Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
COVIDSum: A linguistically enriched SciBERT-based summarization model for COVID-19 scientific papers

Xiaoyan Cai\textsuperscript{a,1,*}, Sen Liu\textsuperscript{a,1}, Libin Yang\textsuperscript{a}, Yan Lu\textsuperscript{b}, Jintao Zhao\textsuperscript{a}, Dinggang Shen\textsuperscript{c}, Tianming Liu\textsuperscript{d}

\textsuperscript{a} School of Automation, Northwestern Polytechnical University, Xi'an 710072, Shaanxi, People's Republic of China
\textsuperscript{b} Department of Cardiovascular Diseases, Xi'an Jiaotong University Hospital, Xi'an 710077, Shaanxi, People's Republic of China
\textsuperscript{c} School of Biomedical Engineering, ShanghaiTech University, Shanghai 201210, People's Republic of China
\textsuperscript{d} Cortical Architecture Imaging and Discovery Lab, Department of Computer Science and Bioimaging Research Center, University of Georgia, Athens, GA 30602, USA

ARTICLE INFO

Keywords:
COVID-19 scientific papers
Abstractive summarization
Linguistically enriched pre-trained language model
SciBERT

ABSTRACT

The coronavirus disease (COVID-19) has claimed the lives of over 350,000 people and infected more than 173 million people worldwide, it triggers researchers from diverse fields are accelerating their research to help diagnostics, therapies, and vaccines. Researchers also publish their recent research progress through scientific papers. However, manually writing the abstract of a paper is time-consuming, and it increases the writing burden of the researchers. Abstractive summarization technique which automatically provides researchers reliable draft abstracts, can alleviate this problem. In this work, we propose a linguistically enriched SciBERT-based summarization model for COVID-19 scientific papers, named COVIDSum. Specifically, we first extract salient sentences from source papers and construct word co-occurrence graphs. Then, we adopt a SciBERT-based sequence encoder and a Graph Attention Networks-based graph encoder to encode sentences and word co-occurrence graphs, respectively. Finally, we fuse the above two encodings and generate an abstractive summary of each scientific paper. When evaluated on the publicly available COVID-19 open research dataset, the performance of our proposed model achieves significant improvement compared with other document summarization models.

1. Introduction

The SARS-CoV-2 virus is having a devastating impact as coronavirus disease 2019 (COVID-19) continues to spread in communities around the world. Researchers from diverse fields are accelerating their research to help diagnostics, therapies and vaccines. Researchers also publish their recent research progress through scientific papers, since scientific publications enable results and ideas to be transmitted throughout the scientific community. However, manually generating the abstract of a scientific paper is time-consuming, increasing writing burden of the researchers and slowing down writing speed and publication time of the scientific paper. Abstractive summarization technique which automatically provides researchers reliable draft abstracts, can alleviate this problem. In this work, we propose a linguistically enriched SciBERT-based summarization model for COVID-19 scientific papers, named COVIDSum. Specifically, we first extract salient sentences from source papers and construct word co-occurrence graphs. Then, we adopt a SciBERT-based sequence encoder and a Graph Attention Networks-based graph encoder to encode sentences and word co-occurrence graphs, respectively. Finally, we fuse the above two encodings and generate an abstractive summary of each scientific paper. When evaluated on the publicly available COVID-19 open research dataset, the performance of our proposed model achieves significant improvement compared with other document summarization models.

\* Corresponding author.
\* E-mail address: xiaoyanc@nwpu.edu.cn (X. Cai).
\* Equal Contribution.
sentences and word co-occurrence graphs, respectively. Finally, we fuse the above two encodings using highway networks [11], incorporating linguistic knowledge into the contextual embeddings of scientific papers, and generating an abstractive summary for each scientific paper. The main contributions of this paper are thus threefold:

1. Heuristic sentence extraction methods based on prior knowledge are developed, and word co-occurrence graphs are utilized as linguistic features of sentences.
2. A novel linguistically enhanced SciBERT-based summarization model is proposed, which utilizes pre-trained language model, graph neural networks and highway networks to incorporate linguistic knowledge into the contextual embeddings of scientific papers.
3. Thorough experimental studies are designed and conducted to verify the effectiveness of the proposed model.

We organize the remaining part of this paper as follows. Section 2 reviews related works. Section 3 introduces our proposed linguistically enriched SciBERT-based summarization model. Section 4 and Section 5 present the experimental settings and the evaluation results, respectively. Conclusions are presented in Section 6.

2. Related works

2.1. Scientific paper summarization approaches

There are two major categories of approaches for scientific paper summarization: abstract generation-based approaches and citation-based approaches [12]. Abstract generation-based approaches aim to automatically generate an abstract of a research paper [13, 14]. Citation-based approaches involves generation of summaries based on a set of citing sentences in other scientific papers pointing to that paper [15–18]. We focus on abstract generation-based approaches in this study.

Previous research on abstract generation for scientific articles has focused almost exclusively on extractive methods, which aim to select sentences from the original text to construct a summary of the scientific paper. Contractor et al. [14] proposed to use Argumentative Zones for extractive summarization of scientific articles. Kinugawa and Tsuruoka [19] presented a hierarchical encoder-decoder extractive summarizer for academic papers. Collins et al. [15] released a benchmark dataset for summarization of computer science publications named CSPubMed, developed a supervised extractive summarization approach, and proposed a new metric named AbstractROUGE. Yang et al. [20] leveraged data-weighted reconstruction to amplify a scientific paper’s abstract. They conducted experiments on the real dataset (AAN2 and Microsoft datasets5) to confirm the effectiveness of their approach. With the release of the COVID-19 Open Research Dataset (CORD-19)6, researchers began to study automatic text summarization of COVID-19 medical research papers [21–24]. Park [23] proposed a Continual BERT for extractive summarization of COVID-19 literature. Su et al. [21] obtained a ranked list of relevant snippets from the COVID-19 literature given a query and then extracted the top-ranked relevant results to generate summaries.

Different from extractive summarization methods, abstractive summarization methods involve understanding of the content in the original documents, and they aim to create a new paragraph by using natural language generation to summarize the original document. Normally, abstractive summarization methods are more difficult and complex than extractive summarization methods, but they can produce a more flexible and concise summary. Alambo et al. [25] proposed to generate an abstractive summary of a scientific paper by developing salient language unit selection and text generation techniques. Recently, neural methods have led to encouraging results in abstractive summarization [1, 6, 26]. However, these methods focus on summarizing news articles which are relatively short. Researchers began to study neural abstractive summarization approaches for scientific papers. Nikolov et al. [27] is among the first to consider supervised generation of the abstract directly from the full body of the paper. They applied a convolutional encoder-decoder model [28] on PubMed open access subset7 to perform abstract generation task. Cohan et al. [29] proposed a discourse-aware attention model, which consists of new hierarchical encoder that models the discourse structure of a document, and an attentive discourse-aware decoder, to generate abstract summaries of scientific papers. Ju et al. [30] presented a modified unsupervised pipeline architecture, SciSummPip, that leverages a transformer-based language model for summarizing scientific papers. Tan et al. [22] adopted BERT [31] and GPT-2 [32] to generate abstractive summaries based on CORD-19 dataset. Esteva et al. [24] took BERT as the encoder and extended the original GPT-2 by adding a cross-attention function alongside every existing self-attention function as the decoder, to generate a single abstractive summary for CORD-19 document dataset.

2.2. Natural Language Generation (NLG) enhanced by graph structures

Many NLG tasks need to better understand global context under a particular generation process. For example, the summarization task requires structured representation to facilitate the connection of relevant entities, and the preservation of global context (e.g., entity interactions) [33] [34]. In order to help NLG, graph-to-sequence (Graph2Seq) models encode the full structural information contained in the graph via a neural encoder-decoder architecture [35]. Zhu et al. [36] extracted factual relations from the article to build a knowledge graph and applied graph attention networks (GAT) [10] to obtain the representation of each node. Then they proposed a Factual Corrector (FC) model to generate abstractive summaries with higher factual correctness. Huang et al. [33] proposed a knowledge graph-augmented abstractive summarization approach, which encodes each paragraph as a sub-KG using GAT and connects all sub-KGs with a Bi-LSTM. Jin et al. [37] proposed a novel model SemSUM, which leverages the information of original input texts and corresponding semantic dependency graphs to guide abstractive summarization process.

2.3. Pre-trained language models

Pre-trained language models (PTMs) [38–41] have achieved significant improvements for a wide range of natural language processing (NLP) tasks. Peters et al. [38] developed Embeddings from Language Models (ELMo), an approach to learn contextualized word representations by training a bidirectional LSTM to optimize a disjoint bidirectional language model objective. Radford et al. [39] proposed to improve language understanding by Generative Pre-Training (GPT), which uses a combination of unsupervised pre-training and supervised fine-tuning. Devlin et al. [31] proposed a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. BERT uses a masked language modeling objective to train a deep bidirectional Transformer encoder, which learns interactions between left and right context. Zhang et al. [40] incorporated knowledge graph into BERT to simultaneously learn lexical, syntactic and knowledge information. PEGASUS [42] is a task-specific PTM which is trained over massive pre-training corpora and via gap sentences generation task.

According to Sinha et al. [43], the language model pre-trained via

---

2 http://clair.eecs.berkeley.edu/aan/index.php/.
3 http://academic.research.microsoft.com/.
4 https://pages.semanticscholar.org/.
5 https://ncbi.nlm.nih.gov/pmc/tools/openftlist.
3. Linguistically enriched scibert-based summarization model for COVID-19 scientific papers (COVIDSum)

Fig. 1 illustrates the framework of our model, COVIDSum (COVID-19 scientific paper Summarization), which consists of four major modules: 1) Dataset Preprocessing, 2) Heuristic Sentence Extraction, 3) Word Co-occurrence Graph Construction, and 4) Linguistically Enriched Abstractive Summarization. The Data Preprocessing module retrieves abstract and textual content of each paper and removes papers which have missed abstracts or are not written in English language. Sentence Extraction module applies three heuristic methods to extract sentences of each paper. Word Co-occurrence Relationship Graph Construction module extracts word co-occurrence relationship to construct an un-weighted directed word co-occurrence graph. Linguistically Enriched Abstractive Summarization module proposes a hybrid summarization approach, which utilizes SciBERT and a GAT-based graph encoder to encode the word co-occurrence graphs respectively, adopts highway networks to fuse the above two encodings for obtaining context vectors of sentences, and applies Transformer decoder to generate summaries. In the following subsections, we will explain each module.

1. Data Preprocessing

| Feature                      | Description                                                                 |
|------------------------------|-----------------------------------------------------------------------------|
| cord_aid                     | link to the article in the CORD-19 corpus                                    |
| sha                          | sha of the paper                                                             |
| source_x                     | source of the paper                                                          |
| title                        | title of the paper                                                           |
| doi                          | doi of the paper                                                             |
| pmc_id                       | pmc id of the article                                                        |
| pubmed_id                    | pubmed id of the article                                                     |
| license                      | license of the paper                                                         |
| abstract                     | abstract of the paper                                                        |
| authors                      | authors of the paper                                                         |
| journal                      | journal of the paper                                                         |
| mag_id                       | mag id of the article                                                        |
| arxiv_id                     | arxiv id of the article                                                      |
| url                          | url of the paper                                                             |
| s2_id                        | s2 id of the paper                                                           |
| pdf_json_files               | json file of the pdf                                                         |
| pmc_json_files               | json file of the pmc                                                         |
| who_covidence_id             | who covidence id                                                             |
| publish_time                 | publish time of the paper                                                    |

2. Heuristic Sentence Extraction

Heuristic 1: Considering the salient information of a paper is mainly mentioned at the beginning of the paper, we sequentially select

3.2. Heuristic sentence extraction

Though the maximum length of input for existing pre-trained language models is 512 words, the average length of papers in CORD-19 corpus is 6972.7 words on average. Too long input sequence not only requires extremely high computational power of the computer, but also prolongs the inference phase. Thus, we propose a two-stage summarization method. In the first stage, we extract sentences from different parts of sections in a paper, which can limit the length of paper content to a reasonable length. Then, in the second stage, based on the selected paper content, we train a linguistically enriched pre-trained language model for abstractive summarization. In the first stage, we propose three heuristic sentence extraction methods:

Heuristic 1: Considering the salient information of a paper is mainly mentioned at the beginning of the paper, we sequentially select...
sentences from the beginning of a paper and integrate them into a single paragraph until the length of the paragraph reaches 512 words.

**Heuristic2:** Since core parts of a scientific paper are often addressed in the Introduction or the Conclusion section, we separately select sentences in the Introduction section and Conclusion section, until the sentence length reaches 300 words for the Introduction section, and 212 words for the Conclusion section.

**Heuristic3:** We found that the first and last sentence of a section are usually conclusive statements. Thus, we sequentially select the first and last sentence of each section to form a paragraph, until the paragraph length reaches 512 words.

We apply the above three heuristic methods to extract sentences from the processed papers and use the extracted sentences as body of the corresponding paper.

### 3.3. Word co-occurrence graph construction

Word co-occurrence relationship expresses the dependency between words since the words occurring in similar context have much closer semantic and syntactic similarities. Traditionally, word co-occurrence graph is an un-weighted directed graph, where word co-occurrence relationships are extracted with a fix-sized sliding window over a sequence of words. All the words in the sequence are deemed as vertices of the graph, and edges will be added between two vertices if the two vertices appear in a window at the same time. In this study, we define the edges of the graph as directed edges, and the direction of edges is consistent with the corresponding order of word vertices in the sequence, which enables information to propagate on the graph while remaining positional information of the original sequence.

Fig. 2 illustrates three wordpiece-level co-occurrence graphs based on the sentence, “Most VA care was provided in VA facilities before the pandemic.”, using the tokenizer of BERT, BioBERT, and SciBERT, respectively. For these words that can be tokenized into subwords, we extend the word-level co-occurrence to wordpiece-level by connecting the head wordpiece to each other wordpieces with edges. Identical words that appear more than once at different positions of a sentence are highlighted with bordered text boxes, such as va in subgraph (a) and (c), and v in subgraph (b). In our settings, identical words within a sentence share their neighbors to aggregate more contextual information. As shown in Fig. 2, we connect these identical words with their neighbors using dotted lines with arrows.

### 3.4. Linguistically enriched abstractive summarization approach

In this subsection, we first describe the encoding process of the source textual sequences and the inner mechanism of the BERT language model. Then we demonstrate using a graph attention to process and encode linguistic patterns, i.e., the word co-occurrence relationships. After that, we adopt highway networks to alleviate difficulties in back propagating gradients and to merge linguistic features with contextualized embeddings. Finally, we introduce Transformer decoder equipped with the copy mechanism and the learning objective for the abstractive summarization task. Fig. 3 illustrates the overall architecture of our proposed approach.

#### 3.4.1. Pre-trained SciBERT-based sequence encoder

Since BERT is a classic example of pre-trained language models, we chose to utilize the original BERT as the sequence encoder in our proposed approach at first. However, the vanilla BERT that was pre-trained on the general corpora may not achieve the state-of-the-art performance in a specific domain (e.g., legal documents, clinical reports, scientific literature). Domain-oriented variants of BERT are randomly initialized the parameters of BERT while remaining its architecture and then pre-trained on the domain-specific corpora. In such a manner, domain knowledge would be integrated into the domain-specific BERT models, which significantly improves performance on the downstream tasks. Since we focus on abstractive summarization of scientific literature in biomedical field, we adopt BioBERT [45] and SciBERT [9] as sequence

---

**Fig. 2.** Illustration of the construction of word co-occurrence graphs based on a sentence. The sliding window size is 3, and the direction of the edge follows the relative order of wordpieces in corresponding sentence.
BPE algorithm \[46\]. BioBERT takes original wordpiece vocabulary of sentence can be also converted into VA facilities before the pandemic. BERT, while SciBERT trains its exclusive vocabulary on its scientific provided encoders separately. In this way, we further investigate the effectiveness -abstractive summarization approach.

Fig. 3. An overall architecture of our proposed linguistically enriched abstractive summarization approach.

Transformer-based encoder naturally lacks the capability of capturing positional relations. Thus, BERT encoder should be provided with explicit position embeddings as inter-word position information and segment embeddings as inter-sentence position information.

We map elements of the input wordpiece sequence into the token embeddings $E_{seq} = \{e_{1}^{seq}, e_{2}^{seq}, \ldots, e_{M}^{seq}\}$, position embeddings $E_{pos} = \{e_{1}^{pos}, e_{2}^{pos}, \ldots, e_{M}^{pos}\}$ and segment embeddings $E_{seg} = \{e_{1}^{seg}, e_{2}^{seg}, \ldots, e_{M}^{seg}\}$, where $e_{i}^{seq}, e_{i}^{pos}, e_{i}^{seg} \in \mathbb{R}^{d_{model}}$ and $d_{model}$ refers to the hidden size in BERT model.

The overall word embedding for the $i$-th word can be presented as:

$$e_{i} = e_{i}^{seq} + e_{i}^{pos} + e_{i}^{seg}$$

(3)

The contextualized representation of each wordpiece $H_{seg}^{i} = \{h_{i}^{seg}\}_{i=1}^{M}$ can be obtained as:

$$H_{seg}^{i} = \text{BERT}(E)$$

(4)

where $E = \{e_{1}, e_{2}, \ldots, e_{M}\}$.

$H_{seg}^{i}$ is then input to the graph attention module as the source for knowledge encoder.

3.4.2. Graph attention networks based knowledge encoder

Graph representation learning aims to learn better vertex representations that are either suited for downstream tasks or have the consistency between vector space and semantic space. Various graph neural networks [10,47,48] have been proposed and achieved promising performance. Though Graph Convolution Networks (GCN) [47] is widely used, it is only compatible with transductive tasks and has limited capability of handling unknown vertex in inference process. On the contrary, GraphSAGE (Graph Sample and aggreGate) [48] and the Graph Attention Networks (GAT) [10] are proposed to perform inductive learning on graphs. Especially, GAT utilizes the masked self-attention mechanism to capture dependency within neighbors and prevent information flow between disconnected nodes.

Therefore, we adopt GAT as knowledge encoder in our proposed approach. Based on the constructed word co-occurrence graph, GAT propagates linguistic representations from vertex to vertex in the graphs, and updates vertex representations with the embeddings of associated neighbor vertices. Moreover, multi-head attention mechanism is utilized in the graph attention layer to enable the knowledge encoder mine diverse linguistic features from different aspects.

To be consistent with the outputs of wordpiece-level sequence encoder, we employ BPE algorithm to the word co-occurrence graph by extending the original vertex into multiple sub-word vertices. We add edges from the head wordpiece to other wordpiece, to guarantee the smooth channels for information propagation between sub-words. Thus, we can smoothly employ the outputs $H_{seg}^{i}$ from sequence encoder as the initial node embeddings of nodes in wordpiece level co-occurrence graphs.

Given the initial node representations $\{u_{1}, u_{2}, \ldots, u_{M}\}$ and the neighbors of $i$-th node $N(u_{i})$ which are inferred from the graph structure, a GAT layer aggregates information from the neighbors and updates the hidden state of the node $u_{i}$ with multi-head attention mechanism as:

$$\bar{u}_{i} = \left|_{i=1}^{M} \text{Att}(u_{i}, u_{i}) \right|$$

(5)

$$\text{Att}(u_{i}, u_{i}) = \sigma \left( \sum_{k \in \text{N}(u_{i})} \alpha_{k}^{i} W_{k} u_{i} \right)$$

(6)

$$\alpha_{k}^{i} = \frac{\exp(\text{LeakyReLU}(u_{i}^{T} W_{k} u_{i})))}{\sum_{k \in \text{N}(u_{i})} \exp(\text{LeakyReLU}(u_{i}^{T} W_{k} u_{i})))}$$

(7)

where $\left|\cdot\right|$ denotes the concatenation operation of two vectors, $K$ is the number of heads in multi-head attentions, $W_{k}$ and $u_{k}$ are learnable parameters.

The knowledge encoder takes the vertex representations $H_{seg}^{i}$ and the
graph structure $G$ (i.e., the adjacency matrix of the word co-occurrence graph) as input, and then outputs the encoded linguistic features, which can be formulated as:

$$H^{\text{enc}} = \text{GAT}(H^{\text{emb}}, G)$$

(8)

where $\text{GAT}()$ represents multiple layers of graph attention networks, i.e., the knowledge encoder, and $H^{\text{enc}}$ is the corresponding output node embeddings.

3.4.3. Highway network based feature fusion

With the increasing depth of deep neural networks, researchers found it difficult to train models, and the experimental performance on both training and test dataset are also degrading. To this end, highway networks [11] and residual connections [49] are proposed to regulate information flow and ease the gradient back propagation using skip connections between layers.

The highway networks with gating mechanism employ a Transform gate $T$ to filter the outputs from a non-linear transformation module, and a Carry gate $C$ to directly pass the inputs of the module after rescaling. Moreover, since the highway networks adopt two gating functions to scale and combine hidden states from two sources and generate one representation, which can be used as a module for fusing features. The gating function of highway networks is the sigmoid function as:

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

(9)

We define the non-linear Transform gate function $T(x) = \sigma(W_x x)$, the Carry gate function $C(x) = \sigma(W_c x)$, which indicates how much information of the output and input are reserved, respectively. $W_x$ and $W_c$ are parameters to be trained. We deem the contextualized wordpiece embeddings from sequence encoder $H^{\text{emb}}$ as the input of the non-linear transformation, the encoded linguistic knowledge $H^{\text{emb}}$ as the output of the transformation. Then we apply the highway networks in our approach, and the Highway Connection layer in our approach can be presented as:

$$H = T(H^{\text{emb}}) \odot H^{\text{emb}} + C(H^{\text{emb}}) \odot H^{\text{emb}}$$

(10)

where $\odot$ denotes the element-wise multiplication, and $H = \{h_1, h_2, \ldots, h_d\}$ is representation sequences of the fused features.

To comprehensively study the effectiveness of skip-connections between layers, we also utilize several variants of the original highway networks:

- **Highway C** indicates the removal of Transform gate as:
  $$H^C = H^{\text{emb}} + C(H^{\text{emb}}) \odot H^{\text{emb}}$$
  (11)

- **Highway T** means removing the Carry gate:
  $$H^T = T(H^{\text{emb}}) \odot H^{\text{emb}} + H^{\text{emb}}$$
  (12)

- **Highway Coupled** associates the two gate functions as:
  $$H^{\text{coupled}} = T(H^{\text{emb}}) \odot H^{\text{emb}} + (1 - T(H^{\text{emb}})) \odot H^{\text{emb}}$$
  (13)

- **Residual Connection** is the degraded version of highway networks which only uses element-wise sum without gating mechanism:
  $$H^{\text{res}} = H^{\text{emb}} + H^{\text{emb}}$$
  (14)

The outputs of the feature fusing are deemed as the linguistic knowledge-aware contextualized representations of wordpieces in the source sequence and are fed into the Transformer decoder for the summary generation.

3.4.4. Transformer decoder based summary generation

We adopt vanilla Transformer as the decoder of our summarization model. The Transformer decoder has multi-head attention layers, a point-wise feed-forward layer as well as residual connection, and layer-normalization layers. The multi-head attention mechanism with the query $Q$, key $K$ and value $V$ is defined as:

$$\text{MHAtt}(Q, K, V) = \sum_{i=1}^{K} \frac{K_{ij}}{\sqrt{d_{	ext{model}}}}$$

(15)

$$\text{Attention}(w_{ij}, w_{ik}, Q, K, V) = W_v(W_e(W_q Q) + W_k W_k K + W_v V)$$

(16)

$$A = \text{softmax}(\frac{(W_q Q) T (W_k K)}{\sqrt{d_{\text{model}}}})$$

(17)

where $W_q, W_k, W_v, W_e$ are learnable parameters, $K$ is the number of heads, $k$ indicates the head number ids and matrix $A$ is the attention matrix. Given the partially generated summary $y = \{y_1, y_2, \ldots, y_i\}$ and the source sequence representation $H = \{h_1, h_2, \ldots, h_u\}$, the decoder predicts the next token $y_{i+1}$ using the following procedures:

First, the initial embeddings of the generated summary sequence $Y = \{y_1, y_2, \ldots, y_i\}$ is obtained by summing the token embedding, positional embedding and segment embedding of each token $y_i$ as:

$$y_i = y^{\text{emb}}_i + y^{\text{pos}}_i + y^{\text{seg}}_i$$

(18)

Then, followed by a layer-normalization layer, the masked multi-head self-attention is adopted to get the contextual representations of elements in the generated sequence as:

$$\tilde{Y} = \text{LayerNorm}(\text{MHAtt}(Y, Y, Y) + Y)$$

(19)

where LayerNorm stands for the layer normalization function.

We use a lower triangular matrix to mask the future information, preventing the attention to unnoticeable tokens in inference, and the attention weights after masking can be represented as:

$$a_{ij} = \begin{cases} 0, & i > j \\ a_{ii}, & i \leq j \end{cases}$$

(20)

where $a_{ij}$ is an element of the attention matrix $A$ in the calculation for masked self-attention.

Finally, the decoder calculates the external attention mechanism using the encoded representation $H$ and masked self-attention output $\tilde{Y}$. The external attention mechanism and the feedforward transformation are represented as follows:

$$\hat{Y} = \text{LayerNorm}(\text{MTAtt}(\tilde{Y}, H, H) + \hat{Y})$$

(21)

$$\hat{Y} = \text{LayerNorm}(\text{FFN}(Y) + Y)$$

(22)

where $Y = \{y_1\}^{i-1}$ and $\hat{Y} = \{y_1\}^{i+1}$ represent the outputs of external attention module and feedforward transformation. FFN is the position-wise feedforward network that contains two layers of linear transformation and a rectified linear unit (ReLU) as hidden activation function.

The probability distribution $p$ over the pre-defined wordpiece vocabulary can be formulated as:

$$p = \text{softmax}(W_x(W_a y_i + b_v) + b_h)$$

(23)

where $W_x, W_a, b_v, b_h$ are parameters to be learned.

In the decoding process, we obtain the output words auto-regressively by sampling from the distribution $p$.

3.4.5. Learning objective

Our proposed COVIDSum model is optimized with the negative log-likelihood loss:

$$L = -\frac{1}{T} \sum_{t=1}^{T} \log p_{w_t}$$

(24)

where $T$ is the length of the target wordpiece sequence and $w_t$ is the
ground-truth word which should be predicted at the t-th timestep.

4. Experiments

4.1. Datasets

Besides CORD-19 summarization dataset, we also conduct statistical analysis of different summarization datasets, including CNN/Daily Mail (CNN/DM) [6], New York Times (NY Times) [8], PubMed [50] and arXiv [51]. CNN/DM and NY Times are widely used benchmark datasets in news domain, and PubMed and arXiv are datasets for scientific literature summarization. The statistics results of each dataset is presented in Table 1.

From Table 1, we can find that CORD-19 dataset has the highest compression ratio, which indicates the difference between the length of source documents and corresponding summary is greater. In other word, the summarization task is more challenging on CORD-19 than other benchmark datasets. We randomly divide the pre-processed CORD-19 dataset into training set, validation set and test set. The number of training, validation and testing documents are 114,415/6,477/6,356. We train our proposed model and all the other comparing models on the training set and conduct model selection from saved checkpoints by evaluating them on the validation set. Once we obtained the best models, we run them on the test set to obtain their final performances and present their experimental results in Section 5.

The data leakage is inevitable if the COVIDSum is trained on CORD-19 and evaluated on other scientific datasets. Because the papers in other scientific paper datasets might also appear in the CORD-19 summarization dataset. Thus, we only test our COVIDSum pipeline on the CORD-19 summarization dataset.

The dataset and source codes of this research are available from the corresponding author upon reasonable request.

4.2. Comparison methods

We utilize variants of BERT as sequence encoder in our proposed architecture, including original BERT and two domain-specific variants on BERT, i.e., BioBERT, SciBERT. We employ vanilla Transformer decoder to learn to generate summaries. Moreover, to constrain the literature summarization. The statistics results of each dataset is presented in Table 1.

| Dataset   | #Doc | Compression ratio | Target | Source |
|-----------|------|-------------------|--------|--------|
|           |      |                   | # Avg. words | # Avg. sents | # Avg. words |
| CNN/DM    | 312 K| 13.54             | 55.4    | 3.9     | 750.2      |
| NY Times  | 655 K| 24.67             | 26.7    | 1.5     | 688.6      |
| PubMed    | 131 K| 15.04             | 214.4   | 6.9     | 3224.4     |
| arXiv     | 215 K| 23.61             | 292.8   | 9.6     | 6913.8     |
| CORD-19   | 127 K| **30.11**         | 231.8   | 22.8    | 6979.5     |

| Hyper-parameter | Value |
|-----------------|-------|
| type_bert       | SciBERT       |
| type_fuse       | Highway       |
| type_sent_ext   | Heuristic1    |
| num_gat         | 3           |
| lr_pt            | 2 × 10⁻³     |
| lr_ri           | 10⁻¹        |
| warmup_pt       | 20,000      |
| warmup_ri       | 10,000      |

HIBERT: A two-stage extractive summarization model [55], which pre-trains a hierarchical Transformer via the sequence labeling task and then applies the pre-trained encoder to the downstream extractive summarization task.

PEGASUS-Large [42]: A large-scale pre-trained language model, which is pre-trained via Gap Sentence Generation objective and achieves State-of-the-Art in several summarization tasks.

Our proposed heuristics are not applied to comparison methods, because it has been claimed that all input documents are truncated to 1024 words for PEGASUS-Large, 512 words for BERTSumAbs and HIBERT, and 400 words for Pointer-Generator Networks. LEAD-3 and TextRank are agnostic to the length of input documents.

4.3. Evaluation metric

ROUGE (Recall Oriented Understudy of Gisting Evaluation) metric [56] is used to evaluate the summarization model. ROUGE is recall-based evaluation method for text generation task, which computes n-gram based recall for the candidate summary with respect to the references. We report F1 scores of ROUGE-1, ROUGE-2 and ROUGE-L, which refer to the overlap of word-level uni-gram, bi-gram and longest common sequence between the predicted and reference summary, respectively.

4.4. Parameter setting

We employ a vanilla Transformer decoder with 12 attention heads and a multiple layers of graph neural networks as knowledge encoder. To be consistent with the setting of pre-trained language models, both the knowledge encoder and the decoder modules have 768-dimensional hidden states for all models based on our proposed framework in this experiment.

In the training process, we adopt two Adam optimizers with \( \beta_1 = 0.9, \beta_2 = 0.99, \varepsilon = 10^{-9} \) to optimize the parameters of pre-trained encoder and the other parameters. As presented in Table 2, two fairly different learning rates are used for model optimization. Because the pre-trained language model-based sequence encoder are already well-trained, whereas other components in our model are trained from scratch, the learning rate for the untrained parameters should be relatively higher to quickly reach a reasonable distribution. Following [57], we adopt the NORM learning rate scheduling strategy to enable the value of learning rate to linearly increase for specific steps and exponentially decrease.

We obtained the optimal hyper-parameters for comparison methods by tuning them on the CORD-19 validation set. The parameters of all comparison methods were tuned on validation split of the CORD-19 summarization dataset, as was COVIDSum.

After tuning the hyper-parameters by evaluating variations of our proposed method on the validation dataset, we obtained a group of best hyper-parameters for COVIDSum, as shown in Table 2 (type_bert and type_fuse are types of models used in our proposed architecture as sequence encoder and feature fusion module, respectively. type_sent_ext denotes the heuristic method of sentence extraction, num_gat represents the number of graph attention layers in the knowledge encoder. lr_pt
indicates the learning rate for the pre-trained encoder and $\text{lr}_\text{ri}$ refers to one for the randomly initialized modules in our proposed architecture. $\text{warmup}_\text{pt}$ and $\text{warmup}_\text{ri}$ are the numbers of warm-up steps for the pre-trained modules and untrained modules, respectively. In the following experiments for our proposed model, we only adjust corresponding modules in each subsection, while the others remain the tuned best settings in Table 2.

In the inference process, we obtain the predicted summaries by the beam search algorithm with a beam width of 5 and length penalty $\alpha = 0.6$. We train and evaluate each of the models in our experiments with 20 K steps on two piece of GeForce GTX 1080 Ti GPU.

5. Results and analysis

5.1. Impact of different heuristic extraction methods

The abstractive summarization module in our proposed model cannot handle the source texts without abridgment, because the original papers are almost 7000 words on average but the maximum sequence length for BERT is restricted to 512. Therefore, the heuristic sentence extraction methods are essential for our two-stage summarization model. In this set of experiments, we study the impact of different heuristic methods for sentence extraction. Statistics of the abstract-body pair datasets using our proposed sentence extraction methods are listed in Table 3. We find that all three proposed heuristic extraction methods can effectively reduce the length of the source papers, so that our pre-trained language model-based approach can handle the input sequences. Table 4 shows experimental results with three different heuristic extraction methods that have been explained in Section 3.2.

The results in Table 4 show that the abstractive summarization performance using Heuristic1 is rather competitive, which indicates the Introduction section contains enough salient information to generate a summary of the corresponding paper. ROUGE-2 values drop 1.34 and 4.40 points when the sentence extraction method is replaced with Heuristic2 or Heuristic3. From this observation, we infer that the essential information is distributed throughout sections in a paper, but relatively more concentrated in the Introduction, followed by the Conclusion section.

Thus, we can safely conclude that Heuristic1 is the best method to extract sentences. More specifically, we deem that most COVID-19 related scientific papers address the salient points in the Introduction sections, and the sentences at the beginning and the end of each section in a scientific paper are less informative for the composition of scientific abstracts.

5.2. Experiments using variants of BERT models

To further explore how the performance of our proposed models vary with different sequence encoders, we adopt original BERT, and its two domain-specific variations, i.e., SciBERT and BioBERT, as sequence encoder separately. In Table 5, we present the detailed information of the three pre-trained models, including their corpora for pre-training, sizes of vocabulary, and hyper-parameters for their structures.

We only change the type of sequence encoder and keep other settings in Table 2 in this experiment. The curves of training losses and perplexities are presented in Fig. 4, and the results on the test set are provided in Table 5.

From Fig. 4, we observe that cross-entropy loss of the three summarization models drop beneath 3.0 within 20,000 fine-tuning steps and continue to decline, the perplexity curves show the same trend. Perplexity is an evaluation metric for language models and reflects the uncertainty when a probabilistic model makes predictions. Among these curves, we find SciBERT-based model offers the worst training performance, and the BioBERT-based model surpasses the other two models in terms of convergence.

Table 6 shows the results of three variations of our proposed summarization model on the test set, from which we can make the following observations: 1) SciBERT-based model performs well on the test set, although the convergences of its training loss curve is the slowest among the three models, which indicates that the overfitting issue on the training set occurs when fine-tuning the pre-trained models; 2) when the BERT-based sequence encoder is replaced with domain-specific pre-trained language models, i.e., the SciBERT and BioBERT, our proposed abstractive summarization model achieves certain performance gains, which proves that in-domain pre-training corpora can help the downstream tasks. In our case, SciBERT and BioBERT improve the performance significantly as expected, since they are mainly pre-trained on the scientific papers from biomedical domain; 3) SciBERT-based model outperforms BioBERT-based model by 1.40 points on ROUGE-1, which emphasizes the necessity of a domain-specific vocabulary. Though both of two domain-specific pre-trained models are all obtained by training on the biomedical corpora and the corpus of BioBERT is much larger, the BioBERT-based model is less competitive because it simply inherits the vocabulary from BERT.

5.3. Effectiveness of variants of highway networks

The feature fusion module learns to merge the features generated by sequence encoder and knowledge encoder. And the quality of fused representations from the feature fusion module is crucial to the summary generation process. In this set of experiments, we analyze the effectiveness of various feature fusion methods in our proposed summarization framework.

As shown in Table 7, we find that the abstractive summarization model with highway networks achieves the highest ROUGE scores, and other variations of highway networks also show satisfactory results. We propose that two learnable gates in the highway networks enable the feature fusion module to learn control the information flows, which not only eases the back propagation of gradients, but also merges the output features from sequence encoder and knowledge encoder. We are surprised to find that the abstractive summarization framework with residual networks performs poorest. We attribute it to the fact that the highway networks are more suitable for abstractive summarization task, although the residual connections usually outperform highway networks in computer vision field.

5.4. Performance with different window size of word co-occurrence graph

A word in the sequence can connect more other words when increasing window size. To verify whether the word co-occurrence graph with more edges would lead to better abstractive summarization performance, we conduct experiments on word co-occurrence graphs with different window sizes. We construct the word co-occurrence graph using a strategy in which the same words in different positions of the sentence share their neighbors. Though this

| Dataset                  | Avg. # Sents per Doc | Avg. # Words per Sent |
|-------------------------|----------------------|-----------------------|
| Abstract                | 10.2                 | 22.7                  |
| Body (Heuristic 1)      | 18.8                 | 24.9                  |
| Body (Heuristic 2)      | 22.6                 | 24.4                  |
| Body (Heuristic 3)      | 21.1                 | 24.9                  |

Table 4: Our proposed model with different sentence extraction methods.

| Model      | ROUGE-1 | ROUGE-2 | ROUGE-L |
|------------|---------|---------|---------|
| Heuristic1 | 40.27   | 14.54   | 32.68   |
| Heuristic2 | 42.81   | 17.59   | 36.30   |
| Heuristic3 | 44.56   | 18.89   | 38.53   |
scheme allows diverse information to propagate to identical words and alleviate the polysemy problem to some extent, the number of edges surges as the window size increases because their neighbors are shared. To this end, we restrict the number of neighbors for each node in the word co-occurrence graph to five times of the corresponding window size at most. We set the window size of the word co-occurrence graph to 2, 3, 5, and 10, respectively, while settings of the other components remain default as represented in Table 2. We run our proposed abstractive framework with Heuristic1 sentence extraction method, SciBERT encoder, and highway networks as the feature fusion module on the CORD-19 test set. Table 8 shows experimental results.

As shown in Table 8, the word co-occurrence graph with a window size of 3 achieves the best performance of abstractive summarization. We observe that the ROUGE scores increase a little when the window size has not reached 3, but the ROUGE scores drop when the window size is above 3. We attribute the above results to the over-smoothing issue, that the node representations become indistinguishable when the graph neural networks go deeper, we deem that increasing number of connections in word co-occurrence graphs might hurt the summarization performance. Thus, we can conclude that the window size of the word co-occurrence graph indirectly affects the performance of our abstractive summarization framework. The underlying insight behind this observation is that the number of connections is associated with the over-smoothing issue for graph attention networks.

5.5. Comparison with other competing methods

After hyper-parameter tuning, we find that our model with Heuristic1 sentence extraction method, SciBERT as sequence encoder, highway networks as feature fusion module and word co-occurrence graph with a window size of 3 achieves the best performance of abstractive summarization. We observe that the ROUGE scores increase a little when the window size has not reached 3, but the ROUGE scores drop when the window size is above 3. We attribute the above results to the over-smoothing issue, that the node representations become indistinguishable when the graph neural networks go deeper, we deem that increasing number of connections in word co-occurrence graphs might hurt the summarization performance. Thus, we can conclude that the window size of the word co-occurrence graph indirectly affects the performance of our abstractive summarization framework. The underlying insight behind this observation is that the number of connections is associated with the over-smoothing issue for graph attention networks.

| Window Size | ROUGE-1 | ROUGE-2 | ROUGE-L |
|-------------|---------|---------|---------|
| 2           | 42.84   | 17.59   | 34.51   |
| 3           | 44.56   | 18.89   | 36.53   |
| 5           | 43.51   | 17.45   | 34.73   |
| 10          | 41.96   | 16.61   | 33.49   |
window size of 3, performs best. Thus, we compare our proposed model with other methods which have been introduced in Section 4.2. Table 9 shows the performance of different summarization methods on the CORD-19 test set.

The traditional extractive summarization methods, i.e., LEAD-3 and TextRank, deliver mediocre performance on all ROUGE scores. Since LEAD-3 only selects the first three sentences in source documents, the extracted summaries are too short compared to the ground-truth summary, which leads to high precisions, low recalls, and overall poor performance. The pointer-generator networks with coverage mechanism also shows unsatisfactory ROUGE scores because LSTM-based encoder-decoder frameworks are more suited for the summarization of short text. Three pre-trained language model-based summarization approaches (i.e., BERTSumAbs, HIBERT, PEGASUS-Large and COVIDSum) all achieve performance gains over the above three approaches. BERTSumAbs is the sequence-to-sequence baseline with BERT as encoder and Transformer as decoder, the prior knowledge injected during the pre-training process enables its performance improvements. HIBERT pre-trains a hierarchical encoder and applies it in an extractive summarization model. However, since training of HIBERT demands the manual labels of target sentences, the quality of labels limits its capability. By incorporating linguistic knowledge, the word co-occurrence relationships specifically, to the summarization model, our proposed COVIDSum achieves the highest ROUGE scores comparing with the other comparing summarization methods. PEGASUS-Large loses its competence in the task of summarizing scientific documents. The PEGASUS-Large is a general-domain pre-trained model rather than a domain-specific one. The COVIDSum model, enhanced with SciBERT, shows high suitability for scientific summarization in the COVID-19 domain. Also, our proposed SciBERT-based COVIDSum model shows performance advantages over the general-domain BART for scientific summarization in the COVID-19 domain.

Furthermore, to examine the significance of improvement, we conduct statistical hypothesis testing. Table 10 presents the results of the two-tailed paired t-test (with \( p < 0.05 \)) comparing the COVIDSum with other abstractive summarization models including PGN + Cov, BERTSumABS, HIBERT, and PEGASUS-Large. The Null hypothesis (H_0) is that the difference of means on ROUGE-2 F1 measure of two methods equals zero. Alternative hypothesis (H_1) is that the difference of means on ROUGE-2 F1 measure of two methods do not equal zero.

As shown in Table 10, the improvements on ROUGE-2 F1 score of our COVIDSum are statistically significant, compared to other neural abstractive approaches.

### 5.6. Ablation study

Since it is difficult to determine whether each component in the COVIDSum contributes to the performance improvements, we compare our full model with three ablated variants. We conduct an ablation study by removing several modules while remaining the rest of the COVIDSum architecture unaltered. We report the following three typical ablation models:

- \( w/o \) (Graph Encoder & Feature Fusion): removing both the graph encoder and the feature fusion module, and the original COVIDSum degrades into a standard sequence-to-sequence model with a pre-trained encoder and a Transformer decoder; \( w/o \) Feature Fusion: removing feature fusion module and simply adding the features from two encoders together; \( w/o \) Pre-training: using a vanilla Transformer encoder without pre-training instead of the variant of a pre-trained BERT.

From Table 11, we observe that the overall performance on ROUGE metrics of COVIDSum model is rather comparative, but the ROUGE-2 score drops significantly when the pre-trained sequence encoder is substituted with a non-pre-trained one. This observation suggests that the pre-training process enables the Transformer encoder to capture the semantic features of input sequences, and further boosts the performance of COVIDSum. Compared to COVIDSum model, performance of the model \( w/o \) (Graph Encoder & Feature Fusion) declines dramatically. We deem that word co-occurrences are essential for summarization, and explicitly providing the word co-occurrence features contributes to the performance improvements of COVIDSum. Based on the results in the first, second and fourth rows, we can infer that both the graph encoder which incorporates word co-occurrence features, and the highway networks which fuses features can benefit the COVIDSum model and their contributions can be accumulated. Experimental results indicate that all features, techniques, and modules are effective for COVIDSum to achieve performance gains.

### 5.7. Human expert evaluation

The ROUGE metric only measures n-gram overlapping between the generated summary and the ground-truth summary. However, merely evaluating our method with ROUGE is not sufficient to prove its capability. To overcome this limitation, we also perform human expert evaluation with different summarization methods on CORD-19 dataset.

We predefine four indicators to evaluate the quality of a generated summary: 1) Informativeness, which indicates how much the salient information of the source documents are remained; 2) Fluency, namely readability, means whether the generated text is grammatically correct and easy to understand; 3) Coherence, which evaluates the logicalness of paragraphs or sentences; 4) Redundancy, which measures the summary should contain few repeated information (higher score in the table indicates lower redundancy).

Both the generated summaries and corresponding reference texts are required in our human evaluation settings. Thus, we did not evaluate the ground-truth summaries here, because they have already been used as reference texts to evaluate the generated summaries. We randomly select 200 samples from the test set of CORD-19 dataset and compare the summaries generated by our proposed COVIDSum and the summaries generated by other abstractive summarization models. We invite eleven expert volunteers to participate in our human evaluation, including three physicians, a fever clinic doctor, and seven trained annotators to rate these samples on a scale of 1 (very bad) to 5 (very good) in terms of the four aspects. Annotators are blind to the

### Table 9

| Methods              | ROUGE-1 | ROUGE-2 | ROUGE-L |
|----------------------|---------|---------|---------|
| LEAD-3               | 31.67   | 10.96   | 27.63   |
| TextRank             | 32.80   | 11.60   | 27.64   |
| PGN + Cov            | 38.11   | 14.47   | 31.81   |
| BERTSumAbs           | 41.90   | 15.50   | 32.92   |
| HIBERT               | 44.18   | 18.79   | 35.65   |
| PEGASUS-Large        | 43.85   | 18.50   | 32.97   |
| BART                 | 44.29   | 18.74   | 36.04   |
| COVIDSum             | 44.56   | 18.89   | 36.53   |

### Table 10

| Methods P-Value | Methods P-Value |
|-----------------|-----------------|
| Our approach (COVIDSum) v.s. PGN + Cov | 0.00609 |
| Our approach (COVIDSum) v.s. BERTSumAbs | 0.02965 |
| Our approach (COVIDSum) v.s. PEGASUS-Large | 0.01459 |
| Our approach (COVIDSum) v.s. HIBERT | 0.01562 |

### Table 11

| Methods               | ROUGE-1 | ROUGE-2 | ROUGE-L |
|-----------------------|---------|---------|---------|
| \( w/o \) (Graph Encoder & Feature Fusion) | 41.90   | 15.50   | 33.08   |
| \( w/o \) Feature Fusion | 42.06   | 16.68   | 34.47   |
| \( w/o \) Pre-training | 39.42   | 15.14   | 32.92   |
| COVIDSum              | 44.56   | 18.89   | 36.53   |
correspondences of the model types to the generated summaries. The blinding was achieved by restoring the original orders of summaries generated by different models and shuffling them before human evaluators. After human evaluation, the results are reorganized to their original orders. The average results are listed in Table 12. For each indicator, the human evaluation results with symbol are significantly different from COVIDSum using two-tailed paired t-test with p < 0.05.

Considering if the annotation process is not even reliable, the annotation results cannot be correct, analysis of Inter-Annotator Agreement (IAA) should also be included as well as the mean of annotations. In Table 12, we also present the Fleiss’s kappas and Krippendorff’s alphas in the parentheses below the mean scores to verify the reliability of our human expert evaluation and the agreement between annotators.

To be specific, when calculating the Krippendorff’s alpha, we model the disagreement using the interval difference function as below:

$$\delta^2(s_i, s_j) = (s_i - s_j)^2$$

where $s_i, s_j$ are scores given by annotators.

Based on the Fleiss’s kappas and Krippendorff’s alphas presented in Table 12, we believe the annotation process is reliable, and substantial agreement is shared among annotators. Given the fact that our annotators are made up of physicians, clinicians, and trained volunteers whose understandings for biomedical academic papers are diverse, the results of IAA analysis are satisfying. Table 12 shows that our COVIDSum model outperforms the other three extractive models on all four aspects. As the pointer-generator networks with coverage mechanism is an RNN-based sequence-to-sequence summarization model, it lacks the ability to capture long-range dependencies, which leads to its inferior performance in the task of summarizing lengthy scientific papers. The mediocre human evaluation results of the other two BERT-based models also are consistent with the automatic ROUGE metrics represented in Section 5.5. We can attribute the outstanding results of our proposed model to the pre-trained sequence encoder’s impressive ability to model sequence, the explicit linguistic knowledge (i.e. word co-occurrence graphs), and the highway networks-based feature fusion module.

### 5.8. Qualitative analysis for hallucinations

We randomly sampled a paper, Modeling the evolution of COVID-19 via compartmental and particle-based approaches: Application to the Cyprus case, to present a case study for hallucinations in automatically generated abstracts. Table 13 shows the reference abstract and summaries generated by BERTSumAbs, HIBERT, and COVIDSum.

As shown in Table 13, though the abstract generated by BERTSumAbs is fluent, some hallucinations still occur. For example, the objective of this paper is to “provide predictions for some scenarios”, instead of “provide more precise definition” as stated in predicted summaries. The proposed model is derived for countries like Cyprus, not just “a given case of Cyprus”. Moreover, BERTSumAbs predicts summaries with repetitions, such as the word “deaths”.

The extractive summarization model, HIBERT, provides a most readable predicted abstract. However, the coherence between extracted sentences seems problematic despite the satisfying fluency. We observe that although the order of sentences in the predicted abstract is consistent with their order in the source paper, they are not coherent enough for comprehension.

As an abstractive summarization model, our proposed COVIDSum model inevitably generates a few repetitions. The two duplicate adverbs (i.e., “actually” and “reliably”) in a context can be considered as repetitions, but they do not create obstacles for readers to understanding the whole paragraph. More importantly, much fewer hallucinations occur in the abstract generated by our proposed COVIDSum, which implies our

| Ground-truth | BERTSumABS | HIBERT | COVIDSum |
|--------------|------------|--------|----------|
| The COVID pandemic has seen worldwide outbreaks, resulting in over five million validated cases of infection. This work aims to provide more precise definition for a number of scenarios using two models. First, we used a time dependent infection rate and detection rate as a function of the number of deaths, intubation, and deaths. Second, we derive a model that is suitable for a given case of Cyprus. We show that the model can be used to predict the spread of the disease in Cyprus, where data typically used for modeling the evolution of COVID is too small for a meaningful data-driven analysis. | The COVID pandemic has seen worldwide outbreaks, resulting in over five million validated cases of infection. This work aims to provide more precise definition for a number of scenarios using two models. First, we used a time dependent infection rate and detection rate as a function of the number of deaths, intubation, and deaths. Second, we derive a model that is suitable for a given case of Cyprus. We show that the model can be used to predict the spread of the disease in Cyprus, where data typically used for modeling the evolution of COVID is too small for a meaningful data-driven analysis. | The COVID pandemic is a new disease but there is not enough understanding on its future evolution. In this study, we use two different models to provide forecasts for the spread and evolution of the disease in Cyprus. Although two models are based on different approaches, they can both yield consistent predictions within their corresponding uncertainties. | The COVID pandemic is a new disease but there is not enough understanding on its future evolution. In this study, we use two different models to provide forecasts for the spread and evolution of the disease in Cyprus. Although two models are based on different approaches, they can both yield consistent predictions within their corresponding uncertainties. |

**Table 12**

| Models | Informativeness | Coherence | Redundancy | Fluency |
|--------|----------------|-----------|------------|---------|
| PGN + Cov | mean/var | 3.35(0.33) | 3.44(0.35) | 3.41(0.30) | 3.55(0.31) |
|         | kappa/alpha | 0.669/0.671 | 0.593/0.596 | 0.646/0.648 | 0.675/0.678 |
| BERTSumAbs | mean/var | 3.83(0.24) | 3.76(0.27) | 3.87(0.24) | 4.04(0.25) |
|         | kappa/alpha | 0.640/0.643 | 0.623/0.626 | 0.644/0.646 | 0.717/0.719 |
| HIBERT | mean/var | 3.84(0.27) | 4.04(0.32) | 3.96(0.23) | 4.19(0.26) |
|         | kappa/alpha | 0.661/0.664 | 0.594/0.597 | 0.651/0.654 | 0.703/0.703 |
| COVIDSum | mean/var | 3.95(0.21) | 4.07(0.25) | 4.15(0.20) | 4.28(0.23) |
|         | kappa/alpha | 0.669/0.671 | 0.602/0.605 | 0.653/0.656 | 0.689/0.692 |

**Table 13**

A qualitative analysis of hallucinations in summaries (contents written in **bold** implies hallucinations, and contents written in *italic* refer to repetition.)
proposed model can provide a reliable draft abstract based on the paper contents before researchers write the final abstract.

Both qualitative and quantitative evaluations support the conclusion that the COVIDSum has overall advantages compared to other models. Moreover, statistical hypothesis tests are conducted for both automatic evaluation and human evaluation, which ensures the improvements of COVIDSum are significant.

6. Conclusion

In this paper, we propose to generate scientific paper summaries related to COVID-19 via a linguistically enriched BioBERT-based summarization model. We first extract salient sentences from source papers and construct word co-occurrence graphs based on the selected sentences. Then we adopt BioBERT and a graph attention network (GAT) based graph encoder to encode the sentences and word co-occurrence graphs respectively, and generate a summary of each scientific paper by fusing the above two encodings using highway networks. Experimental results show that our proposed COVIDSum outperforms other summarization models on COVID-19 open research dataset. The proposed COVIDSum would help researchers in their investigation with COVID-19 by speeding up the research process, and it demonstrates the feasibility and promise of tailoring specific NLP techniques to the domain of COVID.

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.

Acknowledgements

This work was supported in part by the National Key Research and Development Project of China (No 2018YFB1402604), the National Natural Science Foundation of China (Nos. 61872296, 61772429, U20B2065) and MOE (Ministry of Education in China) Project of Humanities and Social Sciences (No. 18YJC870001).

References

[1] A.M. Rush, S. Chopra, J. Weston, A neural attention model for abstractive sentence summarization, arXiv preprint arXiv:1509.00685.
[2] S. Chopra, M. Auli, A.M. Rush, Abstractive sentence summarization with attentive recurrent neural networks, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 93-98.
[3] J. Tan, X. Wan, J. Xiao, Abstractive document summarization with a graph-based attentional neural model, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 1171–1181.
[4] Q. Zhou, N. Yang, F. Wei, M. Zhou, Selective encoding for abstractive sentence summarization, arXiv preprint arXiv:1704.07073.
[5] P. Li, W. Lam, L. Bing, Z. Wang, Deep recurrent generative decoder for abstractive text summarization, arXiv preprint arXiv:1708.06025.
[6] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al., Abstractive text summarization using sequence-to-sequence rnn and beyond, arXiv preprint arXiv:1602.06023.
[7] F. Dernoncourt, A. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.
[8] E. Sandhaus, The new york times annotated corpus, Linguistic Data Consortium, Philadelphia 6 (12) (2008) c00752.
Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 2227–2237.

[39] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training.

[40] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, Q. Liu, Ernie: Enhanced language representation with informative entities, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 1441–1451.

[41] L.L. Wang, K. Lo, Y. Chandrasekhar, R. Rea, J. Yang, D. Burdick, D. Eide, K. Funk, V. Katsis, R.M. Kimmy, et al., Cord-19: The covid-19 open research dataset, in: Proceedings of the 1st Workshop on NLP for COVID-19 at ACL, 2020, 2020.

[42] J. Zhang, Y. Zhao, M. Saleh, P. Liu, Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, in: International Conference on Machine Learning, 2020, pp. 11328–11339.

[43] K. Sinha, R. Jia, D. Hupkes, J. Pineau, A. Williams, D. Kiela, Masked language modeling and the distributional hypothesis: Order word matters pre-training for little, arXiv preprint arXiv:2104.06644.

[44] N. Kassner, H. Schütze, Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly, arXiv preprint arXiv:1911.03343.

[45] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C.H. So, J. Kang, Bobert: A pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (4) (2020) 1234–1240.

[46] R. Sennrich, B. Haddow, A. Birch, Neural machine translation of rare words with subword units, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 1715–1725.

[47] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, Adv. Neural Informat. Process. Syst. 29 (2016) 3844–3852.

[48] W.L. Hamilton, R. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 1025–1035.

[49] S. Jastrzbécki, D. Arpit, N. Ballas, V. Verma, T. Che, Y. Bengio, Residual connections encourage iterative inference, in: International Conference on Learning Representations, 2018.

[50] M. Falagas, E. Pitsouni, G. Malietzis, G. Pappas, Comparison of pubmed, scopus, web of science, and google scholar: strengths and weaknesses, FASEB J.: Off. Publ. Federation Am. Soc. Exp. Biol. 22 (2) (2007) 338–342.

[51] V. Lariviere, C.R. Sugimoto, B. Macaluso, S. Milojevic, B. Cronin, arxiv e-prints and the journal of record: An analysis of rules and relationships, J. Assoc. Inform. Sci. Technol. (Print) 65 (6) (2014) 1157–1169.

[52] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: Bringing order to the web., Tech. Rep., Stanford InfoLab (1999).

[53] R. Mihalcea, P. Tarau, Textrank: Bringing order into text, in: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 2004, pp. 404–411.

[54] Y. Liu, M. Lapata, 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), 2019, pp. 3730–3740.

[55] X. Zhang, F. Wei, M. Zhou, 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, 2019, pp. 5059–5069.

[56] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, Text Summarization Branches Out (2004) 74–81.

[57] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.