ORIGINAL ARTICLE

Automatic Text Summarization of COVID-19 Research Articles Using Recurrent Neural Networks and Coreference Resolution

Mahsa Afsharizadeh [1,*], Hossein Ebrahimpour-Komleh [1], Ayoub Bagheri [2]

[1] *Department of Computer Engineering, Faculty of Computer and Electrical Engineering, University of Kashan, Kashan, Iran*

[2] *Department of Methodology and Statistics, Faculty of Social Sciences, Utrecht University, Utrecht, the Netherlands*

*Corresponding Author: Mahsa Afsharizadeh
Email: ma.afsharizadeh@gmail.com

Received: 11 September 2020 / Accepted: 24 October 2020

Abstract

Purpose: Pandemic COVID-19 has created an emergency for the medical community. Researchers require extensive study of scientific literature in order to discover drugs and vaccines. In this situation where every minute is valuable to save the lives of hundreds of people, a quick understanding of scientific articles will help the medical community. Automatic text summarization makes this possible.

Materials and Methods: In this study, a recurrent neural network-based extractive summarization is proposed. The extractive method identifies the informative parts of the text. Recurrent neural network is very powerful for analyzing sequences such as text. The proposed method has three phases: sentence encoding, sentence ranking, and summary generation. To improve the performance of the summarization system, a coreference resolution procedure is used. Coreference resolution identifies the mentions in the text that refer to the same entity in the real world. This procedure helps to summarization process by discovering the central subject of the text.

Results: The proposed method is evaluated on the COVID-19 research articles extracted from the CORD-19 dataset. The results show that the combination of using recurrent neural network and coreference resolution embedding vectors improves the performance of the summarization system. The Proposed method by achieving the value of ROUGE1-recall 0.53 demonstrates the improvement of summarization performance by using coreference resolution embedding vectors in the RNN-based summarization system.

Conclusion: In this study, coreference information is stored in the form of coreference embedding vectors. Jointly use of recurrent neural network and coreference resolution results in an efficient summarization system.

Keywords: Extractive Summarization; Coreference Resolution; COVID-19; Recurrent Neural Network; Long Short Term Memory; Gated Recurrent Unit.

DOI: https://doi.org/10.18502/fbt.v7i4.5321

1. Introduction

In December 2019, an outbreak of severe acute respiratory syndrome coronavirus2 spread to Wuhan, Hubei Province, China and then throughout China and beyond around the world. In February 2020, the World Health Organization (WHO) named the disease as Coronavirus Disease 2019 (COVID-19) [1]. COVID-19 was declared a pandemic by the WHO after confirmed cases reached 200,000 and deaths in more than 160 countries exceeded 8,000 [2]. The capacity of hospitals was suddenly depleted in a short period of time due to the hospitalization of patients with COVID-19. This undesirable situation, which was suddenly imposed on countries, prompted researchers to start their research on the virus for finding ways of transmission, prevention methods, new drugs, and discovering vaccine. In a situation where every minute is valuable to save the lives of hundreds of people, studying and analyzing scientific papers is a time consuming process [3].

In the current situation, an automatic text summarization system helps the research and medical community by identifying and extracting useful information in the articles. Automatic text summarization is an automated way to generate a compressed version of text documents that helps users find important information in the original text in less time [4]. The generated summary should be short enough and contain important information in the text [5].

In the following, we will first review some of recent articles in the field of extractive summarization using deep learning methods, then we will describe the articles in the field of summarization of COVID-19 research articles.

1.1. Extractive Summarization Using Deep Learning Methods

1.1.1. AttSum [6]

In 2016, Cao *et al.* proposed a query-based extractive summarization model called AttSum [6]. AttSum architecture consists of three layers: CNN (Convolutional Neural Network) layer, pooling layer and ranking layer. The CNN layer uses CNNs to obtain the query and sentence representations. The pooling layer obtains document embedding from sentence embeddings using attention mechanism. Finally, the Ranking layer, ranks each sentence based on the similarity of its embedding with the document embedding.

1.1.2. MVCNN [7]

Zhang *et al.* in 2016 used an enhanced CNN called multiview convolutional neural network for multi-document extractive summarization [7]. Multiview learning is a paradigm designed for issues where data comes from multiple sources, also called multiple views. In the text summarization procedure, each human summarizer produces a summary from his or her own perspective. Multiview learning combines these multiple views to produce the final summary. The proposed MV-CNN architecture uses several CNNs, each of which is considered a distinct summarizer. Each CNN uses a different filter window size. So each CNN is a summarizer that, given its filter window size, looks at the data from a different view and predicts the saliency score accordingly. The predicted saliency scores of different CNNs are blended together using multiview learning to produce the final scores.

1.1.3. SummaRuNNer [8]

In 2017, Nallapati *et al.* proposed a recurrent neural network based sequence model for extractive summarization called SummaRuNNer [8]. The proposed model consists of a two-layer bidirectional GRU. The first layer calculates word hidden representations sequentially and the second layer calculates sentence hidden representations based on word representations of the previous layer. Each sentence is sequentially met in a logistic layer and a binary decision is made as to whether the sentence belongs to the summary.

1.1.4. BoWLer [9]

In 2018, Dhakras and Shrivastava introduced a model called BoWLer (Bag of Word embeddings LearnER) [9]. BoWLer uses a simple sentence encoder based on bag of words embedding. Each sentence representation is calculated based on the average word embedding of its constituent words. The document encoder uses a bidirectional GRU to obtain the document representation. The summary generator component uses an attention-based GRU and performs binary classification using a softmax layer.

1.1.5. Multi-Document Extractive Text Summarization via Deep Learning Approach [10]

In 2019, Rezaei *et al.* used two deep learning architectures for extractive summarization [10]. The first step is to perform feature extraction and generate a feature-sentence matrix for the text sentences. At this step, some of the most important features of sentences in the context of text summation are extracted, such as sentence position, sentence length, TF-IDF and title similarity. Next, this matrix is given as input to two types of neural networks: An Auto-Encoder neural network and a Deep Belief Network. These networks perform matrix enhancement. This matrix calculates the sentence scores, and the higher-score sentences, which are more important, are selected to be included in the summary.

1.2. Text Summarization of COVID-19 Research Articles

So far, few authors have suggested ways to automatically summarize COVID-19 articles.

1.2.1. DeepMINE [3]

Joshi *et al.* have proposed a system called deepMINE [3]. This system consists of two main parts, Mine Article and Article Summarization. In the first part, the user enters the desired keywords and the system returns the related articles and links by searching in the title of the articles provided by CORD-19. The second section performs a summarization for an input article by means of deep learning and Natural Language Processing (NLP).

1.2.2. CAiRE-COVID [11]

Su *et al.* have presented a model called CAiRE-COVID [11]. CAiRE-COVID has three main modules: information retrieval, question answering, and summarization. The information retrieval module receives a query from the user and retrieves *n* top most relevant paragraphs. The question answering module specifies a list of the most relevant sentences retrieved in the previous step as the answer. The question answering module is applied to each of the n paragraphs to select the relevant sentences from each paragraph as the answers to the query. Then, again, these *n* paragraphs are re-ranked according to their highlighted answers and the top *k* paragraphs are specified. The summarizer module receives these *k* paragraphs and generates an extractive summary and an abstractive summary based on them. The extractive summary is generated based on the cosine similarity of the sentences to the query and the abstractive one is generated using the UniLM [12] and BART models [13].

1.2.3. Automatic Text Summarization of COVID-19 Medical Research Articles using BERT and GPT-2 [14]

Tan *et al.* have used pre-trained BERT [15] and GPT-2 [16] to summarize articles related to Corona [14]. The model consists of two parts: Unsupervised extractive summarization and abstractive summarization. The first part is done with a pre-trained BERT model and the second with the GPT-2 model. In the first step, a pre-trained BERT model is used to convert the sentences into sentence embedding. A k-medoid clustering is then applied to the set of sentence embeddings to obtain a set of cluster centers. This set of sentences forms an extractive summary. Then, using POS-tagger, a number of keywords are extracted from the extractive summary. Keyword-reference summary pairs are given to the GPT-2 model, and after training, system summaries are generated.

1.2.4. Continual BERT: Continual Learning for Adaptive Extractive Summarization of COVID-19 Literature [17]

Park have proposed a novel BERT architecture for summarizing articles [17]. This model is continuously trained on new data. This feature is useful for COVID-19 articles that are published daily. Continual BERT uses two separate BERT models. The first BERT model is called Active Column (AC) due to active training. The second BERT model is called the Knowledge Base (KB) to preserve previously learned information. The train phase consists of two steps. In the first step, AC is trained on a new task and also uses previously learned tasks through communication with KB. In the second step, the knowledge gained by AC is given to KB. Table 1 shows the comparison between various extractive summarization techniques proposed by the scholars. Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is a standard evaluation method for text summarization [18].

In this study, we present a model for extracting important sentences from COVID-19 articles using Recurrent Neural Networks and coreference resolution procedure. The proposed model extracts important parts of the articles efficiently and quickly and leads to ease and quick understanding of COVID-19 research articles. The rest of the paper is organized as follows. Part II describes the proposed extractive summarization method using Recurrent Neural Networks (RNN) and coreference resolution. Experimental Results are presented and discussed in section III. Finally, the paper is concluded in section IV.

2. Materials and Methods

2.1. Dataset

The COVID-19 Open Research Dataset (CORD-19) is a resource of scientific papers related to COVID-19 [19]. This database is designed to facilitate the development of text mining and information retrieval systems in the field of COVID-19 related scientific articles. In March 2020, the Allen Institute for AI (AI2), in collaboration with the US White House and other institutes, released the first version of CORD-19. The purpose of CORD-19 is to connect the machine learning community with biomedical experts to discover information from scientific articles more quickly. Users of this dataset use a variety of artificial intelligence-based techniques to extract useful information. This dataset contains more than 59,000 scientific papers, including more than 47,000 complete papers on COVID-19 or related diseases [14].

2.2. Methods

2.2.1. Word Embedding

Each word must be converted to a proper numerical representation to be understood by the computers. Obtaining the appropriate representation for words plays a significant role in the final performance of the NLP task [20].

Word embedding is a way to convert words into dense real vectors [21]. A famous model offered to learn word representation is Global Vectors (GloVe) method [22]. The Glove method is an unsupervised method that uses ratios of word co-occurrence probabilities to obtain word vector representations. This study has used pre-trained word embedding model derived from Wikipedia 2014+Gigaword 5 data based on the Glove model [22].

2.2.2. Dataset Preparation

The proposed method considers extractive summarization as a sentence regression task. Regression is a kind of supervised learning methods. It can only work with labeled datasets. But ready-made labels are not available for the sentences in CORD-19 dataset. So in the dataset preparation section, we first give a score to each sentence as ground truth based on the abstract section of the article containing it using the official ROUGE evaluation tool [18]. Then we can train our model through conducting regression to the scores.

2.2.3. Summarization System

2.2.3.1. General Description

In this study we present an extractive summarization system using deep learning and coreference resolution procedure. Extractive summarization extracts the most important parts of the text. In the first stage, the source documents are sent to the system. Then some preprocessing steps is required such as sentence splitting, tokenization, stemming and stop words removal. To convert words to real valued vectors, we use the pre-trained word embedding model obtained from Wikipedia 2014 + Gigaword 5 data based on the GloVe model [22]. The text is sent to a coreference resolution

Table 1. Comparison matrix of extractive summarization techniques

Model	Sentence Representation	Sentence Selection	Dataset	Performance Evaluation		
				ROUGE-1	ROUGE-2	ROUGE-L
AttSum [6]	CNN	Attention mechanism and similarity score	DUC 2005, 2006, 2007	37.01 (2005) 40.90 (2006) 43.92 (2007)	6.99 (2005) 9.40 (2006) 11.55 (2007)	-
MVCNN [7]	Multiple CNNs	Fully connected layer and regression scores	DUC 2001, 2002, 2004, 2006, 2007	35.99 (2001) 36.71 (2002) 39.07 (2004) 38.65 (2006) 40.92 (2007)	7.91 (2001) 9.02 (2002) 10.06 (2004) 7.91 (2006) 9.11 (2007)	-
SummaRuNNer [8]	Bidirectional GRU	Fully connected layer and softmax layer	CNN/DailyMail	39.6	16.2	35.3
BoWLer [9]	Bag of Words embedding	Attention-based GRU and softmax layer	CNN/DailyMail	40.3	17.5	36.5
Multi-Document Extractive Text Summarization via Deep Learning Approach [10]	Statistical features	Feature-sentence matrix enhancement by AE and DBN and select top ranked sentences based on their scores	DUC 2007	39.1 (DBN) 39.8 (AE)	8.9 (DBN) 9.2 (AE)	-
deepMINE [3]	Deep learning and NLP techniques	Not explained	CORD-19	-	-	-
CAiRE-COVID [11]	ALBERT	Cosine similarity score	CORD-19	-	-	-
Automatic Text Summarization of COVID-19 Medical Research Articles using BERT and GPT-2 [14]	Pretrained BERT	k-medoid clustering	CORD-19	-	-	-
Continual BERT: Continual Learning for Adaptive Extractive Summarization of COVID-19 Literature [17]	Pretrained BERT	Transformer encoder	CORD-19, ScisummNet	35.2 (CORD-19) 33.0 (ScisummNet)	15.5 (CORD-19) 13.4 (ScisummNet)	33.8 (CORD-19) 31.6 (ScisummNet)

system to get coreference information. We propose a new way to use coreference information in the summarization procedure. For the first time, coreference information is generated in the form of coreference embedding vectors and provided to the summarizer. For this purpose we use NeuralCoref [23]. NeuralCoref obtains coreference chains using a neural network-based method. We identify the central subject of the text based on the coreference chains. Simultaneously word embedding vectors are sent to the recurrent layers.

2.3.2.2. System Architecture

The proposed extractive summarization system is considered as a sentence regression task. Sentence regression tries to model the relationship between sentence features (sentence representation vector) and a real valued target label. The proposed method has three phases, sentence encoding, sentence ranking, and summary generation. In the sentence encoding

A sentence representation vector is then generated for each sentence. We enrich this sentence representation with the obtained coreference resolution information. Also we perform the dropout operation as a regularization method to prevent overfitting. Then the generated sentence representations are sent to a fully connected layer to generate sentence significance scores. We rank the sentences based on their scores and select the sentences with a higher rank to be present in the summary. Figure 1 shows the general steps of the proposed method.

phase, we need to get an appropriate representation for each word. For this purpose, we use pre-trained word embedding model [22]. All sentences in an article form a candidate set as $S = \{s_1, s_2, \ldots, s_N\}$, where s_i is the i_{th} sentence in the article. N shows the number of distinct sentences. The goal is to create a summary set called S^* as $S^* = \{s_1^*, s_2^*, \ldots, s_L^*\}$, which $L << N$ and $S^* \subseteq S$.

Figure 2 demonstrates the proposed system architecture. Our architecture consists of three layers: embedding layer, recurrent layers, and output linear layer. Each sentence s_i is a sequence of words as $s_i = w_1 w_2 \ldots w_m$, where w_i shows the i_{th} word in the sentence.

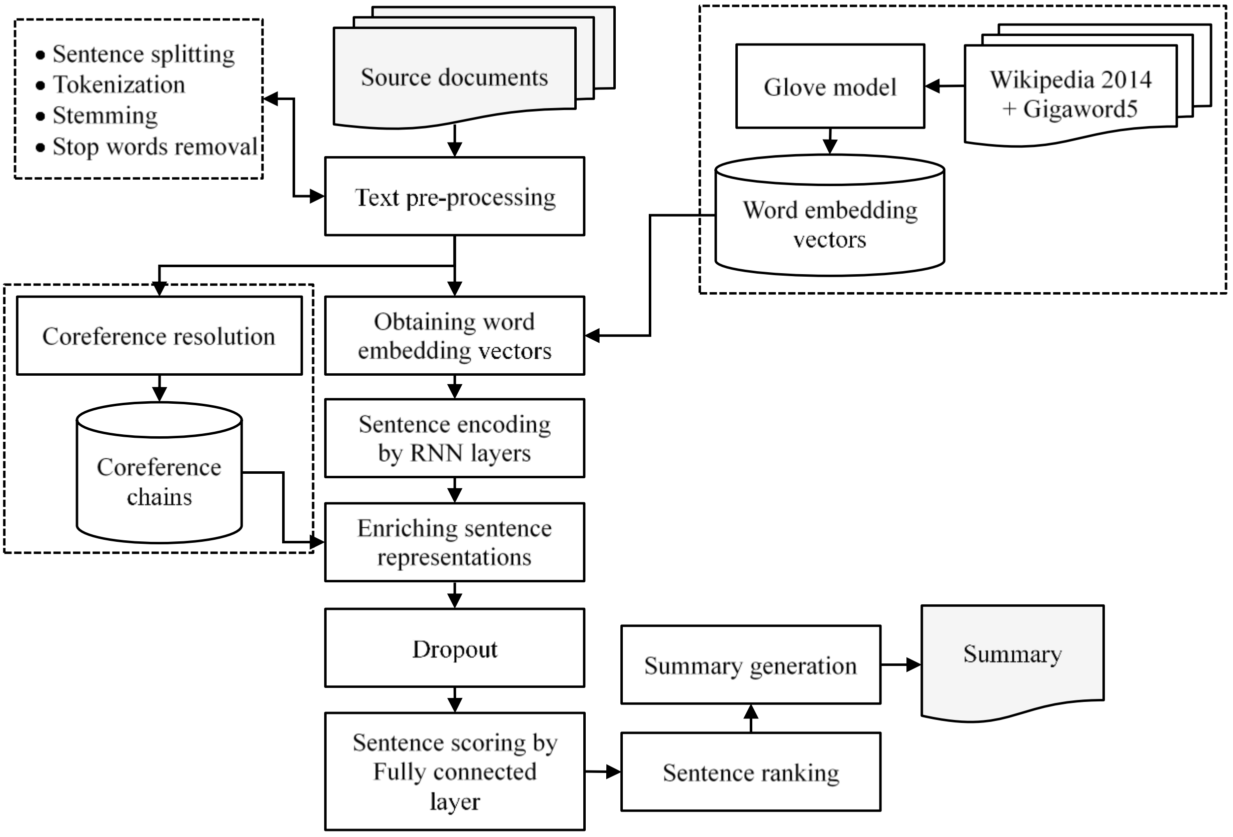

Figure 1. The general steps of the proposed method

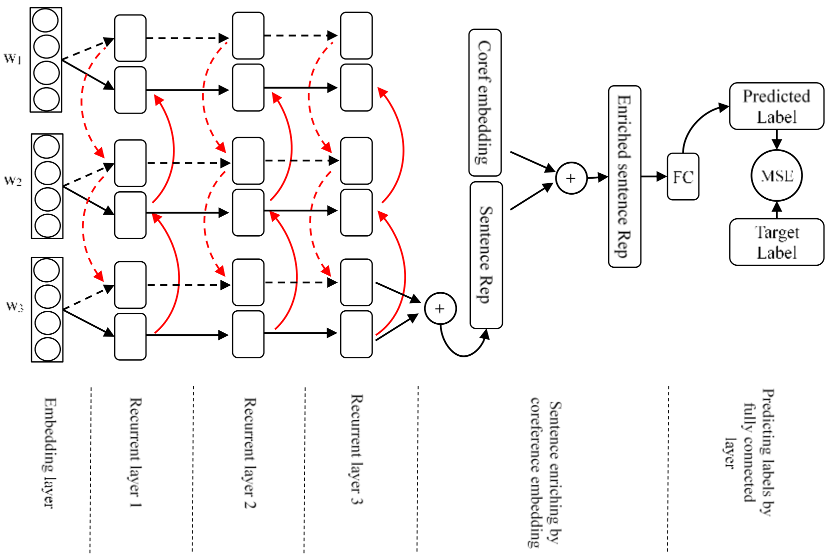

Figure 2. Neural architecture of the proposed method; w_i represents the embedding vector of the i-th word. Dashed arrows represent forward direction and normal arrows show backward direction. "+" means averaging operation. "FC" means Fully Connected layer

Embedding layer works in the word level. It maps each word w_i to the corresponding word embedding vector x_i. It forms a look-up table with the dimension *vocab_size* × *embedding_dim* that retrieves the embedding vectors for all the words in the vocabulary. *vocab_size* is equal to the count of unique words and *embedding_dim* is equal to the length of word embedding vectors.

Recurrent layers work in the sentence level. The output of the embedding layer is passed to the first recurrent layer. Recurrent part is used to generate sentence representation from the constituent words. It receives a sequence of word embedding vectors in the sentence $s_i = (x_1\ x_2\ \ldots\ x_m)$. In this study, we compare all three types of RNNs for extractive summarization, Vanilla RNN, RNN with GRU, and RNN with LSTM units.

If we use vanilla RNN, the hidden state value h_t in time step *t* is calculated by Equation (1). Where *g* is a non-linear function such as *sigmoid* or *hyperbolic tangent*. *W* and *U* are the input and hidden weights, respectively.

$$ht\ =\ g\,(W.xt\ +\ U.ht - 1) \tag{1}$$

Two other types of RNNs use a gating mechanism and have a more complicated recurrent hidden unit structure. We also examine the LSTM-based recurrent neural network [24]. Unlike an ordinary recurrent unit that only calculates a weighted sum of the input signal and applies a nonlinear function, each LSTM unit holds a memory cell at each time. When we use LSTM, the hidden state h_t is calculated by the Equation (2), where o_t is the output calculated by the Equation (3), C_t is memory cell updated via Equation (4), $\tilde{C}_t$ is new memory content obtained according to Equation (5), and f_t and i_t are forget gate and input gate calculated by the Equations (6) and (7), respectively.

$$ht\ =\ ot\ *\ tanh\,(Ct) \tag{2}$$

$$ot\ =\ \sigma\,(Wo.[ht - 1\,, xt])\ +\ bo \tag{3}$$

$$Ct\ =\ ft\ *\ Ct - 1\ +\ it\ *\ Ct \tag{4}$$

$$Ct\ =\ tanh\,(Wc.[ht - 1\,, xt]\ +\ bc) \tag{5}$$

$$it\ =\ \sigma\,(Wi.[ht - 1\,, xt]\ +\ bi) \tag{6}$$

$$ft\ =\ \sigma\,(Wf.[ht - 1\,, xt]\ +\ bf) \tag{7}$$

We also use a GRU-based recurrent neural network [25]. GRU consists of two gates, an update gate *z* and a reset gate *r*. GRU is described by the Equations (8) to (11).

$$zt\ =\ \sigma\,(Wz.[ht - 1\,, xt]) \tag{8}$$

$$rt\ =\ \sigma\,(Wr.[ht - 1\,, xt]) \tag{9}$$

$$\tilde{h}t\ =\ tanh\,(W.[rt\ *\ ht - 1\,, xt]) \tag{10}$$

$$ht\ =\ (1 - zt)\ *\ ht - 1\ +\ zt\ *\ \tilde{h}t \tag{11}$$

Activation h_t is a linear interpolation between the previous activation h_{t-1} and the candidate activation $\tilde{h}_t$ which is described by the Equation (11). The update gate z_t is calculated using the Equation (8). The candidate activation $\tilde{h}_t$ is calculated by the Equation (10), where r_t is the reset gate.

After the recurrent layers, the sentence representation vector h_m is obtained at the last time step. The sentence representation h_m is obtained by visiting the last word embedding vector x_m in the sentence. We can define this sentence representation vector as a function $f\ (s_i \mid ɵ)$ where ɵ represents the model parameters and s_i represents sentence *I* in the text.

3.3.2.2. Enriching Sentence Representation by Coreference Resolution

Coreference resolution is the process of identifying mentions in a text that refer to the same entity in the real world. The entity is a person, place, organization, object, etc. in the real world. A mention is a string in the text that refers to an entity. For example, "Barack Hussein Obama", "Senator Obama" and "President Obama" are mentions that refer to the entity of the 44th President of the United States [26, 27]. In this study, coreference resolution is also used in the summarization process. The integration of coreference resolution with extractive summarization can have a positive effect on extracting informative sentences. The coreference resolution procedure leads to the discovery of coreference chains. A coreference chain is a set of mentions in the text that all refer to the same entity in the real world. Coreference resolution helps to identify the central entity in the text. Sentences in which this entity is more prominent are then selected for the summary [28]. We used NeuralCoref to find coreference chains[1]. NeuralCoref annotates and resolves coreference chains using a neural network [23]. To use coreference information in the summarization procedure, we first identify the mentions in the sentences. Then we consider a coreference weight c_i to each sentence s_i, proportional to the importance of its entities. The importance of each entity is proportional to the length of its respective coreference chain. To consider coreference information in the summarization procedure, we generate coreference embedding vectors. The vector called emb_{ci}, which is a vector of the same length with the sentence representation vector, contains the binary representation of the embedding weight c_i. This is the first time that coreference information has been stored in the form of coreference embedding vectors. We use these vectors to enrich the representation of sentences. The sentence representation is updated via Equation (12):

$$hm\ =\ hm\ +\ embci \tag{12}$$

Enriched sentence representation vector is fed into the last layer. The last layer is a fully connected layer, which applies a linear transform to the sentence embedding vector. The significance of sentence s_i is calculated through a regression procedure by Equation (13):

$$\tilde{y}\ =\ W.f\ (si \mid ɵ)\ \ +\ b \tag{13}$$

All of the model parameters, including the weight matrixes and biases in the hidden layers (we called them as ɵ) and the weight matrix *W* and bias term *b* in the last layer are tuned during the learning process by minimizing the Mean Squared Error (MSE) of the predicted scores $\tilde{y}$ given the target values *y*.

We have previously produced a target score y_i for each sentence s_i in the dataset preparation phase by the Equation (14); where *abs* represents the abstract of the article containing the sentence s_i. ROUGE-1 and ROUGE-2 measure the unigram and bigram overlap between each sentence s_i and the article abstract *abs*, respectively [18].

$$yi\ =\ ROUGE - 1(si|abs)\ +\ ROUGE - 2(si|abs) \tag{14}$$

MSE loss function is computed by (15):

$$MSE\ (y, \tilde{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(ROUGE_2(s_i|abs) - (W.f\ (s_i|\theta) + b))^2 \tag{15}$$

Since the proposed method selects the sentences based on their similarity to the abstract of the article, so all the selected sentences are somehow semantically compatible with each other because in fact all of them are similar to the abstract of the article.

3. Results and Discussion

In order to demonstrate the summarization performance of our model, we compare it with Random, Lead, RNN, GRU and LSTM methods. For our experiments we use CORD-19 dataset. Because extractive summarization is implemented as a sentence regression operation, we use the MSE loss between target labels and predicted significance scores to evaluate the system.

[1] https://github.com/huggingface/neuralcoref

1.3. Parameter Settings

The embeddings of words used in our experiments are initialized using pre-trained word embedding model derived from Wikipedia 2014 + Gigaword5 data based on the Glove model [22]. This dataset contains six billion words. The dimension of each word embedding vector is 50. Our model is composed of three Bi-LSTM layers. The size of each hidden unit is 50. We set the batch size to 16. dropout is considered for regularization and the dropout rate is 0.15. To prevent gradient exploding, we use gradient clipping with a threshold value of 0.25 [29]. we use Adam optimizer to update model parameters with $lr = 0.001$, $\beta_1 = 0.9$ and $\beta_2 = 0.99$ [30]. We have implemented the proposed neural network-based model using PyTorch library. All the simulations are conducted using an NVIDIA GeForce GTX 1060 with 2.8 GHZ CPU and 16GB RAM. Each training epoch takes approximately 15 seconds.

2.3. Experimental Results

In this study, the performance of three types of RNNs is investigated and compared. These three types are Vanilla RNN, RNN with GRU, and RNN with LSTM units. Table 2 compares the performance of different types of RNNs.

Table 2. MSE loss comparison of Vanilla RNN, GRU, and LSTM. The lowest loss (highest efficiency) is for LSTM

Recurrent Neural Network	Vanilla RNN	GRU	LSTM
MSE Loss	0.0102	0.0084	**0.0056**

As shown in this Table, the LSTM network performs better than the other two RNNs in extractive summarization task. In theory, Vanilla RNNs are fully capable of addressing long-term dependencies, but in practice their training is difficult for long-term dependencies [31]. Therefore, GRU and LSTM are expected to be more successful than Vanilla RNN. But regarding the superiority of LSTM over GRU in our extractive summarization system, perhaps it can be said that LSTM is more accurate due to having more parameters and offers more suitable results than GRU.

The performance of the network with unidirectional and bidirectional layers is also compared. Figure 3 shows the superiority of bidirectional RNN. Bidirectional RNN produces better results for our model. Because it extracts more appropriate information from the sentences by considering both forward and backward directions in the sentences.

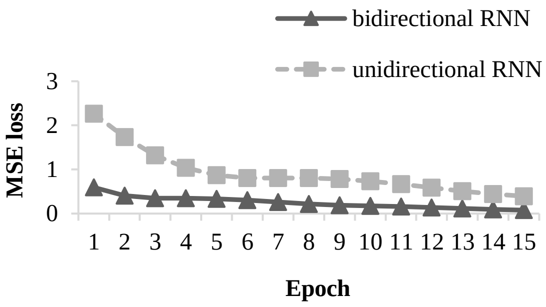

Figure 3. MSE loss comparison between unidirectional and bidirectional RNN. Bidirectional RNN outperforms unidirectional one

In this study the effect of the number of recurrent layers on the performance of the model is investigated. The architecture with one, two, and three layers is compared. The comparison result is shown in Figure 4. As we can see in this Figure, the network with more layers has better performance. The increase in the number of recurrent layers more than three has not been investigated due to the increased complexity. That's why a three-layer network is a good choice for our summarization system.

In this study, we have tested the performance of the system with and without using coreference resolution information. Figure 5 demonstrates that the performance of the summarizer by considering coreference embedding vectors is better.

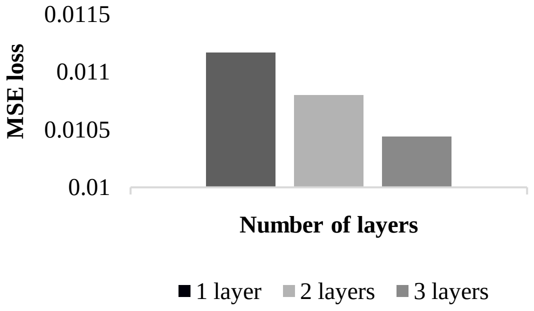

Figure 4. Loss comparison results between different number of recurrent layers. More layers result in higher performance (less MSE loss)

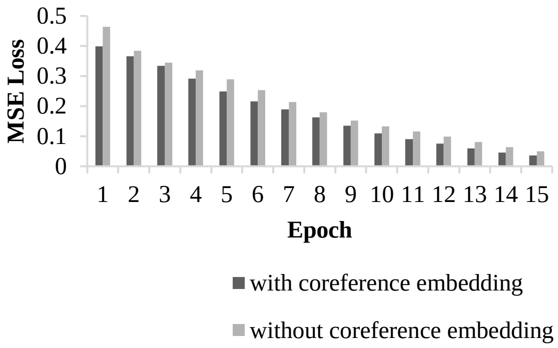

Figure 5. The effect of using coreference information on summarization performance. The use of coreference embedding vectors improves the efficiency of the summarization system during all epochs

This impressive result was expected. Because considering valuable information such as coreference resolution that identifies the central entity of the text and encoding this information in coreference embedding vectors, will certainly have a positive effect on identifying valuable sentences in the text.

We have compared our proposed method with the methods Random, Lead, RNN, GRU, LSTM, and Bi-LSTM. The results of this comparison are shown in Table 3. This Table suggests that encoding text coreference information as coreference embedding vectors and enriching the representation of sentences using them can improve the performance of the RNN-based summarization system.

Table 4 shows an example of an extractive summary produced for an article entitled "*COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images*"[2] and its gold summary. Using this brief summary with the abstract of the article provides a faster and more complete understanding of the article.

4. Conclusion

In this study, we propose an interpretable deep learning based model for extractive summarization of COVID-19 research articles. This task is considered as a sentence regression process. We combine RNN and coreference resolution procedure to better discover the most informative sentences.

We have evaluated the performance of different types of RNNs, namely vanilla RNN, RNN with LSTM units and RNN with GRU units in the field of extractive summarization. Among them, LSTM achieved the best performance. We have shown that bidirectional RNNs select better sentences to include in the summary by extracting more appropriate information from the sentences by considering both forward and backward directions in the sentences.

We also have examined the effect of increasing the number of layers on the performance of the summarization system. As the number of layers increased, more hidden information was extracted from the sentences. However, the excessive increase in the number of layers was not desirable due to the increased computational complexity and time spent.

Table 3. ROUGE Comparison results between the proposed method and some other RNN-based techniques

	ROUGE-1			**ROUGE-2**			**ROUGE-L**			**ROUGE-SU4**		
	Re	**Pr**	**F1**	**Re**	**Pr**	**F1**	**Re**	**Pr**	**F1**	**Re**	**Pr**	**F1**
Random	0.44361	0.22957	0.30256	0.13178	0.06911	0.09067	0.18391	0.11268	0.13974	0.19840	0.10500	0.13732
Lead	0.47368	0.23864	0.31738	0.10853	0.05512	0.07311	0.20690	0.12329	0.15451	0.18880	0.09672	0.12791
Vanilla RNN	0.47368	0.24803	0.32558	0.13178	0.06967	0.09115	0.26437	0.16667	0.20444	0.20800	0.11111	0.14485
GRU	0.48872	0.24621	0.32746	0.14729	0.07510	0.09948	0.20690	0.12500	0.15584	0.22400	0.11570	0.15259
LSTM	0.50376	0.26693	0.34896	0.13178	0.07054	0.09189	0.21839	0.13571	0.16740	0.19360	0.10476	0.13596
Bi-LSTM	0.51880	0.27711	0.36126	0.13953	0.07531	0.09783	0.21839	0.12500	0.15900	0.20640	0.11266	0.14576
Proposed Method	0.53383	0.25357	0.34383	0.17829	0.08614	0.11616	0.25287	0.14966	0.18803	0.23200	0.11364	0.15255

[2] https://arxiv.org/abs/2003.09871

Table 4. Comparison of system generated summary and gold summary for paper [32]

System summary	The main screening method used for detecting COVID-19 cases is reverse transcriptase-polymerase chain reaction (RT-PCR) testing, which can detect SARS-CoV-2 RNA from respiratory specimens (collected through a variety of means such as nasopharyngeal or oropharyngeal swabs). However, to the best of the authors' knowledge at the time of the initial release of the proposed COVID-Net, most of the developed AI systems proposed in research literature have been closed source and unavailable to the research community to build upon for deeper understanding and extension of these systems. Furthermore, most of these systems are unavailable for public access and use. In this study, we leverage generative synthesis as the machine-driven design exploration strategy, which is based on an intricate interplay between a generator-inquisitor pair that work in tandem to garner insights and learn to generate deep neural network architectures that best satisfies human specified design requirements. GSInquire revolves around the notion of an inquisitor I within a generator-inquisitor pair {G, I}, with G denoting a generator, that work in tandem to obtain improved insights about deep neural networks as well as learn to generate networks. In this study, the produced interpretation indicates the critical factors leveraged by COVID-Net in making a detection decision based on a CXR image, and can be visualized spatially relative to the CXR image for greater insights into whether COVID-Net is making the right decisions for the right reasons and validate its performance. The critical factors identified by GSInquire in several example CXR images of COVID-19 cases are shown in Fig. 7.
Gold summary	Motivated by the urgent need to develop solutions to aid in the fight against the COVID-19 pandemic and inspired by the open source and open access efforts by the research community, this study introduces COVID-Net, a deep convolutional neural network design tailored for the detection of COVID-19 cases from CXR images that is open source and available to the general public. We also introduce COVIDx, an open access benchmark dataset that we generated comprising of 13,975 CXR images across 13,870 patient cases, created as a combination and modification of five open access data repositories containing chest radiography images, two of which we introduced. Motivated by the need for faster interpretation of radiography images, a number of artificial intelligence (AI) systems based on deep learning have been proposed and results have shown to be quite promising in terms of accuracy in detecting patients infected with COVID-19 via radiography imaging, with the focus primarily on CT imaging. By no means a production-ready solution, the hope is that the promising results achieved by COVID-Net on the COVIDx test dataset, along with the fact that it is available in open source format alongside the description on constructing the open source dataset, will lead it to be leveraged and build upon by both researchers and citizen data scientists alike to accelerate the development of highly accurate yet practical deep learning solutions for detecting COVID-19 cases from CXR images and accelerate treatment of those who need it the most.

The network with more layers showed better performance. The coreference resolution procedure is also used to better identify the main subject of the text. This operation selects sentences that are more relevant to the central subject of the text and produces a more informative summary.

This article proposed a new way to use text coreference resolution information in the summarizing process. For the first time, coreference information was stored in the form of coreference embedding vectors, and sentence representation vectors were enriched by them. By integrating coreference embedding vectors with the RNN-based summarization system, the results of the summarization system were improved.

The experimental results showed that enriching the representation of sentences with coreference embedding vectors and using them in RNN-based summarization systems, led to improved summarization results. The evaluation results of our model using ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-SU4 measures showed that the proposed method in most cases achieved superior results over other methods.

For future studies, we think that sentence encoding phase is flexible and could be significantly improved by different recurrent layer structures. Also, we plan to further explore combining deep neural networks like convolutional neural network with coreference resolution information. Further, we plan to construct coreference embedding vector in other ways. In the end, we hope to take a small step to help researchers and the medical community in the COVID-19 pandemic situation by providing the possibility to quick understanding of the COVID-19 research articles.

References

1- Z.Y. Zu, M.D. Jiang, P.P. Xu, *et al.*, "Coronavirus disease 2019 (COVID-19): a perspective from China," *Radiology*, p. 200490, 2020.

2- A. Spinelli and G. Pellino, "COVID-19 pandemic: perspectives on an unfolding crisis," *The British journal of surgery*, 2020.

3- B.P. Joshi, V.D. Bakrola, P. Shah, *et al.*, "deepMINE-Natural Language Processing based Automatic Literature Mining and Research Summarization for Early Stage Comprehension in Pandemic Situations specifically for COVID-19," *bioRxiv*, 2020.

4- M. Yousefi-Azar and L. Hamey, "Text summarization using unsupervised deep learning," *Expert Systems with Applications*, 68, p. 93-105, 2017.

5- K. Duraiswamy, "An approach for text summarization using deep learning algorithm," 2014.

6- Z. Cao, W. Li, S. Li, *et al.*, "Attsum: Joint learning of focusing and summarization with neural attention," *arXiv preprint arXiv:1604.00125*, 2016.

7- Y. Zhang, M.J. Er, R. Zhao, *et al.*, "Multiview convolutional neural networks for multidocument extractive summarization," *IEEE transactions on cybernetics*, 47(10), p. 3230-3242, 2016.

8- R. Nallapati, F. Zhai and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in *Thirty-First AAAI Conference on Artificial Intelligence*. of Conference., Year.

9- P. Dhakras and M. Shrivastava, "BoWLer: A neural approach to extractive text summarization," in *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*. of Conference., Year.

10- A. Rezaei, S. Dami and P. Daneshjoo, "Multi-Document Extractive Text Summarization via Deep Learning Approach," in *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*. of Conference.: IEEE, Year.

11- D. Su, Y. Xu, T. Yu, *et al.*, "CAiRE-COVID: A Question Answering and Multi-Document Summarization System for COVID-19 Research," *arXiv preprint arXiv:2005.03975*, 2020.

12- L. Dong, N. Yang, W. Wang, *et al.*, "Unified language model pre-training for natural language understanding and generation," in *Advances in Neural Information Processing Systems*. of Conference., Year.

13- M. Lewis, Y. Liu, N. Goyal, *et al.*, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

14- V. Kieuvongngam, B. Tan and Y. Niu, "Automatic Text Summarization of COVID-19 Medical Research Articles using BERT and GPT-2," *arXiv preprint arXiv:2006.01997*, 2020.

15- J. Devlin, M.-W. Chang, K. Lee, *et al.*, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

16- A. Radford, J. Wu, R. Child, et al., "Language models are unsupervised multitask learners," *OpenAI Blog*, 1(8), p. 9, 2019.

17- J.W. Park, "Continual BERT: Continual Learning for Adaptive Extractive Summarization of COVID-19 Literature," *arXiv preprint arXiv:2007.03405*, 2020.

18- C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*. of Conference., Year.

19- L.L. Wang, K. Lo, Y. Chandrasekhar, et al., "CORD-19: The Covid-19 Open Research Dataset," *ArXiv*, 2020.

20- O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Advances in neural information processing systems*. of Conference., Year.

21- Y. Bengio, R. Ducharme, P. Vincent, et al., "A neural probabilistic language model," *Journal of machine learning research*, 3(Feb), p. 1137-1155, 2003.

22- J. Pennington, R. Socher and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. of Conference., Year.

23- T. Wolf, J. Ravenscroft, J. Chaumond, et al., Neuralcoref: Coreference resolution in spacy with neural networks. 2018.

24- S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, 9(8), p. 1735-1780, 1997.

25- J. Chung, C. Gulcehre, K. Cho, et al., "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

26- S.-M.-R. Beheshti, B. Benatallah, S. Venugopal, et al., "Asystematic review and comparative analysis of cross-document coreference resolution methods and tools," *computing*, p. 1-37, 2016.

27- S.-M.-R. Beheshti, S. Venugopal, S.H. Ryu, et al., Big data and cross-document coreference resolution: current state and future opportunities. 2013, the university of new south wales: Sydney, Australia.

28- J. Steinberger, M. Kabadjov and M. Poesio, Coreference applications to summarization, in *Anaphora Resolution*. 2016, Springer. p. 433-456.

29- R. Pascanu, T. Mikolov and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*. of Conference., Year.

30- D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

31- Y. Bengio, P. Simard and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, 5(2), p. 157-166, 1994.

32- L. Wang and A. Wong, "COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images," *arXiv preprint arXiv:2003.09871*, 2020.