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.

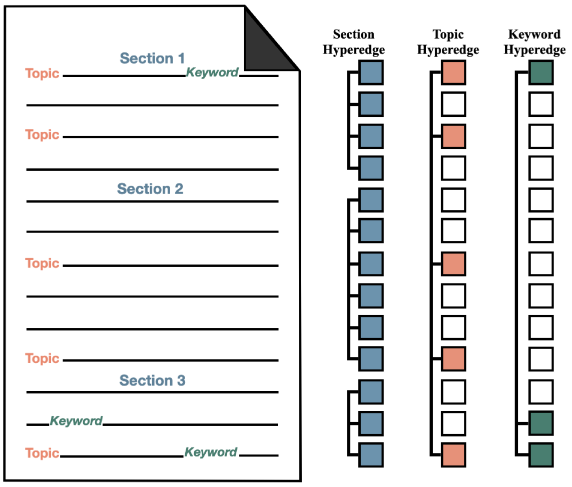


Figure 1: An illustration of modeling cross-sentence relations from section structure, latent topic, and keyword coreference perspectives.

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 pretrained 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

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 10167 - 10176
December 7-11, 2022 ©2022 Association for Computational Linguistics

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 (**H**yp**E**r**G**raph transformer for **E**xtractive **L**ong 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.

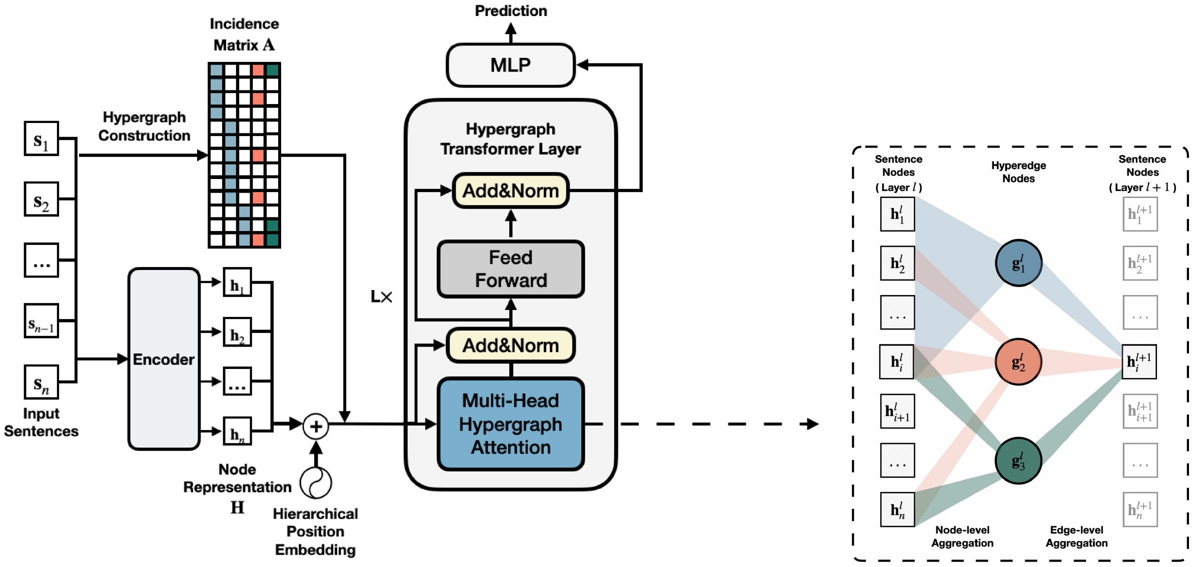
2 Related Works

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 unsupervised extractive approach to summarize long scientific documents based on the Information Bottleneck principle. Dong et al. (2020) came up with an unsupervised ranking model by incorporating hierarchical graph representation and asymmetrical positional cues. Recently, Ruan et al. (2022) proposed to apply pre-trained language model with hierarchical structure information.

2.2 Graph based summarization

Graph-based models have been exploited for extractive summarization to capture cross-sentence dependencies. Unsupervised graph summarization methods rely on graph connectivity to score and rank sentences (Radev et al., 2004; Zheng and Lapata, 2019; Dong et al., 2020). Researchers also explore supervised graph neural networks for summarization. Yasunaga et al. (2017) applied Graph Convolutional Network (GCN) on the approximate discourse graph. Xu et al. (2019) proposed to apply 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) (b)

Figure 2: (a) The overall architecture of HEGEL. (b) Two-phase message passing in hypergraph transformer layer

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 $\mathcal{V} = \{v_1, \dots, v_n\}$ represents the set of nodes, and $\mathcal{E} = \{e_1, \dots, e_m\}$ 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 $\mathbf{A} \in \mathbb{R}^{n \times m}$:

$$\mathbf{A}_{ij} = \begin{cases} 1, & \text{if } v_i \in e_j \\ 0, & \text{if } v_i \notin e_j \end{cases} \tag{1}$$

Given a document $D = \{s_1, s_2, ..., s_n\}$, 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 $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$. 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:

$$\text{HPE}(s_i) = \gamma_1 \text{PE}(p_i^{sec}) + \gamma_2 \text{PE}(p_i^{sen}), \tag{2}$$

where γ_1, γ_2 are two hyperparameters to adjust the scale of positional encoding and PE$(\cdot)$ refers to the position encoding function:

$$\mathrm{PE}(pos, 2i) = \sin(pos/10000^{2i/d_{model}}), \tag{3}$$

$$\mathrm{PE}(pos, 2i+1) = \cos(pos/10000^{2i/d_{model}}). \tag{4}$$

Then we can get the initial input node representations $\mathbf{H}^0 = \{\mathbf{h}_1^0, \mathbf{h}_2^0, ..., \mathbf{h}_n^0\}$, with vector $\mathbf{h}_i^0$ defined as:

$$\mathbf{h}_i^0 = \mathbf{x}_i + \mathrm{HPE}(s_i) \tag{5}$$

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_j^{sec} for the j-th section can be represented formally in its corresponding incidence matrix $\mathbf{A}_{sec} \in \mathbb{R}^{n \times q}$ as:

$$A_{ij}^{sec} = \begin{cases} 1, \text{ if } s_i \in e_j^{sec} \\ 0, \text{ if } s_i \notin e_j^{sec} \end{cases} \tag{6}$$

where A_{ij}^{sec} 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_j^{topic}, represented by the entry A_{ij}^{topic} in the incidence matrix $\mathbf{A}_{topic} \in \mathbb{R}^{n \times p}$ as:

$$A_{ij}^{topic} = \begin{cases} 1, \text{ if } s_i \in e_j^{topic} \\ 0, \text{ if } s_i \notin e_j^{topic} \end{cases} \tag{7}$$

where A_{ij}^{topic} 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_j^{kw}, represented in the incidence matrix $\mathbf{A}_{kw} \in \mathbb{R}^{n \times k}$ as:

$$A_{ij}^{kw} = \begin{cases} 1, \text{ if } s_i \in e_j^{kw} \\ 0, \text{ if } s_i \notin e_j^{kw}, \end{cases} \tag{8}$$

where $s_i \in e_j^{kw}$ means the i-th sentence contains the j-th keyword.

We finally fuse the three hyperedges by concatenation $\|$ and get the overall incidence matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ as:

$$\mathbf{A} = \mathbf{A}_{sec} \| \mathbf{A}_{topic} \| \mathbf{A}_{kw}, \tag{9}$$

where dimension $m = q + p + k$

The initial input node representations $\mathbf{H}^0 = \{\mathbf{h}_1^0, \mathbf{h}_2^0, ..., \mathbf{h}_n^0\}$ and the overall hyperedge incidence matrix $\mathbf{A}$ 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;

Zhang et al., 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 $\mathbf{H}^0 = \{\mathbf{h}_1^0, \mathbf{h}_2^0, ..., \mathbf{h}_n^0\}$ and hyperedge incidence matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$, a l-layer hypergraph transformer computes hypergraph attention (HGA) and updates node representations $\mathbf{H}$ in an iterative manner as shown in Algorithm 1.

Specifically, in each iteration, we first obtain all m hyperedge representations $\{\mathbf{g}_1^l, \mathbf{g}_2^l, ..., \mathbf{g}_m^l\}$ as:

$$\mathbf{g}_j^l = \text{LeakyReLU}\left(\sum_{v_k \in e_j} \alpha_{jk} \mathbf{W}_h \mathbf{h}_k^{l-1}\right), \quad (10)$$

$$\begin{aligned} \alpha_{jk} &= \frac{\exp\left(\mathbf{w}_{ah}^{\mathrm{T}} \mathbf{u}_k\right)}{\sum_{v_p \in e_j} \exp\left(\mathbf{w}_{ah}^{\mathrm{T}} \mathbf{u}_p\right)}, \\ \mathbf{u}_k &= \text{LeakyReLU}\left(\mathbf{W}_h \mathbf{h}_k^{l-1}\right), \end{aligned} \quad (11)$$

where the superscript l denotes the model layer, matrices $\mathbf{W}_h, \mathbf{w}_{ah}$ are trainable weights and α_{jk} is the attention weight of node v_k in hyperedge e_j.

The second step is to update node representations $\mathbf{H}^{l-1}$ based on the updataed hyperedge representations $\{\mathbf{g}_1^l, \mathbf{g}_2^l, ..., \mathbf{g}_m^l\}$ by:

$$\mathbf{h}_i^l = \text{LeakyReLU}\left(\sum_{v_i \in e_k} \beta_{ij} \mathbf{W}_e \mathbf{g}_k^l\right), \quad (12)$$

$$\begin{aligned} \beta_{ki} &= \frac{\exp\left(\mathbf{w}_{ae}^{\mathrm{T}} \mathbf{z}_k\right)}{\sum_{v_i \in e_q} \exp\left(\mathbf{w}_{ae}^{\mathrm{T}} \mathbf{z}_i\right)}, \\ \mathbf{z}_k &= \text{LeakyReLU}\left(\left[\mathbf{W}_e \mathbf{g}_k^l \| \mathbf{W}_h \mathbf{h}_i^{l-1}\right]\right), \end{aligned} \quad (13)$$

where $\mathbf{h}_i^l$ is the representation of node v_i, $\mathbf{W}_e, \mathbf{w}_{ae}$ are trainable weights, and β_{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.

Multi-Head Hypergraph Attention As in Transformer, we also extend hypergraph attention (HGA) into multi-head hypergraph attention (MH-HGA) to expand the model's representation subspaces, represented as:

$$\begin{aligned} \text{MH-HGA}(\mathbf{H}, \mathbf{A}) &= \sigma(\mathbf{W}_O \|_{i=1}^h \text{head}_i), \\ \text{head}_i &= \text{HGA}_i(\mathbf{H}, \mathbf{A}), \end{aligned} \quad (14)$$

where $\text{HGA}(\cdot)$ denotes hypergraph attention, σ is the activation function, $\mathbf{W}_O$ is the multi-head weight, and $\|$ denotes concatenation.

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:

$$\begin{aligned} \mathbf{H}'^{(l)} &= \text{LN}(\text{MH-HGA}(\mathbf{H}^{l-1}, \mathbf{A}) + \mathbf{H}^{l-1}) \\ \mathbf{H}^l &= \text{LN}(\text{FFN}(\mathbf{H}'^{(l)}) + \mathbf{H}'^{(l)}) \end{aligned} \quad (15)$$

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

```
input  : node representation $\mathbf{H}^{l-1} \in \mathbb{R}^{n \times d}$,
         incidence matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$
output : updated representation $\mathbf{H}^l \in \mathbb{R}^{n \times d}$
for $head = 1, 2, ..., h$ do
    // update hyperedges from nodes
    for $j = 1, 2, ..., m$ do
        for node $v_k \in e_j$ do
            compute attention $\alpha_{jk}$ with Eq. 11;
            update hyperedge representation $\mathbf{g}_j^l$
              with Eq. 10;
        end
    end
    // update node representations
    for $i = 1, 2, ..., n$ do
        for hyperedge that $v_i \in e_k$ do
            compute attention $\beta_{ki}$ with Eq. 13;
              update node representation $\mathbf{h}_i^l$ with
              Eq. 12;
        end
    end
end
```

3.3 Training Objective

After passing L hypergraph transformer layers, we obtain the final sentence node representations $\mathbf{H}^L = \{\mathbf{h}_1^L, \mathbf{h}_2^L, ..., \mathbf{h}_n^L\}$. 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:

$$\begin{aligned} \mathbf{z}_i &= \text{LeakyReLU}(\mathbf{W}_{p1} \mathbf{h}_i^L), \\ \hat{y}_i &= \text{sigmoid}(\mathbf{W}_{p2} \mathbf{z}_i), \end{aligned} \quad (16)$$

	Arxiv	PubMed
# train	201,427	112,291
# validation	6,431	6,402
# test	6,436	6,449
avg. document length	4,938	3,016
avg. summary length	203	220

Table 1: Statistics of PubMed and Arxiv datasets.

where $\mathbf{W}_{p1}, \mathbf{W}_{p2}$ are trainable parameters.

Compared with the sentence ground truth label y_i, we train HEGEL in an end-to-end manner and optimize with binary cross-entropy loss as:

$$\mathcal{L} = -\frac{1}{N \cdot N_d} \sum_{d=1}^{N} \sum_{i=1}^{N_d} (y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i)), \tag{17}$$

where N denotes the number of training instances in the training set, and N_d denotes the number of sentences in the document.

4 Experiment

This section presents experimental details on two benchmarked academic paper summarization datasets. We compare our proposed model with state-of-the-art baselines and conduct detailed analysis to validate the effectiveness of HEGEL.

4.1 Experiment Setup

Datsasets Scientific papers are an example of long documents with section discourse structure. Here we validate HEGEL on two benchmark scientific paper summarization datasets: ArXiv and PubMed (Cohan et al., 2018). PubMed contains academic papers from the biomedical domain, while arXiv contains papers from different scientific domains. We use the original train, validation, and testing splits as in (Cohan et al., 2018). The detailed statistics of datasets are shown in Table 1.

Compared Baselines We perform a systematic comparison with state-of-the-art baseline approaches as follows:

- Unsupervised methods: LEAD that selects the first few sentences as summary; graph-based methods LexRank (Erkan and Radev, 2004), PACSUM (Zheng and Lapata, 2019), and HIPORANK (Dong et al., 2020).
- Neural extractive models: encoder-decoder based model Cheng&Lapata (Cheng and Lapata, 2016) and SummaRuNNer (Nallapati et al., 2016a); local and global context model ExtSum-LG (Xiao and Carenini, 2019) and its variant RdLoss/MMR (Xiao and Carenini, 2020); transformer-based models SentCLF, SentPTR (Subramanian et al., 2019), and HiStruct+ (Ruan et al., 2022).
- Neural abstractive models: pointer network PGN (See et al., 2017), hierarchical attention model DiscourseAware (Cohan et al., 2018), transformer-based model TLM-I+E (Subramanian et al., 2019), and divide-and-conquer method DANGER (Gidiotis et al., 2020).

4.2 Implementation Details

We use pre-trained sentence-BERT (Reimers and Gurevych, 2019) checkpoint *all-mpnet-base-v2* as the encoder for initial sentence representations. The embedding dimension is 768, and the input layer dimension is 1024. In our experiment, we stack two layers of hypergraph transformer, and each has 8 attention heads with a hidden dimension of 128. The output layer's hidden dimension is set to 4096. We generate at most 100 topics for each document and filter out the topic and keyword hyperedges that connect less than 5 sentence nodes or greater than 25 sentence nodes. For position encodings, we set the rescale weights γ_1 and γ_2 to 0.001.

The model is optimized with Adam optimizer (Loshchilov and Hutter, 2017) with a learning rate of 0.0001 and a dropout rate of 0.3. We train the model on an RTX A6000 GPU for 20 epochs and validate after each epoch using ROUGE-1 F-score to choose checkpoints. Early stopping is employed to select the best model with the patience of 3.

Following the standard-setting, we use ROUGE F-scores (Lin and Hovy, 2003) for performance evaluation. Specifically, ROUGE-1/2 scores measure summary informativeness, and the ROUGE-L score measures summary fluency. Following prior work (Nallapati et al., 2016b), we construct extractive ground truth (ORACLE) by greedily optimizing the ROUGE score on the gold-standard abstracts for extractive summary labeling.

4.3 Experiment Results

The performance of HEGEL and baseline methods on arXiv and Pubmed datasets are shown in Table 2. The first block lists the extractive ground truth ORACLE and the unsupervised methods. The second block includes recent extractive summarization models, and the third contains state-of-the-art abstractive methods.

The LEAD method has limited performance on scientific paper summarization compared to

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	30.17	42.24	15.97	27.88
SummaRuNNer (2016)	43.89	18.78	30.36	42.81	16.52	28.23
ExtSum-LG (2019)	44.85	19.70	31.43	43.62	17.36	29.14
SentCLF (2020)	45.01	19.91	41.16	34.01	8.71	30.41
SentPTR (2020)	43.30	17.92	39.47	42.32	15.63	38.06
ExtSum-LG + RdLoss (2021)	45.30	20.42	40.95	44.01	17.79	39.09
ExtSum-LG + MMR (2021)	45.39	20.37	40.99	43.87	17.50	38.97
HiStruct+ (2022)	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-I+E (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 2: Experimental Results on PubMed and Arxiv datasets.

its strong performance on short news summarization like CNN/Daily Mail (Hermann et al., 2015) and New York Times (Sandhaus, 2008). The phenomenon indicates that academic paper has less positional bias than news articles, and the ground truth sentence distributes more evenly. For graph-based unsupervised baselines, HIPORANK (Dong et al., 2020) achieves state-of-the-art performance that could even compete with some supervised methods. This demonstrates the significance of incorporating discourse structural information when modeling cross-sentence relations for long documents. In general, neural extractive methods perform better than abstractive methods due to the extended context. Among extractive baselines, transformer-based methods like SentPTR and HiStruct+ show substantial performance gain, demonstrating the effectiveness of the attention mechanism. HiStruct+ achieves strong performance by injecting inherent hierarchical structures into large pre-trained language models Longformer. In contrast, our model HEGEL only relies on hypergraph transformer layers for sentence representation learning and requires no pre-trained knowledge.

As shown in Table 2, HEGEL outperforms state-of-the-art extractive and abstractive baselines on both datasets. The supreme performance of HEGEL shows hypergraphs' capability of modeling high-order cross-sentence relations and the importance of fusing both semantic and structural information. We conduct an extensive ablation study and performance analysis next.

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

Table 3: Ablation study results on PubMed dataset.

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

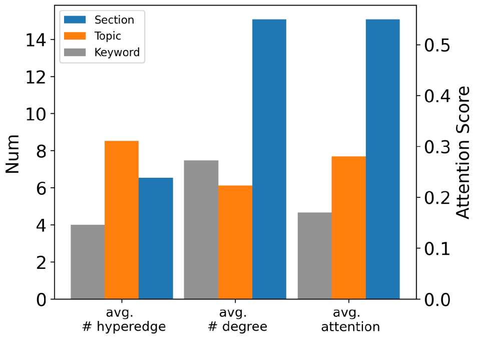


Figure 3: Average attention distribution over three types of hyperedges on PubMed dataset.

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.

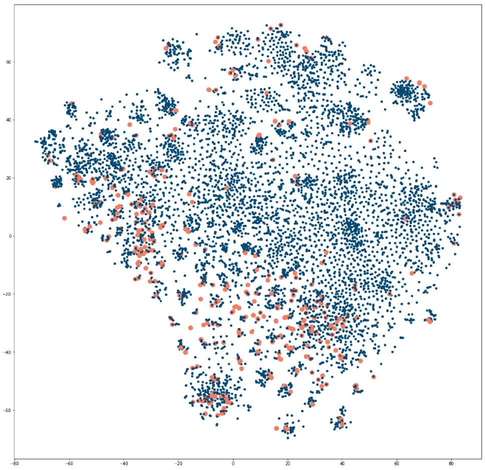


Figure 4: Visualization of sentence nodes embeddings for 100 documents in PubMed test set.

5.3 Embedding Analysis

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**) [Information] Additional information regarding 2 persons with asymptomatic merscov **infection** and other persons tested in the study. (**Section 2**) [Information] Our findings provide further evidence that asymptomatic human **infections** can be caused by zoonotic transmission. (**Section 2**) [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] 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:1603.07252*.

Arman Cohan, Franck Dernoncourt, Doo 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 Grigorios 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 bottleneck 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.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 150–157.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2016a. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *arXiv preprint arXiv:1611.04230*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016b. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. *arXiv preprint arXiv:1802.08636*.

Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Qian Ruan, Malte Ostendorff, and Georg Rehm. 2022. Histruct+: Improving extractive text summarization with hierarchical structure information. *arXiv preprint arXiv:2203.09629*.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Sandeep Subramanian, Raymond Li, Jonathan Pilault, and Christopher Pal. 2019. On extractive and abstractive neural document summarization with transformer language models. *arXiv preprint arXiv:1909.03186*.

Frederick Suppe. 1998. The structure of a scientific paper. *Philosophy of Science*, 65(3):381–405.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. *arXiv preprint arXiv:2004.12393*.

Lu Wang and Claire Cardie. 2013. Domain-independent abstract generation for focused meeting summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1395–1405.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. *arXiv preprint arXiv:1909.08089*.

Wen Xiao and Giuseppe Carenini. 2020. Systematically exploring redundancy reduction in summarizing long documents. *arXiv preprint arXiv:2012.00052*.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Discourse-aware neural extractive model for text summarization. *arXiv preprint arXiv:1910.14142*.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. *arXiv preprint arXiv:1706.06681*.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34.

Haopeng Zhang, Semih Yavuz, Wojciech Kryściński, Kazuma Hashimoto, and Yingbo Zhou. 2022. Improving the faithfulness of abstractive summarization via entity coverage control. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 528–535.

Haopeng Zhang and Jiawei Zhang. 2020. Text graph transformer for document classification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jiawei Zhang, Haopeng Zhang, Li Sun, and Congying Xia. 2020. Graph-bert: Only attention is needed for learning graph representations. *arXiv preprint arXiv:2001.05140*.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. *arXiv preprint arXiv:1906.03508*.