Expert Systems With Applications 210 (2022) 118335

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

HetTreeSum: A Heterogeneous Tree Structure-based Extractive Summarization Model for Scientific Papers

Jintao Zhao [a,1], Libin Yang [b,1], Xiaoyan Cai [a,*]

[a] School of Automation, Northwestern Polytechnical University, Xi'an, Shaanxi 710129, China
[b] School of Cyber Science and Technology, Northwestern Polytechnical University, Xi'an, Shaanxi 710129, China

ARTICLE INFO

Keywords:
Scientific paper summarization
Heterogeneous tree structure
Inter-sentence relations
Structural information
Iterative updating strategy

ABSTRACT

Scientific paper summarization aims at generating a short and concise digest while preserving important information of the original document. Currently, scientific paper summarization faces two main challenges. First, inter-sentence relations are hard to learn, especially in the case of long-form scientific papers. Second, structural information of the well-structured scientific papers has not been fully exploited. To overcome the above two challenges, we propose a novel **Het**erogeneous **Tree** structure-based extractive **Sum**marization (**HetTreSum**) model, where each document is modeled as a tree structure to learn inter-sentence relations and structural information of the original document is incorporated, enabling the tree structure to have a global perspective of the whole document. Then an iterative updating strategy is presented to interactively refine nodes of the tree structure for better contextualized representations, which can further enhance summarization performance. Experimental results on PubMed and arXiv datasets show that our proposed HetTreeSum model achieves significantly advanced performance compared with various scientific paper summarization models.

1. Introduction

Document summarization aims at generating a short and coherent summary of a given text. Creating summaries by hand is a costly and time-consuming task, thus automatic document summarization is a necessary technique to solve this problem. There are two major categories of document summarization: extractive approaches and abstractive approaches. Extractive summarization approaches generate summaries by extracting text from original documents, while abstractive summarization approaches rewrite documents by paraphrasing or deleting some words or phrases. In this paper, we focus on extractive document summarization.

Most recent works on neural extractive summarization have been rather successful in generating summaries of news articles by applying neural seq2seq models (Nallapati, Zhou, Gulcehre, Xiang, et al., 2016). However, summarizing long documents is a very different problem from newswire summarization. Scientific papers are an example of longer documents, the input text can range from 2000 to 7000 words, while in the case of news articles it rarely exceeds 700 words (Gidiotis & Tsoumakas, 2020). Moreover, news articles tend to be concise and contain one specific topic, while long documents are more sophisticated and typically cover multiple topics. In general, the longer a document is, the more topics are discussed. In reality, when humans write long documents, they organize them in chapters or sections. Take scientific papers as an example, they follow a standard discourse structure such as introduction, related work, methodology, experiments, results, and finally conclusions (Cohan et al., 2018). Besides, the expected summary of a news article is less than 100 words long, while the abstract of a scientific paper can easily exceed 200 words. Thus existing neural extractive summarization approaches could not be directly applied to scientific papers.

Collins, Augenstein, and Riedel (2017) are the first to leverage section information for extractive summarization of scientific papers. The section information fed into the sentence classifier is a categorical feature. Xiao and Carenini (2019) incorporated local context within each topic, along with global context of the whole document for extractive summarization of long documents. However, the combination method of local and global context is rigid and lacks flexible interaction between the two parts. Liu et al. (2021) treated tokens, entities and sentences as different types of nodes in the construction of a heterogeneous graph. Nevertheless, structure information is ignored, which may be inherently defective when applied to summarize highly structured scientific papers.

In this study, we propose **HetTreeSum**, a **Het**erogeneous **Tree** structure-based extractive **Sum**marization model for scientific papers.

* Corresponding author.
E-mail addresses: jtzhao@mail.nwpu.edu.cn (J. Zhao), libiny@nwpu.edu.cn (L. Yang), xiaoyanc@nwpu.edu.cn (X. Cai).
[1] These authors have contributed equally to this work.

https://doi.org/10.1016/j.eswa.2022.118335
Received 10 April 2022; Received in revised form 15 July 2022; Accepted 30 July 2022
Available online 5 August 2022

Specifically, we first construct a heterogeneous document tree, where each word in a scientific paper is used as a basic semantic unit. We deem each word as a leaf node of the tree structure, sentences and sections as branch nodes of the tree structure, respectively. Each word node is connected to the sentence node in which it occurs, and each sentence node connects to a section node which it belongs to, but there are no connections between nodes of the same type. The root of the heterogeneous document tree can be viewed as a global node representing the whole scientific paper. Thus both raw text information and the hierarchical structure of the document can be captured in the constructed document tree. After that, all the nodes in the constructed heterogeneous document tree will be updated iteratively according to the connection relationship to get enriched node representations. Finally, the concatenation of the global node and each sentence node's representation will be used to conduct extractive summarization. To the best of our knowledge, we are the first to introduce tree structure in extractive summarization for scientific papers, which naturally retains hierarchical structure of the original document. We hope it may shed light on the evolution of summarization research.

The three main contributions of the paper are:

(1) A heterogeneous document tree is constructed to capture both raw text information and the hierarchical structure of a scientific paper.

(2) A novel iterative updating strategy is developed, which involves both Bottom-Up and Top-Down process in dynamically propagating information over the tree structure.

(3) Thorough experimental studies on two widely used scientific paper summarization datasets show that our proposed HetTreeSum model achieves state-of-the-art performance.

2. Related work

In this paper, we propose a heterogeneous tree structure-based extractive summarization model for scientific papers. In this section, we first review previous extractive summarization works, which mainly focus on graph-based approaches and recent pretrained language models based approaches. Then we make a comprehensive review of works related to extractive summarization on scientific papers.

2.1. Extractive summarization

Traditional extractive summarization approaches use feature-based methods to rank sentences for their salience, in order to identify the most important sentences in a document or a set of documents (Filatova & Hatzivassiloglou, 2004; Jones, 2007; Kupiec, Pedersen, & Chen, 1995; Nenkova, Vanderwende, & McKeown, 2006; Radev, 2004). The centroid-based approach (Radev, Jing, Styś, & Tam, 2004) was among the most popular feature-based methods. Other statistical features and linguistic features, such as term frequency, sentence position, and sentence dependency structure have also been extensively investigated in the past.

Graph-based approaches have shown promising results for text summarization. Examples of graph-based text summarization include LexRank (Erkan & Radev, 2004) and TextRank (Mihalcea, 2005; Mihalcea & Tarau, 2004), which modeled a document or a set of documents as a homogeneous text graph by taking sentences as vertices and the similarities between sentences as edge weights. During calculation of sentence significance, they further considered global information instead of simply utilizing unconnected sentences over the text graph. Some researchers model documents as heterogeneous graphs to further enhance the performance of extractive summarization. Wei (2012) proposed to model a document as a heterogeneous graph that consists of three types of nodes, i.e., word nodes, sentence nodes and topic nodes. Then he applied a ranking algorithm to calculate scores of nodes and select the sentences with the highest scores as the summary. Wang, Liu, Zheng, Qiu, and Huang (2020) constructed a heterogeneous word-sentence graph and developed a heterogeneous graph-based neural network for extractive summarization.

Recently, pre-trained language models based on Transformer (Vaswani et al., 2017) (e.g. BERT Devlin, Chang, Lee, and Toutanova (2018)) breathed new life into text summarization, they have led to significant performance improvements on CNN/Daily Mail benchmark dataset (Zhang, Wei, & Zhou, 2019; Zhong, Liu, Wang, Qiu, & Huang, 2019). Liu and Lapata (2019) proposed a novel document-level encoder for document modeling based on BERT. Xu, Gan, Cheng, and Liu (2020) presented a discourse-aware summarization model DiscoBert, which extracts text spans rather than sentences with the representations obtained from BERT. Notably, Zhong et al. (2020) regarded extractive summarization as a semantic text matching problem and proposed a Siamese-BERT architecture to implement their model. However, due to the input length limitation of standard BERT, most of these models are poor in generalization and cannot be adapted to long documents effectively.

2.2. Extractive summarization on scientific papers

Most of the extractive summarization approaches are mainly focused on general domain summarization and news articles. Differences between scientific papers and news articles lie in multiple aspects, such as length, complexity and structure (Teufel & Moens, 2002). Thus it is inappropriate to directly apply the above-mentioned approaches in summarizing scientific papers. Teufel and Moens (2002) are the first to explore summarization of scientific papers. They trained a supervised naive Bayes classifier to construct a summary by selecting informative content from the document. Contractor, Guo, and Korhonen (2012) investigated effectiveness of introducing annotated argumentative zones (Guo, Korhonen, & Poibeau, 2011) in generating extractive summaries for scientific papers. Liakata, Dobnik, Saha, Batchelor, and Schuhmann (2013) used the scientific discourse to create a content model for extractive summarization.

Some works exploited citation contexts for scientific paper summarization. Qazvinian et al. (2013) presented a C-LexRank model, which shows the usefulness of extracting salient sentences from citations in automatic creation of technical summaries. Cohan and Goharian (2015) took advantage of both the discourse structure of scientific papers and extracted citation contexts from reference papers to conduct scientific paper summarization. Cohan and Goharian (2018) also extracted citation contexts from reference papers, which are then combined with the discourse structure of scientific documents to extract summary-worthy sentences. Similarly, Zerva, Nghiem, Nguyen, and Ananiadou (2020) focused on identification of citation text spans and used pre-trained encoders to generate summaries. Some works focus on generating surveys of scientific papers automatically. Mohammad et al. (2009) used citations to generate a technical survey on a given topic from multiple research papers. Jha, Coke, and Radev (2015) presented Surveyor, which combines a content model with a discourse model to generate coherent and readable surveys of scientific papers. Wang, Zhang, Zhang, and Deng (2018) utilized citing sentences in the original papers to generate surveys.

Other researchers exploited the unique properties of scientific documents such as long length and structure to summarize scientific papers. Conroy and Davis (2018) exploited the well-organized structure of scientific papers, and proposed a section mixture models approach. The approach is an advanced version of a bigram mixture model, which is based on important sections of the original document and sentences from papers referring to the current document. Xiao and Carenini (2019) incorporated both local and global context to their neural extractive model for summarizing long documents. Despite their efforts in summarizing scientific papers, how to effectively integrate structure information and content information of scientific papers still remains an unsolved problem.

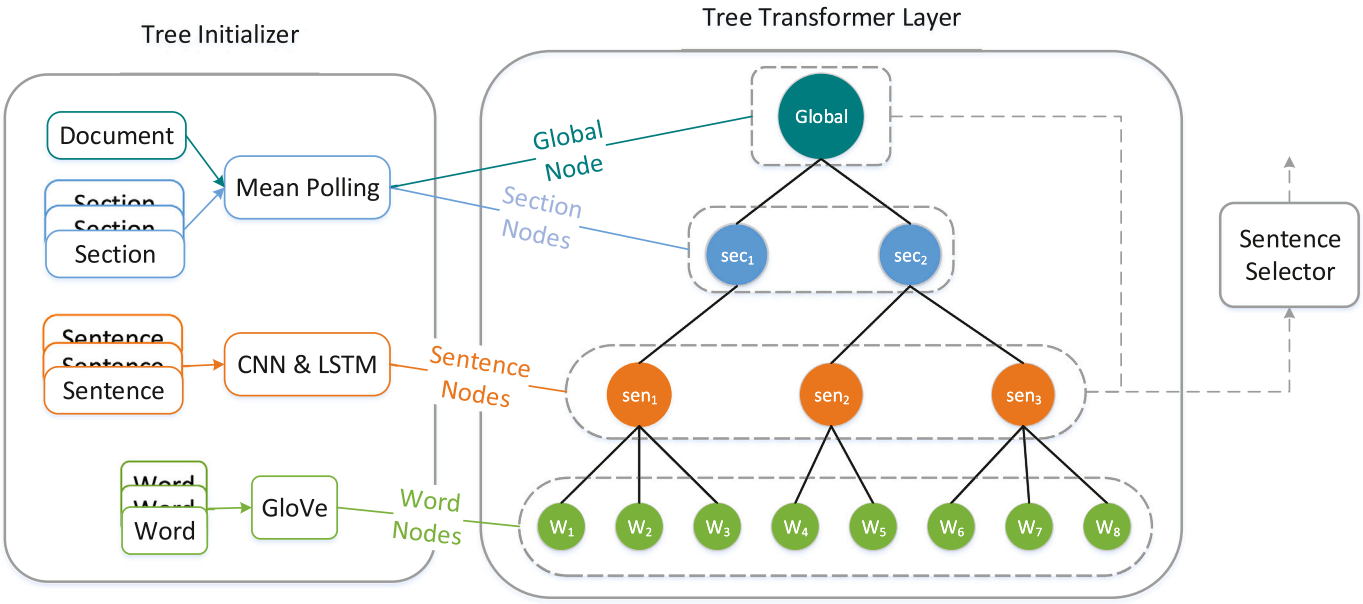


Fig. 1. The whole architecture of HetTreeSum. Units in color green, orange and blue represent for word nodes, sentence nodes and section nodes, respectively. In particular, root of the tree structure in color dark cyan refers to the global node.

3. Methodology

3.1. Problem formulation

Given a document D consisting of n sentences $D = \{s_1, s_2, \ldots, s_n\}$, we treat extractive summarization as a classification task. We aim at labeling each sentence s_i in D with a label $y_i \in \{0, 1\}$, where $y_i = 1$ indicates that s_i should be included in the summary and 0 otherwise.

Generally speaking, the nodes of our heterogeneous summarization tree structure can be divided into two granularities, one is fine-grained nodes containing words, the other is coarse-grained nodes containing sentences, sections and the document. Each sentence node connects with basic word nodes contained in it, each section node connects with sentence nodes contained in it, and the document node connects with all the section nodes, respectively. Thus, sentence nodes can establish relationships between each other via high-level discourse nodes (i.e. section nodes, document node).

3.2. Document as a heterogeneous tree structure

As mentioned above, scientific papers are usually organized in a fixed structure (i.e., introduction, related work, methodology, experiments, results and conclusion), for sentences in each section as well as words in each sentence, they are in a relationship of the latter containing the former. We treat words contained in a sentence as children of the sentence, and the section that includes this sentence as its parents, which is most similar to a tree structure. Therefore, we model a scientific paper as a heterogeneous tree structure which is a hierarchical data structure.

Specifically, we model sections of a scientific paper as section nodes, for sentences in each section, we add them as children nodes of the corresponding section node. Each sentence node is also the parents node of word nodes (leaf nodes) which are contained in the sentence. Finally, an extra global node which represents the whole scientific paper is introduced as the root of the tree, forming a complete tree structure.

Given a tree $T = \{V, E\}$, where V stands for a node set and E denotes an edge set, our constructed heterogeneous document tree can be formally defined as $V = V_w \cup V_{sen} \cup V_{sec} \cup V_G$ and $E = \{e_{ij}\}$. Here, $V_w = \{w_1, w_2, \ldots, w_m\}$ refers to words that appear in the document, $V_{sent} = \{sent_1, sent_2, \ldots, sent_n\}$ represents n sentences, $V_{sec} = \{sec_1, sec_2, \ldots, sec_p\}$ corresponds to p corresponding sections and $V_G = \{global\}$ denotes the document. E is the set of edges in our tree structure, where $e_{ij} = 1$ indicates that there is a connection between node i and j, otherwise $e_{ij} = 0$.

Fig. 1 illustrates the overall architecture of our proposed HetTreeSum model, which is composed of three parts: *Tree Initializer*, *Tree Transformer Layer* and *Sentence Selector*. *Tree Initializer* is intended for modeling the document as a tree structure as well as initializing all nodes in the tree, including word nodes, sentence nodes, section nodes and a global node. Note that if the same word appears in two different sentences, we add two separate word nodes to enable the model to learn semantic information in different contexts. Then the *Tree Transformer Layer* updates each node representation iteratively by attending over other nodes connected to it. Finally, concatenations of the global node and each sentence node representation obtained from the last tree transformer layer are fed to *Sentence Selector* to conduct summary selection.

3.3. Tree initializer

Let $\mathbf{F}_w \in \mathbb{R}^{m \times d_w}$, $\mathbf{F}_{sen} \in \mathbb{R}^{n \times d_{sen}}$, $\mathbf{F}_{sec} \in \mathbb{R}^{p \times d_{sec}}$ and $\mathbf{F}_{glob} \in \mathbb{R}^{1 \times d_{glob}}$ represent the input feature matrix of word nodes, sentence nodes, section nodes and a global node, respectively, d_w, d_{sent}, d_{sec} and d_{glob} denote the dimension of corresponding node representation vector.

Specifically, we use an unsupervised learning algorithm GloVe (Pennington, Socher, & Manning, 2014) to initialize word embeddings. For each sentence s_j, we first use Convolutional Neural Networks (CNN) (LeCun, Bottou, Bengio, & Haffner, 1998) with different kernel sizes to get local n-gram feature $local_j$, and then use Bidirectional Long Short-Term Memory Network (BiLSTM) (Hochreiter & Schmidhuber, 1997) to get sentence-level feature $global_j$. After that, we concatenate both local n-gram feature and global sentence-level feature as the initial representation of sentence node $F_{s_j} = [local_j : global_j]$. As for section nodes, we take the mean pooling of initial representations of sentences contained in each section as its initial representation. And the same goes for the global node.

3.4. Tree Transformer Layer

Given a constructed document tree T with node features $\mathbf{F}_w \cup \mathbf{F}_{sen} \cup \mathbf{F}_{sec} \cup \mathbf{F}_{glob}$, We iteratively update the representation of each node in our heterogeneous document tree structure with a transformer-style architecture, which we call Tree Transformer. For a specific node in our model, $\mathbf{h}_i \in \mathbb{R}^{d_h} (i \in \{1, \ldots, (m+n+p+1)\})$ refers to its hidden state.

3.4.1. Tree Transformer

In general, our tree transformer includes three operation steps: Mutual Attention, Message Extraction and Message Fusion. For simplicity, we denote the current node that needs to be updated as the central node, nodes connected to the central node as source nodes. In the following, we will illustrate the three steps in detail.

(1) Mutual Attention

In the first step, we estimate the relative importance of each source node to the central node by conducting mutual attention between the central node and source nodes. The self-attention based vanilla transformer model is proposed by Vaswani et al. (2017) in 2017. The famous self-attention mechanism provides a novel way of interacting and enables the model more dedicated to important information among multiple inputs. The vanilla transformer model achieves great success in various areas, including NLP, CV and even graphs. Inspired by this, we map the central node and source nodes to a query vector and key vectors, respectively. Then we compute dot products of the query and each key vector, multiplied by a scaling factor $1/\sqrt{d}$ (d is the dimension of the query vector) to impede gradient flow, and apply a softmax function to get the weight of corresponding key vectors.

We adopt multi-head attention mechanism with k heads to enable the model to attend over different representation subspaces, which is similar to different convolution kernels in CNN. Specifically, for the i-th attention head $AttHead^i(c, s)$ related to the central node c and a source node s, we first project c to a query vector $\mathbf{Q}^i(c)$ with a linear projection: $Linear^i_Q : \mathbb{R}^{d_h} \to \mathbb{R}^{\frac{d_h}{k}}$, where $\frac{d_h}{k}$ is the dimension of each query vector. And then we generate key vector $\mathbf{K}^i(s)$ with another linear projection $Linear^i_K : \mathbb{R}^{d_h} \to \mathbb{R}^{\frac{d_h}{k}}$ as:

$$\mathbf{Q}^i(c) = Linear^i_Q(\mathbf{h}_i(c)) \tag{1}$$

$$\mathbf{K}^i(s) = Linear^i_K(\mathbf{h}_i(s)) \tag{2}$$

Then we calculate the attention score between central node c and source node s as follows:

$$Attention(c, s) = \mathrm{softmax}\Big(\big[AttHead^i(c, s), \ldots, AttHead^k(c, s)\big]\Big) \tag{3}$$

$$AttHead^i(c, s) = \frac{\mathbf{Q}^i(c)\mathbf{K}^i(s)^T}{\sqrt{d}} \tag{4}$$

where $[\cdot]$ denotes the concatenation operation.

(2) Message Extraction

Message extraction process extracts information to pass from source nodes to the central node, which is in parallel with the calculation of mutual attention. We calculate the multi-head message $Message(c, s)$ through a concatenation operation over all attention heads as:

$$Message(c, s) = \big[MesHead^i(c, s), \ldots, MesHead^k(c, s)\big] \tag{5}$$

$$MesHead^i(c, s) = Linear^i_V(\mathbf{h}_i(s)) \tag{6}$$

where $MesHead^i(c, s)$ refers to the extracted message from source node s of the attention head with a linear projection $Linear^i_V : \mathbb{R}^{d_h} \to \mathbb{R}^{\frac{d_h}{k}}$.

(3) Message Fusion

Message fusion step aims to get a contextualized node representation via aggregating the neighborhood message by corresponding attention weight. Attention score obtained from the mutual attention stage implies the importance of source nodes relative to the central node. Thus with the extracted message, we calculate the updated representation of the central node as:

$$\tilde{\mathbf{h}}_i(c) = \sum_{\forall s \in N'(c)} \Big(Attention(c, s) \cdot Message(c, s)\Big) \tag{7}$$

where $N'(c)$ denotes a subset of source nodes connected to node c, the subset varies according to different updating stages described in Section 3.4.2.

And then a non-linear activation function σ is applied to $\tilde{\mathbf{h}}_i(c)$, followed by a residual connection to avoid gradient vanishing:

$$\hat{\mathbf{h}}_i(c) = \sigma(\tilde{\mathbf{h}}_i(c)) + \mathbf{h}_i(c) \tag{8}$$

Next, we apply layer normalization and a position-wise feedforward network(FFN) to get the final representation of the central node as follows:

$$\mathbf{h}''_i(c) = LayerNorm\Big(\mathbf{h}'_i(c) + \hat{\mathbf{h}}_i(c)\Big) \tag{9}$$

$$\mathbf{h}'_i(c) = FFN\Big(\hat{\mathbf{h}}_i(c)\Big) \tag{10}$$

In a nutshell, the updating process of central node c utilizing source nodes with aforementioned operations (denoted as $TTL(\cdot)$) can be simply formulated as:

$$\mathbf{h}''_i(c) = TTL\Big(\mathbf{h}_i(c), \mathbf{h}_i(s)\Big) \tag{11}$$

3.4.2. Iterative updating

The message passing process in our heterogeneous document tree structure is illustrated in Fig. 2. In general, we follow a Bottom-Up and Top-Down (BUTD) iterative updating strategy, which will be described in detail as follows.

More specifically, for the t-th iteration, we first update node representations from leaf nodes to root node in the heterogeneous document tree structure. Take the updating process of sentence nodes for example, Fig. 2(a) shows that sentence nodes gather fine-grained semantic information from connected word nodes with graph transformer network, to get enriched sentence representations. Similarly, representations of section nodes and the global node are updated sequentially. We call the above procedure as Bottom-Up updating stage.

And then we go backward from the root node to leaf nodes, which is called the Top-Down updating stage in our study. In this stage, we still take the updating of sentence nodes as an instance to illustrate the difference between two updating stages. Sentence nodes receive high-level coarse-grained information from section nodes to get updated sentence representations, as shown in Fig. 2(b). As a whole, representations of section nodes, sentence nodes and word nodes get updated in sequence during Top-Down updating stage.

Concretely speaking, the whole updating process consists of two stages. For the t-th iteration, the updating process can be represented as follows:

Bottom-Up:

$$\mathbf{H}^t_{sen\_up} = TTL(\mathbf{H}^{t-1}_w, \mathbf{H}^{t-1}_{sen\_down}) \tag{12}$$

$$\mathbf{H}^t_{sec\_up} = TTL(\mathbf{H}^t_{sen\_up}, \mathbf{H}^{t-1}_{sec\_down}) \tag{13}$$

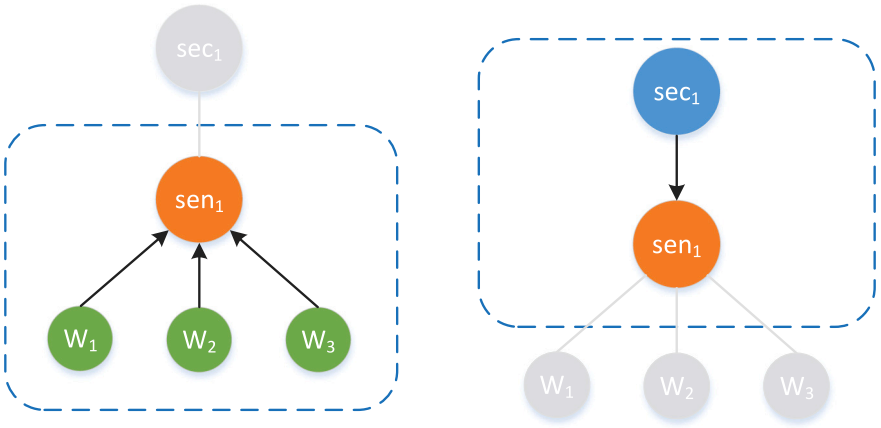
$$\mathbf{H}^t_{glob} = TTL(\mathbf{H}^{t-1}_{glob}, \mathbf{H}^t_{sec\_up}) \tag{14}$$

Top-Down:

$$\mathbf{H}^t_{sec\_down} = TTL(\mathbf{H}^t_{glob}, \mathbf{H}^t_{sec\_up}) \tag{15}$$

$$\mathbf{H}^t_{sen\_down} = TTL(\mathbf{H}^t_{sec\_down}, \mathbf{H}^t_{sen\_up}) \tag{16}$$

$$\mathbf{H}^t_w = TTL(\mathbf{H}^t_{sen\_down}, \mathbf{H}^{t-1}_w) \tag{17}$$



(a) Update sen_1 in Bottom-Up (b) Update sen_1 in Top-Down

Fig. 2. The detailed process of our Bottom-Up and Top-Down iterative updating strategy, which takes the updating process of sentence node sen_1 in different stages for example. 2(a) shows that sen_1 aggregates message flow by attending over all connected word nodes in the heterogeneous document tree structure during the Bottom-Up updating stage. In 2(b), the update of sen_1 follows a Top-Down strategy, which gathers information from section node utilizing the representation obtained from previous updating steps.

where $\mathbf{H}^0_{sen\_up} = \mathbf{H}^0_{sen\_down} = \mathbf{F}_{sen}$, $\mathbf{H}^0_{sec\_up} = \mathbf{H}^0_{sec\_down} = \mathbf{F}_{sec}$, $\mathbf{H}^0_{w} = \mathbf{F}_{w}$ and $\mathbf{H}^0_{glob} = \mathbf{F}_{glob}$.

Through iterative updating process, sentence representations not only contain sentence information itself, but also integrate fine-grained semantic information and coarse-grained structural information. In this manner, semantically reinforced and structure enhanced sentence representations $\mathbf{H}_{sen}$ are obtained for downstream summarization task.

3.5. Sentence selector

Once the sentence representations are obtained from the aforementioned method, we need to select sentences as the final summary. We treat extractive summarization as a sentence classification task and label sentences with 0 and 1. The training objective of our summarization system is to minimize the binary cross-entropy loss function as:

$$L = -\sum_{i=1}^{n} \left(y_i log(\tilde{y}_i) + (1 - y_i) log(1 - \tilde{y}_i) \right) \tag{18}$$

where $\tilde{y}_i$ represents the predicted label, y_i is the ground truth sentence label. The ground truth sentences, also known as ORACLE, are generated by extracting sentences from the original document using the greedy algorithm introduced by Kedzie, Mckeown, and Daumé III (2018).

4. Experiments

We conduct thorough experiments to evaluate the performance of our proposed HetTreeSum model. Below, we start with the description of the datasets used in our experiments.

4.1. Datasets

Our heterogeneous document tree structure is particularly designed for extractive summarization on highly-structured scientific papers. So we conduct experiments on scientific paper datasets instead of relatively short news articles (e.g., CNN, Daily Mail and New York Times articles), since short texts lack rich structural information compared with long-form scientific papers. We use two large-scale scientific paper datasets, i.e., PubMed and arXiv (Cohan et al., 2018), to evaluate our proposed HetTreeSum model. Table 1 shows statistical analysis of PubMed and arXiv datasets.

As shown in Table 2, we follow the original split of the two datasets in our research. On the basis of this, we further remove excessively long or too short (e.g. number of sentences less than 6, where the number 6 is defined according to basic sections (Introduction, Related work, Method, Experiments, Results and Analysis, Conclusion) contained in most papers) documents. We train our proposed model on the training set, and select the best models for testing from saved checkpoints by evaluating them on the validation set. Once the best models are obtained, we run them on the test set to get their final performance and report their experimental results in Section 5.

Table 1
Statistics of PubMed and arXiv datasets.

Datasets	#Doc	Avg.doc length		Avg.summary length	
		Words	Sentences	Words	Sentences
PubMed	133k	3016	88	203	7
arXiv	215k	4938	206	220	10

Table 2
Data split of the two datasets.

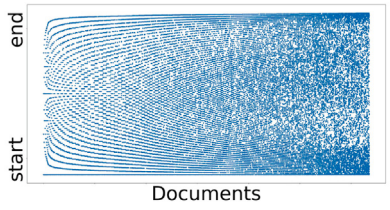
Datasets	Train	Validation	Test	Total
PubMed	119,924	6633	6658	133,215
arXiv	203,037	6436	6440	215,913

Fig. 3 shows relative position distribution of oracle sentences of the two datasets. It is easy to observe that oracle sentences are distributed across the document, which requires summarization models having robust abilities to capture long-range dependencies while understanding the whole document. More importantly, a large percentage of oracle sentences cannot be accessed by most state-of-the-art summarization models based on BERT, according to the input limitation 512 of the standard BERT model.

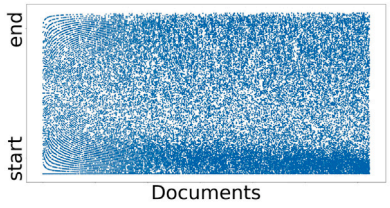
4.2. Implementation details

For tree initialization, we set vocabulary size to 50,000 for the two datasets and use an unsupervised learning algorithm GloVe (Pennington et al., 2014) to initialize word nodes with a dimension of 300. Sentence nodes are initialized with the dimension of $d_{sent} = 128$. There are 8 attention heads in our graph transformer layer, where the hidden size is set to $d_h = 64$. The iteration number is set to 3 according to model performance on the validation set (see Table 7 for details). For FFN, the hidden size is set to 512 in our experiments.

We train our model on a single Tesla-V 100 GPU with a small batch size of 32. During the training process, Adam optimizer (Kingma & Ba, 2014) is used to optimize parameters in our model with an initial



(a) PubMed



(b) arXiv

Fig. 3. Relative position distribution of oracle sentences in source documents on the validation set of the two datasets. Documents on the x-axis are ordered by increasing article length from the shortest to the longest. As for the y-axis, "start" and "end" corresponds to the beginning and ending of the source document, respectively.

learning rate of 5e−4. The number of training epoch is set to 20 and an early stop strategy (Caruana & Lawrence, 2001) is applied when the validation loss does not descend in three continuous epochs. Following Grail, Perez, and Gaussier (2021), the number of extracted sentences is set to 7 for PubMed dataset and 6 for arXiv dataset.

4.3. Comparison models

We conduct a systematic comparison with existing extractive summarization models for long texts. Besides this, several abstractive models are also compared for completeness.

EXT-Summ-LG (Xiao & Carenini, 2019): A recent extractive summarization model for long documents. It takes both local and global information into consideration and encodes them jointly to conduct extractive summarization.

SummaRuNNer (Nallapati, Zhai, & Zhou, 2017): A sequence model based on recurrent neural network for extractive summarization.

Match-Sum (Zhong et al., 2020): A state-of-the-art BERT-based extractive summarization approach, which formulates extractive summarization as a semantic text matching problem, where candidate summaries and the source document are matched in a semantic space.

Topic-GraphSum (Cui, Hu, & Liu, 2020): A graph neural network-based extractive summarization framework, where latent topics can be learned and utilized to enrich sentence representations via a joint neural topic model.

HEROES (Zhu, Hua, Qu, & Zhou, 2021): A recent graph-based summarization model that exploits the effect of rich discourse structural information in summarizing long-form documents.

Pointer Generator Network (PGN) (See, Liu, & Manning, 2017): An abstractive summarization model, which is a hybrid pointer-generator network combining copy mechanism and coverage mechanism.

Discourse-Aware (Cohan et al., 2018): A neural abstractive summarization model specially designed for long-form documents with discourse structure.

BERTSUMEXT(SW) (Grail et al., 2021): A varient of BERTSUMEXT model, which performs a sliding window strategy on the original document to get token representations for summarization.

4.4. Evaluation metrics

The widely used automatic summarization evaluation metric ROUGE (Lin, 2004) is chosen to evaluate the performance of our summarization model HetTreeSum. Following the common practice, we report F1 scores of ROUGE-1, ROUGE-2 and ROUGE-L to balance the precision and recall, where ROUGE-1 and ROUGE-2 measures informativeness by computing n-gram overlaps between system summaries and reference summaries while ROUGE-L measures fluency through the longest common subsequence. The calculation of ROUGE-N is formulated as:

$$ROUGE-N = \frac{\sum_{S\in\{ReferenceSummaries\}}\sum_{gram_n\in S} Count_{match}\left(gram_n\right)}{\sum_{S\in\{ReferenceSummaries\}}\sum_{gram_n\in S} Count\left(gram_n\right)} \tag{19}$$

where n stands for the length of the n-gram: $gram_n$, and $Count_{match}\left(gram_n\right)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries.

5. Results and analysis

In this section, we showcase the performance of our HetTreeSum model, carry out extensive experiments and conduct comprehensive analysis to demonstrate the strength and effectiveness of our model, including the overall performance, ablation study, hyperparameter tuning and case study.

5.1. Overall performance

Table 3 shows overall performance of different models on the test set of the two datasets. As shown in the table, there are four blocks showing comparison models and our proposed HetTreeSum model. Specifically, the first block presents unsupervised summarization models. The second and third blocks show the results of abstractive and extractive models, respectively. Finally, the performance of our model is reported in the last block.

For unsupervised summarization models, our model beats them by a large margin in terms of informativeness (evaluated by ROUGE-1 and ROUGE-2) and fluency (evaluated by ROUGE-L) on both datasets. We attribute it to the fact that our HetTreeSum model is trained in a supervised manner utilizing ORACLE summaries generated by the greedy algorithm, which leads to higher performance than unsupervised models. Notably, even compared with the state-of-the-art unsupervised model HIPORANK, which also leverages the discourse structure of scientific papers to enhance summarization performance, HetTreeSum still achieves a remarkable improvement of 5.22/6.09/5.16 on PubMed and 8.90/8.85/8.36 on arXiv in terms of ROUGE-1/2/L F1 scores.

As shown in the second block, among the listed abstractive summarization models, our proposed HetTreeSum model has better performance of all ROUGE metrics on both datasets, even when it comes to strong baselines: HAT-BART and DANCER-PEGASUS, which is based on the standard transformer architecture and powerful pre-trained language model, respectively. This can be due to that HAT-BART fails to consider structural information and DANCER-PEGASUS ignores the interaction between sections.

Compared with extractive summarization baselines reported in the third block, one can observe that our HetTreeSum model surpasses almost all other models in terms of all ROUGE metrics on both datasets, except for the BERT-based model TopicGraphSum-BERT in terms of ROUGE-1 score on the PubMed dataset. TopicGraphSum leverages a neural topic model to learn latent topics but essentially ignores the inherent discourse structure of long documents. However, HetTreeSum outperforms its non-BERT-based counterpart Topic-GraphSum-BiGRU by 2.67 of ROUGE-1 score on the PubMed dataset. This shows that HetTreeSum model is comparable and even better than pretrained language models that use large-scale corpora to train their models. Among

Table 3
ROUGE F1 scores on the corresponding test set of two datasets. Except for comparison models listed in Section 4.3, we also present results reported by previous works. Results with * and + are taken from Cohan et al. (2018) and Xiao and Carenini (2019), respectively. Results in bold font correspond to the highest score of the corresponding metric.

Models	PubMed			arXiv		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
Oracle	58.15	34.16	52.99	57.78	30.43	51.24
LSA+(2004)	33.89	9.93	29.70	29.91	7.42	25.67
SumBasic+(2007)	37.15	11.36	33.43	29.47	6.95	26.3
LexRank+(2004)	39.19	13.89	34.59	33.85	10.73	28.99
HIPORANK(2021)	43.58	17.00	39.31	39.34	12.56	34.89
Att-Seq2seq+(2016)	31.55	8.52	27.38	29.30	6.00	25.56
PGN(2017)	35.86	10.22	29.69	32.06	9.04	25.16
Discourse-Aware+(2018)	38.93	15.37	35.21	35.80	11.05	31.8
PEGASUS(2019)	45.97	20.15	28.25	44.21	16.95	25.67
DANCER-PEGAS(2020)	46.34	19.97	42.42	45.01	17.60	40.56
HAT-BART(2021)	48.36	21.43	37.00	46.68	19.07	42.17
Ext-Summ-LG*(2019)	44.81	19.74	31.48	43.58	17.37	29.3
Sent-CLF(2019)	45.01	19.91	41.16	34.01	8.71	30.41
Sent-PTR(2019)	43.30	17.92	39.47	42.32	15.63	38.06
BERTSUMEXT(SW)(2019)	45.01	20.00	40.43	42.93	15.08	37.22
MatchSum(2020)	41.21	14.91	36.75	40.59	12.98	32.64
TopicGraphSum-BiGRU(2020)	46.13	20.91	33.27	44.71	18.84	32.58
TopicGraphSum-BERT(2020)	**48.85**	21.76	35.19	46.05	19.97	33.61
GBT-EXTSUM(2021)	46.87	20.19	42.68	48.08	19.21	42.68
HEROES(2021)	48.14	21.82	43.33	47.74	20.46	42.39
HetTreeSum	48.80	**23.09**	**44.47**	**48.24**	**21.41**	**43.25**

the selected models, Sent-CLF and Sent-PTR use a hierarchical structure that combines a token-level and sentence-level RNN as its encoder, where section-level and higher-level hierarchy structure is ignored. For ExtSumm-LG, encoding both local and global contexts with RNN indeed helps in summarization. However, it is very limited for RNNs to capture long-range dependencies and complex structural information of long-form documents. Moreover, interactions between local and global contexts in their architecture are not fully explored. GBT-EXTSUM uses a hierarchical propagation layer to spread information in multiple transformer windows, but fails to consider structural information of long documents.

While for our HetTreeSum model, structural information is incorporated into the tree structure, and interactions between different granularities of information in the document are considered.

5.2. Ablation study

In this part, we design and conduct experiments to analyze the contributions of different components in our summarization model. All ablation experiments are performed on the PubMed dataset. Except for the differences of each variant detailed below, all the hyperparameters and experimental settings are kept unchanged with respect to the implementation details described in Section 4.2.

5.2.1. Different document modeling approaches

To better understand whether the tree structure is more effective than the general graph structure or not, we design a Heterogeneous Graph Summarization (HetGraSum) model, which models the document as a heterogeneous graph. The difference between the heterogeneous graph and the tree structure lies in the edge connections between word nodes and sentence nodes. To be specific, for a word that occurs in two different sentences, the heterogeneous graph does not distinguish different contexts and adds two edge connections between the word node and two sentence nodes, while the tree structure is context-sensitive, indicating that the meaning of the same word in different sentences is different. Therefore, the tree structure treats the same word as different word nodes and adds an edge connection for each word-sentence node pair.

From Table 4, one can observe that our HetTreeSum model achieves higher scores on all ROUGE metrics than the HetGraSum model. This shows that our model benefits from modeling the document as a tree, which is more powerful and enables HetTreeSum to capture context-intensive word meanings for a better understanding of sentences and the whole document.

Table 4
Summarization performance using different document modeling approaches.

Model	ROUGE-1	ROUGE-2	ROUGE-L
HetGraSum	48.30	22.96	43.88
HetTreeSum	**48.80**	**23.09**	**44.47**

Table 5
Summarization performance with and without section information.

Model	ROUGE-1	ROUGE-2	ROUGE-L
HetTreeSum	**48.80**	**23.09**	**44.47**
w/o section information	48.37	22.81	43.98

5.2.2. Utility of structural information

In order to study the effectiveness of structural information (i.e. section nodes in our tree structure), we remove section nodes in the tree structure and keep other settings the same. In this set of experiments, we simply use the original division of each document (i.e. different sections organized by their authors) as structural information.

As shown in Table 5, removing structural information results in a dramatic decrease in model performance, which shows that the structural information counts in understanding well-structured scientific papers. Comparable results reported by models that make use of the discourse structure (such as HIPORANK, HEROES) also verify the importance of structural information in summarizing long documents.

5.2.3. Influence of different iterative updating strategies

To analyze the influence of different iterative updating strategies, we design experiments with two variants of HetTreeSum: (1) HetTreeSum *w/i* BU: substitute the BUTD updating strategy with Bottom-Up updating strategy. Different from vanilla HetTreeSum model, connections between the global node and all word nodes are added. And nodes in this scenario follow a fixed updating order of word→sent→sec→glob→word. (2) HetTreeSum *w/i* TD: substitute the BUTD updating strategy with Top-Down updating strategy, where

Table 6
Summarization performance using different updating strategies.

Model	ROUGE-1	ROUGE-2	ROUGE-L
HetTreeSum	**48.80**	**23.09**	**44.47**
HetTreeSum *w/i* BU	48.61	22.85	44.26
HetTreeSum *w/i* TD	47.95	22.70	43.57
w/o iterative updating	48.16	22.51	43.80

Table 7
Summarization performance using different iteration numbers of the graph transformer layer. Time refers to the average time cost of one single epoch.

Iteration number	ROUGE-1	ROUGE-2	ROUGE-L	Time
1	47.10	22.13	40.90	4.49 h
2	47.48	22.45	41.19	5.34 h
3	**47.67**	22.48	**41.31**	6.71 h
4	47.53	**22.61**	41.3	8.20 h
5	47.19	22.56	41.22	10.21 h

nodes in this scenario follow the updating order of glob→sec→sent→word→glob. In addition, we also conduct an experiment by setting the number of iteration to 0, which means that the iterative updating process in our HetTreeSum model is disabled and the initial representation of nodes are used for summarization.

Table 6 shows effectiveness of the proposed BUTD updating strategy for our HetTreeSum model. Intuitively, disabling the iterative updating process leads to dramatic drop of ROUGE scores on all metrics, which indicates that node representations are refined and dependencies between sentences can be captured through iteratively updating each other. Interestingly, the Top-Down updating strategy gets the lowest scores on all ROUGE metrics, while the combination of both Bottom-Up and Top-Down updating strategy (i.e. BUTD) produces the best performance. The order of node updating in the Bottom-Up updating strategy resembles the way the human understand a document. It first learns the basic semantic information of each word in a sentence, and then understands the meaning of each sentence. Similarly, the meaning of each section and the gist of the whole document can be learned step by step through the updating strategy. Conversely, the Top-Down updating strategy understands the whole document in an opposite way. Generally speaking, the Bottom-Up updating strategy understands the whole from the parts like RNNs, while the Top-Down updating strategy grasps the parts from the whole. The combination of the two complements each other, which is similar to Bidirectional RNNs and enables our model to have a deep and comprehensive understanding of the whole document.

5.3. Results on varying iteration numbers

In order to select a proper iteration number for our proposed HetTreeSum model, we compare the performance of the model when iteration number ranges from 1 to 5. All the models are trained on a single Tesla V100 32 GB GPU for 5 epochs. As Table 7 shows, the model achieves comparable results when iteration number is set to 3 and 4. However, when the number of iteration goes from 3 to 4, the performance is slightly boosted only in terms of ROUGE-2 F1 score. Additionally, the performance gain is not always proportionate to iteration number. As reflected in the table, larger iteration number just results in more time cost but less gain. To balance the effectiveness and performance of our model, the iteration number is set to 3 for the PubMed dataset. Similarly, according to the experimental results on the validation set of the arXiv dataset, the iteration number is also set to 3 for the dataset.

5.4. Case study

Table 8 provides a case study of a document from the PubMed test set, where the first block presents the gold summary, the second and third block reports the output summaries of our HetTreeSum model and BERTSUMEXT(SW), respectively. The number on the left of each sentence indicates its position in the original document that is composed of 78 sentences in total. The ROUGE score between each sentence and the gold summary is reflected in different shades of red (darker colors mean higher ROUGE scores). As can be seen from the table, our HetTreeSum model concentrates on the main parts (e.g. introduction, discussion) of the document and extracts summary sentences from these sections, while BERTSUMEXT(SW) mainly focuses on the beginning part rather than the whole document. Since the sliding window strategy of BERTSUMEXT(SW) is very limited to capture long range dependencies in complex scientific papers, it fails to understand both local and global information of the document simultaneously. From the perspective of informativeness, our HetTreeSum model tends to extract sentences with higher ROUGE scores, which means that these sentences are more meaningful and summary-worthy. For example, the sentence with the highest ROUGE score, with the position of 63 in the discussion section, is selected by our model. In contrast, sentences selected by BERTSUMEXT(SW) model are less informative and generally have lower ROUGE scores. The above analysis shows that our HetTreeSum model is more powerful and can generate high-quality summaries for scientific papers.

6. Conclusion

Extractive summarization of scientific papers has been a hot research spot in the area of text summarization. Recent years have witnessed a vast amount of works related to scientific paper summarization. However, how to utilize structural information of well-structured scientific papers effectively remains an open challenge. What is more, the meaning of word in different contexts tend to be overlooked in previous works. In this paper, we propose a heterogeneous tree structure-based extractive summarization model for scientific papers, which models each document as a more expressive tree structure and incorporates structural information simultaneously. Context-sensitive word meanings can also be learned through the tree structure. We also design a Bottom-Up-Top-Down updating strategy for the tree structure to refine node representations iteratively. Experimental results demonstrate effectiveness of our model in dealing with complex and lengthy scientific papers.

In the future, we plan to conduct an in-depth study of scientific paper summarization from two aspects. First of all, the source text information can be further explored. Besides basic words used in this research, other fine-grained semantic units can also be included in the tree structure, such as keywords, phrases and even titles. From another perspective, extra knowledge stored in knowledge graphs can also be introduced to assist the model in understanding domain-specific terminologies.

CRediT authorship contribution statement

Jintao Zhao: Software, Investigation, Writing – original draft. **Libin Yang:** Conceptualization, Writing – reviewing and editing, Funding acquisition, Supervision. **Xiaoyan Cai:** Methodology, Writing – reviewing and editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Table 8
An example of summaries generated by our HetTreeSum model and BERTSUMEXT(SW) model as well as the gold summary.

GOLD		Purpose : to investigate whether the glc3a locus harboring the cyp1b1 gene is associated with normal tension glaucoma (ntg) in japanese patients.materials and methods : one hundred forty-two japanese patients with ntg and 101 japanese healthy controls were recruited. Patients exhibiting a comparatively early onset were selected as this suggests that genetic factors may show stronger involvement. Genotyping and assessment of allelic diversity was performed on 13 highly polymorphic microsatellite markers in and around the glc3a locus.results:there were decreased frequencies of the 444 allele of d2s0416i and the 258 allele of d2s0425i in cases compared to controls (p = 0.022 and p = 0.034, respectively). However, this statistical significance disappeared when corrected (pc > 0.05). We did not find any significant association between the remaining 11 microsatellite markers, including d2s177, which may be associated with cyp1b1, and ntg (p > 0.05). Conclusions : our study showed no association between the glca3 locus and ntg, suggesting that the cyp1b1 gene, which is reportedly involved in a range of glaucoma phenotypes, may not be an associated factor in the pathogenesis of ntg.
HetTreeSum	0-	Glaucoma is a progressive optic neuropathy leading to permanent visual loss that is often associated with elevated intraocular pressure (iop).
	1-	Primary open angle glaucoma (poag) is the most common type of glaucoma.
	5-	Glaucoma is genetically heterogeneous and the detection of susceptibility genes could provide useful information for early diagnosis of glaucoma. to date, over 30 genetic loci for glaucoma have been identified by linkage analysis in multiple pedigrees ; 1012 14 loci of poag, 3 loci of primary congenital glaucoma (p...
	7-	Of these subjects, 142 were diagnosed with ntg, and 101 were control subjects.
	61-	We genotyped 13 polymorphic microsatellite markers in and around the glc3a locus in 142 patients and 101 controls (figure 1).
	63-	Only two adjacent markers, d2s0416i and d2s0425i, were significantly positive, as shown in (table 2), and the frequency of the 444 allele of d2s0416i and the 258 allele of d2s0425i were decreased in cases compared to controls (p = 0.022, or = 0.59 and p = 0.034, or = 0.42, respectively).
	65-	The magnitude of ld between these two markers was low, with pairwise d = 0.25, and the comparison of haplotype consisting of two alleles (d2s0416i_444 and d2s0425i_258) rendered no significant difference between cases and controls (cases vs. controls = 3.5% vs. 7.3%, p = 0.055) (data not shown).
BERTSUMEXT(SW)	1-	Primary open angle glaucoma (poag) is the most common type of glaucoma.
	2-	Normal tension glaucoma (ntg) is an important subset of poag ; while many poag patients have high iop,1 patients with ntg have statistically normal iop.24 the prevalence of ntg is higher among the japanese population than among caucasians, and recent studies reported that 92% of poag patients in japan had ntg.58 the diagnosis of glaucoma is based on a combination of factors including optic nerve damage and specific field defects for which iop is the only treatable risk factor.
	7-	Of these subjects, 142 were diagnosed with ntg, and 101 were control subjects.
	20-	Genomic dna was extracted using the qiaamp dna blood mini kit (qiagen, hilden, germany) or the guanidine method. In this association study, we selected 13 highly polymorphic microsatellite markers that are located in and around the glc3a locus as shown in (figure 1).
	22-	Polymerase chain reaction (pcr) was performed in a reaction mixture with a total volume of 12.5 1 containing pcr buffer, genomic dna, 0.2 mm dinucleotide triphosphates (dntps), 0.5 m primers, and 0.35 u taq polymerase.
	28-	The number of microsatellite repeats was estimated automatically using the genescan 672 software (applied biosystems) by the local southern method with a size marker of gs500 tamra (applied biosystems).

Acknowledgments

This work has been supported by National Natural Science Foundation of China (nos. 61872296, U20B2065), and MOE (Ministry of Education in China) Project of Humanities and Social Sciences (no. 18YJC870001).

References

Caruana, R., & Lawrence, S. (2001). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in neural information processing systems 13: Proceedings of the 2000 conference, Vol. 13* (p. 402). MIT Press.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., et al. (2018). A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of NAACL-HLT* (pp. 615–621).

Cohan, A., & Goharian, N. (2015). Scientific article summarization using citation-context and article's discourse structure. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 390–400).

Cohan, A., & Goharian, N. (2018). Scientific document summarization via citation contextualization and scientific discourse. *International Journal on Digital Libraries*, *19*(2), 287–303.

Collins, E., Augenstein, I., & Riedel, S. (2017). A supervised approach to extractive summarisation of scientific papers. In *Proceedings of the 21st conference on computational natural language learning (CoNLL 2017)* (pp. 195–205).

Conroy, J. M., & Davis, S. T. (2018). Section mixture models for scientific document summarization. *International Journal on Digital Libraries*, *19*(2), 305–322.

Contractor, D., Guo, Y., & Korhonen, A. (2012). Using argumentative zones for extractive summarization of scientific articles. In *Proceedings of COLING 2012* (pp. 663–678).

Cui, P., Hu, L., & Liu, Y. (2020). Enhancing extractive text summarization with topic-aware graph neural networks. In *Proceedings of the 28th international conference on computational linguistics* (pp. 5360–5371).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, *22*, 457–479.

Filatova, E., & Hatzivassiloglou, V. (2004). Event-based extractive summarization. In *Text summarization branches out* (pp. 104–111).

Gidiotis, A., & Tsoumakas, G. (2020). A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*, 3029–3040.

Grail, Q., Perez, J., & Gaussier, E. (2021). Globalizing BERT-based transformer architectures for long document summarization. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: Main volume* (pp. 1792–1810).

Guo, Y., Korhonen, A., & Poibeau, T. (2011). A weakly-supervised approach to argumentative zoning of scientific documents. In *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 273–283).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Jha, R., Coke, R., & Radev, D. (2015). Surveyor: A system for generating coherent survey articles for scientific topics. In *Twenty-ninth AAAI conference on artificial intelligence*.

Jones, K. S. (2007). Automatic summarising: The state of the art. *Information Processing & Management*, *43*(6), 1449–1481.

Kedzie, C., Mckeown, K., & Daumé III, H. (2018). Content selection in deep learning models of summarization. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 1818–1828).

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 68–73).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Liakata, M., Dobnik, S., Saha, S., Batchelor, C., & Schuhmann, D. R. (2013). A discourse-driven content model for summarising scientific articles evaluated in a complex question answering task. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 747–757).

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81).

Liu, Y., & Lapata, M. (2019). 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)* (pp. 3730–3740).

Liu, Y., Zhang, J., Wan, Y., Xia, C., He, L., & Philip, S. Y. (2021). HETFORMER: Heterogeneous transformer with sparse attention for long-text extractive summarization. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 146–154).

Mihalcea, R. (2005). Language independent extractive summarization. In *ACL, Vol. 5* (pp. 49–52).

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 404–411).

Mohammad, S., Dorr, B., Egan, M., Hassan, A., Muthukrishnan, P., Qazvinian, V., et al. (2009). Using citations to generate surveys of scientific paradigms. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics* (pp. 584–592).

Nallapati, R., Zhai, F., & Zhou, B. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-first AAAI conference on artificial intelligence*.

Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023.

Nenkova, A., Vanderwende, L., & McKeown, K. (2006). A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 573–580).

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).

Qazvinian, V., Radev, D. R., Mohammad, S. M., Dorr, B., Zajic, D., Whidby, M., et al. (2013). Generating extractive summaries of scientific paradigms. *Journal of Artificial Intelligence Research*, *46*, 165–201.

Radev, D. (2004). MEAD-a platform for multidocument multilingual text summarization. In *Proceedings of the 4th international conference on language resources and evaluation, Lisbon, Portugal, 2004*.

Radev, D. R., Jing, H., Styś, M., & Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing & Management*, *40*(6), 919–938.

See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368.

Teufel, S., & Moens, M. (2002). Articles summarizing scientific articles: Experiments with relevance and rhetorical status. *Computational Linguistics*, *28*(4), 409–445.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Wang, D., Liu, P., Zheng, Y., Qiu, X., & Huang, X.-J. (2020). Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 6209–6219).

Wang, J., Zhang, C., Zhang, M., & Deng, S. (2018). Citationas: A tool of automatic survey generation based on citation content. *Journal of Data and Information Science*, *3*(2), 20–37.

Wei, Y. (2012). Document summarization method based on heterogeneous graph. In *2012 9th international conference on fuzzy systems and knowledge discovery* (pp. 1285–1289). IEEE.

Xiao, W., & Carenini, G. (2019). Extractive summarization of long documents by combining global and local context. 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)* (pp. 3011–3021).

Xu, J., Gan, Z., Cheng, Y., & Liu, J. (2020). Discourse-aware neural extractive text summarization. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 5021–5031).

Zerva, C., Nghiem, M.-Q., Nguyen, N. T., & Ananiadou, S. (2020). Cited text span identification for scientific summarisation using pre-trained encoders. *Scientometrics*, *125*(3), 3109–3137.

Zhang, X., Wei, F., & Zhou, M. (2019). HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 5059–5069).

Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X., & Huang, X.-J. (2020). Extractive summarization as text matching. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 6197–6208).

Zhong, M., Liu, P., Wang, D., Qiu, X., & Huang, X.-J. (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* (pp. 1049–1058).

Zhu, T., Hua, W., Qu, J., & Zhou, X. (2021). Summarizing long-form document with rich discourse information. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 2770–2779).