HEGEL: Hypergraph Transformer for Long Document Summarization

Haopeng Zhang, Xiao Liu, Jiawei Zhang
IFM Lab, Department of Computer Science, University of California, Davis, CA, USA
haopeng,xiao,jiawei@ifmlab.org

Abstract

Extractive summarization for long documents is challenging due to the extended structured input context. The long-distance sentence dependency hinders cross-sentence relations modeling, the critical step of extractive summarization. This paper proposes HEGEL, a hypergraph neural network for long document summarization by capturing high-order cross-sentence relations. HEGEL updates and learns effective sentence representations with hypergraph transformer layers and fuses different types of sentence dependencies, including latent topics, keywords coreference, and section structure. We validate HEGEL by conducting extensive experiments on two benchmark datasets, and experimental results demonstrate the effectiveness and efficiency of HEGEL.

1 Introduction

Extractive summarization aims to generate a shorter version of a document while preserving the most salient information by directly extracting relevant sentences from the original document. With recent advances in neural networks and large pre-trained language models (Devlin et al., 2018; Lewis et al., 2019), researchers have achieved promising results in news summarization (around 650 words/document) (Nallapati et al., 2016a; Cheng and Lapata, 2016; See et al., 2017; Zhang et al., 2022; Narayan et al., 2018; Liu and Lapata, 2019). However, these models struggle when applied to long documents like scientific papers. The input length of a scientific paper can range from 2000 to 7,000 words, and the expected summary (abstract) is more than 200 words compared to 40 words in news headlines.

Scientific paper extractive summarization is highly challenging due to the long structured input. The extended context hinders sequential models like RNN from capturing sentence-level long-distance dependency and cross-sentence relations, which are essential for extractive summarization. In addition, the quadratic computation complexity of attention with respect to the input tokens length makes Transformer (Vaswani et al., 2017) based models not applicable. Moreover, long documents typically cover diverse topics and have richer structural information than short news, which is difficult for sequential models to capture.

As a result, researchers have turned to graph neural network (GNN) approaches to model cross-sentence relations. They generally represent a document with a sentence-level graph and turn extractive summarization into a node classification problem. These work construct graph from document in different manners, such as inter-sentence cosine similarity graph in (Erkan and Radev, 2004; Dong et al., 2020), Rhetorical Structure Theory (RST) tree relation graph in (Xu et al., 2019), approximate discourse graph in (Yasunaga et al., 2017), topic-sentence graph in (Cui and Hu, 2021) and word-document heterogeneous graph in (Wang et al., 2020). However, the usability of these approaches...
is limited by the following two aspects: (1) These methods only model the pairwise interaction between sentences, while sentence interactions could be triadic, tetradic, or of a higher-order in natural language (Ding et al., 2020). How to capture high-order cross-sentence relations for extractive summarization is still an open question. (2) These graph-based approaches rely on either semantic or discourses structure cross-sentence relation but are incapable of fusing sentence interactions from different perspectives. Sentences within a document could have various types of interactions, such as embedding similarity, keywords coreference, topical modeling from the semantic perspective, and section or rhetorical structure from the discourse perspective. Capturing multi-type cross-sentence relations could benefit sentence representation learning and sentence salience modeling. Figure 1 is an illustration showing different types of sentence interactions provide different connectivity for document graph construction, which covers both local and global context information.

To address the above issues, we propose HEGEL (HyperGraph transformer for Extractive Long document summarization), a graph-based model designed for summarizing long documents with rich discourse information. To better model high-order cross-sentence relations, we represent a document as a hypergraph, a generalization of graph structure, in which an edge can join any number of vertices. We then introduce three types of hyperedges that model sentence relations from different perspectives, including section structure, latent topic, and keywords coreference, respectively. We also propose hypergraph transformer layers to update and learn effective sentence embeddings on hypergraphs. We validate HEGEL by conducting extensive experiments and analyses on two benchmark datasets, and experimental results demonstrate the effectiveness and efficiency of HEGEL. We highlight our contributions as follows:

(i) We propose a hypergraph neural model, HEGEL, for long document summarization. To the best of our knowledge, we are the first to model high-order cross-sentence relations with hypergraphs for extractive document summarization.

(ii) We propose three types of hyperedges (section, topic, and keyword) that capture sentence dependency from different perspectives. Hypergraph transformer layers are then designed to update and learn effective sentence representations by message passing on the hypergraph.

(iii) We validate HEGEL on two benchmarked datasets (arXiv and PubMed), and the experimental results demonstrate its effectiveness over state-of-the-art baselines. We also conduct ablation studies and qualitative analysis to investigate the model performance further.

2 Related Works

2.1 Scientific Paper Summarization

With the promising progress on short news summarization, research interest in long-form documents like academic papers has arisen. Cohan et al. (2018) proposed benchmark datasets ArXiv and PubMed, and employed pointer generator network with hierarchical encoder and discourse-aware decoder. Xiao and Carenini (2019) proposed an encoder-decoder model by incorporating global and local contexts. Ju et al. (2021) introduced an encoder-decoder approach to summarize long scientific documents based on the Information Bottleneck principle. Dong et al. (2020) came up with an unsupervised extractive approach to summarize long scientific documents. Researchers also explore supervised graph neural networks for summarization. Yasunaga et al. (2017) applied Graph Convolutional Network (GCN) on structural discourse graphs based on RST trees and coreference mentions. Cui et al. (2020) leveraged topical information by building topic-sentence graphs. Recently, Wang et al. (2020) proposed to construct word-document heterogeneous graphs and use word nodes as the intermediary between sentences. Jing et al. (2021) proposed to use multiplex graph to consider different sentence relations. Our paper follows this line of work on developing novel graph neural networks for single document extractive summarization. The main difference is that we construct a hypergraph from
a document that could capture high-order cross-sentence relations instead of pairwise relations, and fuse different types of sentence dependencies, including section structure, latent topics, and keywords coreference.

3 Method

In this section, we introduce HEGEL in great detail. We first present how to construct a hypergraph for a given long document. After encoding sentences into contextualized representations, we extract their section, latent topic, and keyword coreference relations and fuse them into a hypergraph. Then, our hypergraph transformer layer will update and learn sentence representations according to the hypergraph. Finally, HEGEL will score the salience of sentences based on the updated sentence representations to determine if the sentence should be included in the summary. The overall architecture of our model is shown in Figure 2(a).

3.1 Document as a Hypergraph

A hypergraph is defined as a graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ represents the set of nodes, and $\mathcal{E} = \{e_1, \ldots, e_m\}$ represents the set of hyperedges in the graph. Here each hyperedge $e$ connects two or more nodes (i.e., $\sigma(e) \geq 2$). Specifically, we use the notations $v \in e$ and $v \notin e$ to denote node $v$ is connected to hyperedge $e$ or not in the graph $G$, respectively. The topological structure of hypergraph can also be represented by its incidence matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$:

$$A_{ij} = \begin{cases} 1, & \text{if } v_i \in e_j \\ 0, & \text{if } v_i \notin e_j \end{cases}$$

Given a document $D = \{s_1, s_2, \ldots, s_n\}$, each sentence $s_i$ is represented by a corresponding node $v_i \in \mathcal{V}$. A Hyperedge $e_j$ will be created if a subset of nodes $\mathcal{V}_j \subset \mathcal{V}$ share common semantic or structural information.

3.1.1 Node Representation

We first adopt sentence-BERT (Reimers and Gurevych, 2019) as sentence encoder to embed the semantic meanings of sentences as $X = \{x_1, x_2, \ldots, x_n\}$. Note that the sentence-BERT is only used for initial sentence embedding, but not updated in HEGEL.

To preserve the sequential information, we also add positional encoding following Transformer (Vaswani et al., 2017). We adopt the hierarchical position embedding (Ruan et al., 2022), where position of each sentence $s_i$ can be represented as two parts: the section index of the sentence $p_i^{sec}$, and the sentence index in its corresponding section $p_i^{sen}$. The hierarchical position embedding (HPE) of sentence $s_i$ can be calculated as:

$$\text{HPE}(s_i) = \gamma_1 \text{PE}(p_i^{sec}) + \gamma_2 \text{PE}(p_i^{sen}),$$

where $\gamma_1, \gamma_2$ are two hyperparameters to adjust the scale of positional encoding and $\text{PE}(-)$ refers to the position encoding function:
\[ \text{PE}(pos, 2i) = \sin(pos/10000^{2i/d_{\text{model}}}), \]
\[ \text{PE}(pos, 2i + 1) = \cos(pos/10000^{2i/d_{\text{model}}}). \]

Then we can get the initial input node representations \( H^0 = \{h^0_1, h^0_2, ..., h^0_n\} \), with vector \( h^0_i \) defined as:
\[ h^0_i = x_i + \text{HPE}(s_i) \]

### 3.1.2 Hyperedge Construction

To effectively model multi-type cross-sentence relations in a long context, we propose the following three hyperedges. These hyperedges could capture high-order context information via the multi-node connection and model both local and global context through document structures from different perspectives.

#### Section Hyperedges:

Scientific papers mostly follow a standard discourse structure describing the problem, methodology, experiments/results, and finally conclusions, so sentences within the same section tend to have the same semantic focus (Suppe, 1998). To capture the local sequential context, we build section hyperedges that consider each section as a hyperedge that connects all the sentences in this section. Section hyperedges could also address the incidence matrix sparsity issue and ensure all nodes of the graph are connected by at least one hyperedge. Assume a document has \( q \) sections, section hyperedge \( e^\text{sec}_j \) for the \( j \)-th section can be represented formally in its corresponding incidence matrix \( A^\text{sec} \in \mathbb{R}^{n \times q} \) as:
\[ A^\text{sec}_{ij} = \begin{cases} 1, & \text{if } s_i \in e^\text{sec}_j \\ 0, & \text{if } s_i \notin e^\text{sec}_j \end{cases} \]

where \( A^\text{sec}_{ij} \) denotes whether the \( i \)-th sentence is in the \( j \)-th section.

#### Topic Hyperedges:

Topical information has been demonstrated to be effective in capturing important content (Cui et al., 2020). To leverage topical information of the document, we first apply the Latent Dirichlet Allocation (LDA) model (Blei et al., 2003) to extract the latent topic relationships between sentences and then construct the topic hyperedge. In addition, topic hyperedges could address the long-distance dependency problem by capturing global topical information of the document. After extracting \( p \) topics from LDA, we construct \( p \) corresponding topic hyperedges \( e^\text{topic}_j \), represented by the entry \( A^\text{topic}_{ij} \) in the incidence matrix \( A^\text{topic} \in \mathbb{R}^{n \times p} \) as:
\[ A^\text{topic}_{ij} = \begin{cases} 1, & \text{if } s_i \in e^\text{topic}_j \\ 0, & \text{if } s_i \notin e^\text{topic}_j \end{cases} \]

where \( A^\text{topic}_{ij} \) denotes whether the \( i \)-th sentence belongs to the \( j \)-th latent topic.

#### Keyword Hyperedges:

Previous work finds that keywords compose the main body of the sentence, which are regarded as the indicators for important sentence selection (Wang and Cardie, 2013; Li et al., 2020). Keywords in the original sentence provide significant clues for the main points of the sentence. To utilize keyword information, we first extract keywords for academic papers with KeyBERT (Grootendorst, 2020) and construct keyword hyperedges to link the sentences that contain the same keyword regardless of their sequential distance. Like topic hyperedges, keyword hyperedges also capture global context relations and thus, address the long-distance dependency problem. After extracting \( k \) keywords for a document, we construct \( k \) corresponding keyword hyperedges \( e^\text{kw}_j \), represented in the incidence matrix \( A^\text{kw} \in \mathbb{R}^{n \times k} \) as:
\[ A^\text{kw}_{ij} = \begin{cases} 1, & \text{if } s_i \in e^\text{kw}_j \\ 0, & \text{if } s_i \notin e^\text{kw}_j \end{cases} \]

where \( s_i \in e^\text{kw}_j \) means the \( i \)-th sentence contains the \( j \)-th keyword.

We finally fuse the three hyperedges by concatenation \( || \) and get the overall incidence matrix \( A \in \mathbb{R}^{n \times m} \) as:
\[ A = A^\text{sec} || A^\text{topic} || A^\text{kw}, \]

where dimension \( m = q + p + k \).

The initial input node representations \( H^0 = \{h^0_1, h^0_2, ..., h^0_n\} \) and the overall hyperedge incidence matrix \( A \) will be fed into hypergraph transformer layers to learn effective sentence embeddings.

### 3.2 Hypergraph Transformer Layer

The self-attention mechanism in Transformer (Vaswani et al., 2017) has demonstrated its effectiveness for learning text representation and graph representations (Veličković et al., 2017; Ying et al., 2021; Ding et al., 2020; Zhang and Zhang, 2020;
To model cross-sentence relations and learn effective sentence (node) representations in hypergraphs, we propose the Hypergraph Transformer Layer as in Figure 2(b).

3.2.1 Hypergraph Attention
Given node representations $H^0 = \{h_1^0, h_2^0, ..., h_n^0\}$ and hyperedge incidence matrix $A \in \mathbb{R}^{n \times m}$, a $l$-layer hypergraph transformer computes hypergraph attention (HGA) and updates node representations $H$ in an iterative manner as shown in Algorithm 1.

Specifically, in each iteration, we first obtain all $m$ hyperedge representations $\{g_1^l, g_2^l, ..., g_m^l\}$ as:

$$g_j^l = \text{LeakyReLU} \left( \sum_{u_k \in e_j} \alpha_{jk} W_h h_k^{l-1} \right),$$

$$\alpha_{jk} = \frac{\exp (w_{ah}^T u_k)}{\sum_{p \in e_j} \exp (w_{ah}^T u_p)},$$

$$u_k = \text{LeakyReLU} \left( W_h h_k^{l-1} \right),$$

where the superscript $l$ denotes the model layer, matrices $W_h, w_{ah}$ are trainable weights and $\alpha_{jk}$ is the attention weight of node $u_k$ in hyperedge $e_j$.

The second step is to update node representations $H^{l-1}$ based on the updated hyperedge representations $\{g_1^l, g_2^l, ..., g_m^l\}$ by:

$$h_i^l = \text{LeakyReLU} \left( \sum_{v_k \in e_k} \beta_{ki} W_e g_k^l \right),$$

$$\beta_{ki} = \frac{\exp (w_{ae}^T z_k)}{\sum_{v_l \in e_k} \exp (w_{ae}^T z_l)},$$

$$z_k = \text{LeakyReLU} \left( \left[ W_v g_k^l \| W_h h_k^{l-1} \right] \right),$$

where $h_i^l$ is the representation of node $v_i$, $W_e, w_{ae}$ are trainable weights, and $\beta_{ki}$ is the attention weight of hyperedge $e_k$ that connects node $v_i$. Here is the concatenation operation. In this way, information of different granularities and types can be fully exploited through the hypergraph attention message passing processes.

3.2.2 Hypergraph Transformer
After obtaining the multi-head attention, we also introduce the feed-forward blocks (FFN) with residual connection and layer normalization (LN) like in Transformer. We formally characterize the Hypergraph Transformer layer as below:

$$H^{(l)} = \text{LN}(\text{MH-HGA}(H^{l-1}, A) + H^{l-1})$$

$$H^l = \text{LN}(\text{FFN}(H^{(l)}) + H^{(l)})$$

### Algorithm 1: MH-HGA$\text{head}(H, A)$

| input | node representation $H^{l-1} \in \mathbb{R}^{n \times d}$, incidence matrix $A \in \mathbb{R}^{n \times m}$ | output | updated representation $H^l \in \mathbb{R}^{n \times d}$ |
|-------|-------------------------------------------------|--------|--------------------------------|
| for | head = 1, 2, ..., h do | for | node $v_k \in e_j$ do |
| end | // update hyperedges from nodes | end | // update node representations |
| for | j = 1, 2, ..., m do | for | hyperedge that $v_i \in e_k$ do |
| end | // update hyperedge representation $g_j^l$ | end | update node representation $h_i^l$ with Eq. 13; |
| end | // update node representations | end | compute attention $\beta_{ki}$ with Eq. 13; |
| for | i = 1, 2, ..., n do | end | update node representation $h_i^l$ with Eq. 12; |
| end | // update hyperedges from nodes |

3.3 Training Objective
After passing $L$ hypergraph transformer layers, we obtain the final sentence node representations $H^L = \{h_1^L, h_2^L, ..., h_n^L\}$. We then add a multi-layer perceptron(MLP) followed by a sigmoid activation function indicating the confidence score for selecting each sentence. Formally, the predicted confidence score $\hat{y}_i$ for sentence $s_i$ is:

$$z_i = \text{LeakyReLU}(W_p h_i^L),$$

$$\hat{y}_i = \text{sigmoid}(W_{pz} z_i),$$

$$\text{MH-HGA}(H, A) = \sigma(W_O ||_{i=1}^{h} \text{head}_i),$$

where HGA$(\cdot)$ denotes hypergraph attention, $\sigma$ is the activation function, $W_O$ is the multi-head weight, and $\|\|$ denotes concatenation.
The LEAD method has limited performance on scientific paper summarization compared to...
Table 2: Experimental Results on PubMed and Arxiv datasets.

| Models               | PubMed | Arxiv |
|----------------------|--------|-------|
|                      | ROUGE-1 | ROUGE-2 | ROUGE-L | ROUGE-1 | ROUGE-2 | ROUGE-L |
| ORACLE               | 55.05  | 27.48  | 49.11   | 53.88  | 23.05  | 46.54   |
| LEAD                 | 35.63  | 12.28  | 25.17   | 33.66  | 8.94   | 22.19   |
| LexRank (2004)       | 39.19  | 13.89  | 34.59   | 33.85  | 10.73  | 28.99   |
| PACSUM (2019)        | 39.79  | 14.00  | 36.09   | 38.57  | 10.93  | 34.33   |
| HIPORANK (2021)      | 43.58  | 17.00  | 39.31   | 39.34  | 12.56  | 34.89   |
| Cheng&Lapata (2016)  | 43.89  | 18.53  | 34.59   | 43.89  | 18.53  | 30.89   |
| SummaRuNNer (2016)   | 44.85  | 19.70  | 31.43   | 43.89  | 18.53  | 29.14   |
| ExtSum-LG (2019)     | 45.01  | 19.91  | 41.16   | 43.62  | 17.36  | 30.41   |
| SentCLF (2020)       | 43.30  | 17.92  | 39.47   | 42.32  | 15.63  | 38.06   |
| SentPTR (2020)       | 45.30  | 20.42  | 40.95   | 44.01  | 17.79  | 39.09   |
| ExtSum-LG + RdLoss (2021) | 45.39  | 20.37  | 40.99   | 43.87  | 17.50  | 38.97   |
| ExtSum-LG + MMR (2021) | 46.59  | 20.39  | 42.11   | 45.22  | 17.67  | 40.16   |
| PGN (2017)           | 35.86  | 10.22  | 29.69   | 32.06  | 9.04   | 25.16   |
| DiscourseAware (2018)| 38.93  | 15.37  | 35.21   | 35.80  | 11.05  | 31.80   |
| TLM-IE (2020)        | 42.13  | 16.27  | 39.21   | 41.62  | 14.69  | 38.03   |
| DANCER-LSTM (2020)   | 44.09  | 17.69  | 40.27   | 41.87  | 15.92  | 37.61   |
| DANCER-RUM (2020)    | 43.98  | 17.65  | 40.25   | 42.70  | 16.54  | 38.44   |
| **HEGEL** (ours)     | **47.13** | **21.00** | **42.18** | **46.41** | **18.17** | **39.89** |

Table 3: Ablation study results on PubMed dataset.

| Model               | ROUGE-1 | ROUGE-2 | ROUGE-L |
|---------------------|---------|---------|---------|
| full HEGEL          | **47.13** | **21.00** | **42.18** |
| w/o Position        | 46.86   | 20.05   | 41.91   |
| w/o Keyword         | 46.92   | 20.71   | 42.03   |
| w/o Topic           | 46.35   | 20.30   | 41.48   |
| w/o Section         | 45.63   | 19.30   | 40.71   |

5 Analysis

5.1 Ablation Study

We first analyze the influence of different components of HEGEL. Table 3 shows the experimental results of removing hyperedges and the hierarchical position encoding of HEGEL on the PubMed dataset. As shown in the second row, removing the hierarchical position embedding hurts the model performance, which indicates the importance of injecting sequential order information. Regarding hyperedges (row 3-5), we can see that all three types of hyperedges (section, keyword, and topic) help boost the overall model performance. Specifically, the performance drops most when the section hyperedges are removed. The hypergraph becomes sparse and hurts its connectivity. This indicates that the section hyperedges, which contain local context information, play an essential role in the information aggregation process. Note that although we only discuss three types of hyperedges (section, keyword, and topic) in this work, it is easy to extend our model with hyperedges from other perspectives like syntactic for future work.
5.2 Hyperedge Analysis

![Figure 3: Average attention distribution over three types of hyperedges on PubMed dataset.](image)

We also explore the hyperedge pattern to understand the performance of HEGEL further. As shown in Figure 3, we have the most topic hyperedges on average, and section hyperedges have the largest degree (number of connected nodes). In terms of cross attention over the predicted sentence nodes, HEGEL pays more than half of the attention to section hyperedges and pays least to keywords edges. The results are consistent with the earlier ablation study that local section context information plays a more critical role in long document summarization.

5.3 Embedding Analysis

![Figure 4: Visualization of sentence nodes embeddings for 100 documents in PubMed test set.](image)

To explore the sentence embedding learned by HEGEL, we show a visualization of the output sentence node embedding from the last hypergraph transformer layer. We employ T-SNE (van der Maaten and Hinton, 2008) and reduce each node’s dimension to 2, as shown in Figure 4. The orange dots represent the ground truth sentences, and the blue dots are the non-ground truth sentences. We can see some clustering effects of the ground truth nodes, which also tend to appear in the bottom left zone of the plot. The results indicate that HEGEL learns effective sentence embeddings as indicators for salient sentence selection.

5.4 Case Study

Here we also provide an example output summary from HEGEL in Table 4. We could see that the selected sentences span a long distance in the original document, but are triadically related according to the latent topic and keyword coreference. As a result, HEGEL effectively captures high-order cross-sentence relations through multi-type hyperedges and selects these salient sentences according to learned high-order representation.

| Method | Phylogenetic analyses of partial middle east respiratory syndrome coronavirus genomic sequences for viruses detected in dromedaries imported from oman to united arab emirates, may 2015. (Section 1) |
|--------|----------------------------------------------------------------------------------------------------------------------------------|
| Method | Merscov genomic sequences determined in this study are similar to those of viruses detected in 2015 in patients in saudi arabia and south korea with hospital-acquired infections. (Section 3) |
| Information | Our findings provide further evidence that asymptomatic human infections can be caused by zoonotic transmission. (Section 2) |
| Information | The infected dromedaries were imported from oman, which suggests that viruses from this clade are circulating on the arabian peninsula. (Section 4) |

Table 4: An example output summary of HEGEL. Topics are marked in orange, key words are marked in green, and sections are marked in blue.

6 Conclusion

This paper presents HEGEL for long document summarization. HEGEL represents a document as a hypergraph to address the long dependency issue and captures higher-order cross-sentence relations through multi-type hyperedges. The strong performance of HEGEL demonstrates the importance of modeling high-order sentence interactions and fusing semantic and structural information for future research in long document extractive summarization.
Limitations

Despite the strong performance of HEGEL, its design still has the following limitations. First, HEGEL relies on existing keyword and topic models to pre-process the document and construct hypergraphs. In addition, we only explore academic paper datasets as a typical example for long document summarization.

The above limitations may raise concerns about the model’s performance. However, HEGEL is an end-to-end model, so the pre-process steps do not add the model computation complexity. Indeed, HEGEL relies on hyperedge for cross-sentence attention, so it is parameter-efficient and uses 50% less parameters than heterogeneous graph model (Wang et al., 2020) and 90% less parameters than Longformer-base (Beltagy et al., 2020). On the other hand, our experimental design follows a series of previous long document summarization work (Xiao and Carenini, 2019, 2020; Subramanian et al., 2019; Ruan et al., 2022; Dong et al., 2020; Cohan et al., 2018) on benchmark datasets ArXiv and PubMed. These two new datasets contain much longer documents, richer discourse structure than all the news datasets and are therefore ideal test-beds for long document summarization.

Acknowledgements

This work is supported by NSF through grants IIS-1763365 and IIS-2106972. We thank the anonymous reviewers for the helpful feedback.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. Journal of machine Learning research, 3(Jan):993–1022.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. arXiv preprint arXiv:1603.07232.

Arman Cohan, Franck Dernoncourt, Dooho Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. arXiv preprint arXiv:1804.05685.

Peng Cui and Le Hu. 2021. Topic-guided abstractive multi-document summarization. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 1463–1472, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Peng Cui, Le Hu, and Yuanchao Liu. 2020. Enhancing extractive text summarization with topic-aware graph neural networks. arXiv preprint arXiv:2010.06253.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. 2020. Be more with less: Hypergraph attention networks for inductive text classification. arXiv preprint arXiv:2011.00387.

Yue Dong, Andrei Mircea, and Jackie CK Cheung. 2020. Discourse-aware unsupervised summarization of long scientific documents. arXiv preprint arXiv:2005.00513.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of artificial intelligence research, 22:457–479.

Alexios Gidiotis, Stefanos Stefanidis, and Grigoris Tsoumakas. 2020. AUTH @ CLSciSumm 20, LaySumm 20, LongSumm 20. In Proceedings of the First Workshop on Scholarly Document Processing, pages 251–260, Online. Association for Computational Linguistics.

Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert.

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

Baoyu Jing, Zeyu You, Tao Yang, Wei Fan, and Hanghang Tong. 2021. Multiplex graph neural network for extractive text summarization. arXiv preprint arXiv:2108.12870.

Jiaxin Ju, Ming Liu, Huan Yee Koh, Yuan Jin, Lan Du, and Shirui Pan. 2021. Leveraging information bottlenecks for scientific document summarization. arXiv preprint arXiv:2110.01280.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

Haoran Li, Junnan Zhu, Jiajun Zhang, Chengqing Zong, and Xiaodong He. 2020. Keywords-guided abstractive sentence summarization. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 8196–8203.
