{W} + \mathbf{b}) \quad (3)$$

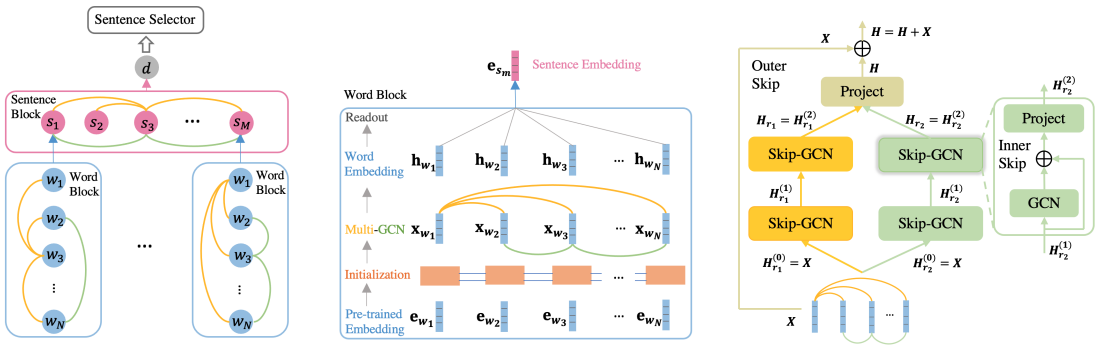
where cat denotes the concatenation operation and $\mathbf{W}$ and $\mathbf{b}$ denote the weight and bias of the project block in Figure 2c.

Finally, we use an outer skip connection to directly connect $\mathbf{X}$ with $\mathbf{H}$:

$$\mathbf{H} = \mathbf{H} + \mathbf{X} \quad (4)$$

2.2 The Multi-GraS model

The overview of the proposed Multi-GraS is illustrated in Figure 2a. Multi-GraS is comprised of three major components: the word block, the sentence block, and the sentence selector. The word block and the sentence block share a similar "Initialization – Multi-GCN – Readout" structure to extract the sentence and document embeddings. The sentence selector picks the most representative sentences as the summary based on the extracted embeddings.



(a) Multi-GraS (b) Word Block (c) Multi-GCN

Figure 2: Overview of the proposed Multi-GraS, the word block and Multi-GCN.

2.2.1 The Word Block

The architecture of a word block is illustrated in Figure 2b. Given a sentence s_m with N words $\{w_n\}_{n=1}^N$, the word block takes the pre-trained word embeddings $\{\mathbf{e}_{w_n}\}_{n=1}^N$ as inputs, and produces the sentence embedding $\mathbf{e}_{s_m}$. Specifically, the **Initialization** module produces contextualized word embeddings $\{\mathbf{x}_{w_n}\}_{n=1}^N$ via Bi-LSTM. The **Multi-GCN** module jointly captures multiple relations for $\{\mathbf{x}_{w_n}\}_{n=1}^N$ and produces $\{\mathbf{h}_{w_n}\}_{n=1}^N$. The **Readout** module produces the sentence embedding $\mathbf{e}_{s_m}$ based on max pooling over $\{\mathbf{h}_{w_n}\}_{n=1}^N$.

In this paper, we jointly consider the syntactic and semantic relation among words. For the syntactic relation, we use a dependency parser to construct a syntactic graph $\mathbf{A}_{syn}$: if a word w_n and another word $w_{n'}$ has a dependence link between them, then $\mathbf{A}_{syn}[n, n'] = 1$, otherwise $\mathbf{A}_{syn}[n, n'] = 0$. For the semantic relation, we use the absolute value of dot product between the embeddings of words to construct the graph: $\mathbf{A}_{sem_w}[n, n'] = |\mathbf{x}_{w_n}^T \cdot \mathbf{x}_{w_{n'}}|$. Note that we use the absolute value since GCN (Kipf and Welling, 2016) requires the values in the adjacency matrix to be non-negative.

2.2.2 The Sentence Block

Given a document with M sentences $\{s_m\}_{m=1}^M$, the sentence block takes the sentence embeddings $\{\mathbf{e}_{s_m}\}_{m=1}^M$ as inputs, and generates a document embedding $\mathbf{d}$ through a Bi-LSTM, a Multi-GCN and a pooling module. Essentially, the architecture of the sentence block resembles the word block, thus we only elaborate the construction of graphs for sentences.

In this paper, we consider the natural connection and the semantic relation between sentences. The semantic similarity between s_m and s'_m is the absolute value of dot product between $\mathbf{x}_{s_m}$ and $\mathbf{x}_{s_{m'}}$, and thus the semantic similarity graph $\mathbf{A}_{sem_s}$ can be constructed by $\mathbf{A}_{sem_s}[m, m'] = |\mathbf{x}_{s_m}^T \cdot \mathbf{x}_{s_{m'}}|$. For the natural connection, if two sentences share a common keyword, then we consider they are naturally connected. Such a relation helps to cover more sections of a document by connecting faraway sentences (not necessarily semantic similar) via their shared keywords, as shown in Figure 1.

$$\mathbf{A}_{nat}[m, m'] = \sum_{w \in \mathcal{W}} \text{tfidf}_{(s_m, w)} \cdot \text{tfidf}_{(s_{m'}, w)}, \quad (5)$$

where $\text{tfidf}_{(s_m,w)}$ is the tfidf score of the keyword w within s_m; $\mathcal{W}$ is the set of keywords.

2.2.3 Sentence Selector

The sentence selector first scores the sentences $\{s_m\}_{m=1}^M$ and then selects the top-K sentences as the summary. The model design for scoring the sentences follows the human reading comprehension strategy (Pressley and Afflerbach, 1995; Luo et al., 2019), which contains reading and post-reading processes. The reading process extracts rough meaning of s_m:

$$\mathbf{o}_m = \tanh(\mathbf{W}_{Reading}\mathbf{h}_{s_m} + \mathbf{b}_{Reading}). \quad (6)$$

The post-reading process further captures the auxiliary contextual information – document embedding $\mathbf{e}_d$ and the initial sentence embedding $\mathbf{e}_{s_m}$:

$$\mathbf{o}_m = \tanh(\mathbf{W}_{Post}[\mathbf{o}_m, \mathbf{e}_d, \mathbf{e}_{s_m}] + \mathbf{b}_{Post}). \quad (7)$$

The final score for s_m is given by:

$$p_m = \sigma(\mathbf{W}_p\mathbf{o}_m + \mathbf{b}_p), \quad (8)$$

where $\sigma()$ denotes the sigmoid activation.

When ranking the sentences $\{s_m\}_{m=1}^M$, we follow (Paulus et al., 2018; Liu and Lapata, 2019b; Wang et al., 2020a) and use the tri-gram blocking technique to reduce the redundancy.

3 Experiments

3.1 Experimental Setup

3.1.1 Datasets

We evaluate our propose model on the benchmark CNN/DailMail (Hermann et al., 2015) dataset. This dataset is a combination of the CNN and DailyMail datasets, which contains $287,227$, $13,368$ and $11,490$ articles for training, validating and testing respectively.

For the DailyMail dataset (Hermann et al., 2015), the news articles were collected from the DailyMail website. Each article contains a story and highlights, and the story and highlights are treated as the document and the summary respectively. The dataset contains $219,506$ articles, which is split into $196,961/12,148/10,397$ for training, validating and testing.

For the CNN dataset (Hermann et al., 2015), the news articles were collected from the CNN website. Each article is comprised of a story and highlights, where the story is treated as the document and highlights are considered as the summary. The CNN dataset contains $92,579$ articles in total, $90,266$ are used for training, $1,220$ for validation and $1,093$ for testing.

3.1.2 Comparison Methods

For the task of extractive text summarization, we compare the proposed Multi-GraS method with the following methods in three categories: (1) deep learning based methods: NN-SE (Cheng and Lapata, 2016), LATENT (Zhang et al., 2018), NeuSUM (Zhou et al., 2018), JECS (Xu and Durrett, 2019) and EXCONSUMM$_{\text{Extractuve}}$ (Mendes et al., 2019); (2) reinforcement-learning based methods: REFRESH (Narayan et al., 2018), BanditSum (Dong et al., 2018), LSTM+PN+RL (Zhong et al., 2019) and HER (Luo et al., 2019); (3) graph based methods: TextRank (Mihalcea and Tarau, 2004) and HSG (Wang et al., 2020a).

3.2 Implementation Details

The vocabulary size is fixed as $50,000$ and the pre-trained Glove embeddings (Pennington et al., 2014) are used for the input word embeddings. For both of the word block and the sentence block, the Initialization modules employ two-layer Bi-LSTMs. The Multi-GCN modules use two-layer Skip-GCNs. We fix all the hidden dimensions as 300. We use the Stanford CoreNLP(Manning et al., 2014) to extract syntactic graphs. For the natural connection graphs, we filter out the stop words, punctuation, and the words whose document frequency is less than 100. During training, we use the Adam optimizer (Kingma and Ba, 2014), and the learning rates for CNN, DailyMail, and CNN/DailyMail datasets are set to be 0.0001, 0.0005, and 0.0005, respectively. When generating summaries, we select the top-2 and top-3 sentences for the CNN and DailyMail datasets, respectively.

Methods	R-1	R-2	R-L
NN-SE	35.50	14.70	32.20
LATENT	41.05	18.77	37.54
NeuSUM	41.59	19.01	37.98
JECS	41.70	18.50	37.90
EXCONSUMM$_{\text{Extractive}}$	41.70	18.60	37.80
REFRESH	40.00	18.20	36.60
BanditSum	41.50	18.70	37.60
LSTM+PN+RL	41.85	18.93	38.13
HER	42.30	18.90	37.60
TextRank	40.20	17.56	36.44
HSG	42.95	19.76	39.23
Multi-GraS	**43.16**	**20.14**	**39.49**

Table 1: ROUGE scores of different methods.

3.2.1 Oracle Label Extraction

The summaries of the documents are the highlights of the news written by human experts, hence the sentence-level labels are not provided. Given a document and its summary, we follow (Wang et al., 2020a; Liu and Lapata, 2019b; Mendes et al., 2019; Narayan et al., 2018) to identify the set of sentences (or oracles) of the document which has the highest ROUGE scores with respect to its summary.

3.2.2 Evaluation Metrics

We evaluate the quality of the summarization by the ROUGE scores (Lin, 2004), including R-1, R-2 and R-L for calculating the unigram, bigram and the longest common sub-sequence overlapping between the generated summarization and the ground-truth summary. In addition to automatic evaluation via ROUGE, we follow (Luo et al., 2019; Wu and Hu, 2018) and conduct human evaluation to score the quality of the generated summaries.

3.3 Overall Performance

The ROUGE (Lin, 2004) scores of all comparison methods are presented in Table 1. Within baseline methods, HSG achieves the highest performance, which indicates that considering graph structures could improve performance. We also observe that Multi-GraS outperforms all of the comparison methods and it achieves $0.21/0.38/0.26$ performance increase on R-1/R-2/R-L scores.

Methods	R-1	R-2	R-L
Multi-GraS	**43.16**	**20.14**	**39.49**
- trigram blocking	42.15	19.62	38.55
- contextual information	43.12	20.04	39.18
- outer skip	43.08	20.08	39.44
- inner skip	42.95	20.05	39.33
- semantic relation	43.03	20.10	39.40
- natural connection relation	42.74	19.80	39.14
- weights for natural connection	42.54	19.67	38.92
Multi-GraS$_{\text{word}}$	42.67	19.80	39.06
- outer skip	42.44	19.52	38.81
- inner skip	42.64	19.76	39.04
- semantic relation	42.63	19.68	39.02
- syntactic relation	42.42	19.57	38.82
LSTM	42.35	19.51	38.73
LSTM (w/o tri-gram blocking)	41.55	19.14	37.98
Transformer (w/o tri-gram blocking)	41.33	18.83	37.65

Table 2: Ablation Study.

3.4 Ablation Study

Firstly, as shown in Table 2, tri-gram blocking and contextual information within the sentence selector help improve model's performance.

Then we study the influence of the Multi-GCN within the sentence block and the word block separately. To do so, we remove the Multi-GCN within the sentence block (Multi-GraS$_{\text{word}}$) and further remove the Multi-GCN within the word block (LSTM). By comparing LSTM, Multi-GraS$_{\text{word}}$ and Multi-GraS, it can be observed that Multi-GCN in both sentence and word blocks significantly improve the performance. Next, we study the influence of the components within Multi-GCN. Table 2 indicates that the inner and outer skip connections play an important role in Multi-GCN. Besides, jointly considering different relations is always better than considering one relation alone.

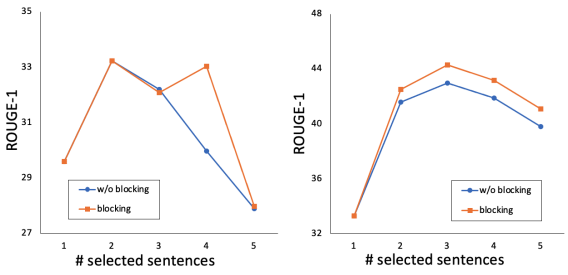
Finally, for the Initialization module in the word and sentence blocks, LSTM performs better than Transformer (Vaswani et al., 2017).

3.5 Human Evaluation

We randomly select 50 documents along with the summaries obtained by HSG, Multi-GraS, Multi-GraS$_{\text{word}}$, LSTM as well as the oracle summaries. Three volunteers (proficiency in English) rank the summaries from 1 to 5 in terms of the overall quality, coverage and non-redundancy. The human evaluation results are presented in Table 3: oracle ranks the highest, Multi-GraS ranks higher than HSG.

Methods	Overall	Coverage	Non-Redundancy
LSTM	3.07	2.97	2.77
Multi-GraS$_{\text{word}}$	2.97	2.93	2.77
HSG	2.87	2.87	2.67
Multi-GraS	2.20	2.23	2.13
Oracle	1.70	1.57	1.57

Table 3: Human evaluation: the lower the better.



(a) CNN (b) DailyMail

Figure 3: Rouge-1 score vs. the number of selected sentences.

4 Sensitivity Experiments

To check the performance on the number of selected sentences, we conduct a sensitivity experiment for both CNN and DailyMail datasets. The results in Figure 3 show that the Multi-GraS performs the best when the number of the selected sentences is 2 for the CNN dataset and 3 for the DailyMail dataset.

5 Conclusion

In this paper, we propose a novel Multi-GCN to jointly model multiple relationships among words and sentences. Based on Multi-GCN, we propose a novel Multi-GraS model for extractive text summarization. Experimental results on the benchmark CNN/DailyMail dataset demonstrate the effectiveness of the proposed methods.

Acknowledgements

Jing and Tong are partially supported by NSF (1947135 and 1939725).

References

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

Manlio De Domenico, Albert Solé-Ribalta, Emanuele Cozzo, Mikko Kivelä, Yamir Moreno, Mason A Porter, Sergio Gómez, and Alex Arenas. 2013. Mathematical formulation of multilayer networks. *Physical Review X*, 3(4):041022.

Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. 2018. Banditsum: Extractive summarization as a contextual bandit. *arXiv preprint arXiv:1809.09672*.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

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

Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. 2020. Neural extractive summarization with hierarchical attentive heterogeneous graph network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3622–3631.

Baoyu Jing, Chenwei Lu, Deqing Wang, Fuzhen Zhuang, and Cheng Niu. 2018. Cross-domain labeled lda for cross-domain text classification. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 187–196. IEEE.

Baoyu Jing, Chanyoung Park, and Hanghang Tong. 2021a. Hdmi: High-order deep multiplex infomax. In *Proceedings of the Web Conference 2021*.

Baoyu Jing, Hanghang Tong, and Yada Zhu. 2021b. Network of tensor time series. In *Proceedings of the Web Conference 2021*, pages 2425–2437.

Baoyu Jing, Zeya Wang, and Eric Xing. 2019. Show, describe and conclude: On exploiting the structure information of chest x-ray reports. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6570–6580.

Baoyu Jing, Yuejia Xiang, Xi Chen, Yu Chen, and Hanghang Tong. 2021c. Graph-mvp: Multi-view prototypical contrastive learning for multiplex graphs. *arXiv preprint arXiv:2109.03560*.

Tom Kenter and Maarten De Rijke. 2015. Short text similarity with word embeddings. In *Proceedings of the 24th ACM international on conference on information and knowledge management*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, 1.

Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, Barcelona, Spain*.

Yang Liu and Mirella Lapata. 2019a. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy.

Yang Liu and Mirella Lapata. 2019b. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740.

Ling Luo, Xiang Ao, Yan Song, Feiyang Pan, Min Yang, and Qing He. 2019. Reading like HER: Human reading inspired extractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Alfonso Mendes, Shashi Narayan, Sebastião Miranda, Zita Marinho, André FT Martins, and Shay B Cohen. 2019. Jointly extracting and compressing documents with summary state representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3955–3966.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*.

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

Ani Nenkova and Kathleen McKeown. 2011. *Automatic summarization*. Now Publishers Inc.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Chanyoung Park, Donghyun Kim, Jiawei Han, and Hwanjo Yu. 2020. Unsupervised attributed multiplex network embedding. In *AAAI*, pages 5371–5378.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *ICLR*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

M. Pressley and Peter P. Afflerbach. 1995. Verbal protocols of reading: The nature of constructively responsive reading.

Vahed Qazvinian and Dragomir R Radev. 2008. Scientific paper summarization using citation summary networks. *arXiv preprint arXiv:0807.1560*.

Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Çelebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, Jahna Otterbacher, Hong Qi, Horacio Saggion, Simone Teufel, Michael Topper, Adam Winkel, and Zhu Zhang. 2004. MEAD - a platform for multidocument multilingual text summarization. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal.

Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Qiqihar Junior Teachers College*, 22:2004.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

Giannis Varelas, Epimenidis Voutsakis, Paraskevi Raftopoulou, Euripides GM Petrakis, and Evangelos E Milios. 2005. Semantic similarity methods in wordnet and their application to information retrieval on the web. In *Proceedings of the 7th annual ACM international workshop on Web information and data management*, pages 10–16.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020a. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.

Deqing Wang, Baoyu Jing, Chenwei Lu, Junjie Wu, Guannan Liu, Chenguang Du, and Fuzhen Zhuang. 2020b. Coarse alignment of topic and sentiment: A unified model for cross-lingual sentiment classification. *IEEE transactions on neural networks and learning systems*, 32(2):736–747.

Deqing Wang, Junjie Wu, Jingyuan Yang, Baoyu Jing, Wenjie Zhang, Xiaonan He, and Hui Zhang. 2021. Cross-lingual knowledge transferring by structural correspondence and space transfer. *IEEE Transactions on Cybernetics*.

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The World Wide Web Conference*, pages 2022–2032.

Yuxiang Wu and Baotian Hu. 2018. Learning to extract coherent summary via deep reinforcement learning. In *AAAI*.

Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3283–3294.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 5021–5031.

Yuchen Yan, Lihui Liu, Yikun Ban, Baoyu Jing, and Hanghang Tong. 2021. Dynamic knowledge graph alignment. In *AAAI*.

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

Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. 2018. Neural latent extractive document summarization. *arXiv preprint arXiv:1808.07187*.

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

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663.