Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 3106631, 16 pages
https://doi.org/10.1155/2023/3106631

Research Article

Ensemble Text Summarization Model for COVID-19-Associated Datasets

T. Chellatamilan,[1] Senthil Kumar Narayanasamy,[2] Lalit Garg,[3] Kathiravan Srinivasan,[1] and Sardar M. N. Islam[4]

[1] *School of Computer Science and Engineering, Vellore Institute of Technology, Vellore 632014, India*
[2] *School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore 632014, India*
[3] *Faculty of Information and Communication Technology, University of Malta, Msida MSD2080, Malta*
[4] *ISILC, Decision Sciences and Modelling Program, Victoria University, Footscray, Australia*

Correspondence should be addressed to Sardar M. N. Islam; sardar.islam@vu.edu.au

Received 29 June 2023; Revised 28 October 2023; Accepted 25 November 2023; Published 11 December 2023

Academic Editor: Said El Kafhali

The work of text summarization in question-and-answer systems has gained tremendous popularity recently and has influenced numerous real-world applications for efficient decision-making processes. In this regard, the exponential growth of COVID-19-related healthcare records has necessitated the extraction of fine-grained results to forecast or estimate the potential course of the disease. Machine learning and deep learning models are frequently used to extract relevant insights from textual data sources. However, in order to summarize the textual information relevant to coronavirus, we have concentrated on a number of natural language processing (NLP) models in this research, including Bidirectional Encoder Representations of Transformers (BERT), Sequence-to-Sequence, and Attention models. This ensemble model is built on the previously mentioned models, which primarily concentrate on the segmented context terms included in the textual input. Most crucially, this research has concentrated on two key variations: grouping-related sentences using hierarchical clustering approaches and the distributional semantics of the terms found in the COVID-19 dataset. The gist evaluation (ROUGE) score result shows a significant and respectable accuracy of 0.40 average recalls.

1. Introduction to Text Summarization

As web development has advanced to a new level in recent years, there is a greater need than ever for efficient text summarizing techniques for a variety of practical uses. Text summaries are typically used to extract potential information from text documents and provide a meaningful summary of the content. It is additionally seen as a beneficial substitute for information overload. Text summarization seeks to extract the appropriate representative subset of the provided text documents and collaboratively finds the inherent semantic meanings by determining the key subjects of the textual content utilizing some of the conceptual viewpoints. The technical method of extracting and abstracting precise brief summaries from a large text source is known as text summarization [1]. Text summarizing typically uses one of two main mechanisms: extractive text summarization or abstractive text summarization. The extractive TS locates, highlights, and extracts the essential phrases from the source text before combining them to effectively summarize the entire text. It is really simple and constantly checks for proper grammatical structure. It has primarily been employed for lengthy texts that offer more focal points for summarizing the text. The location of the text passage where the summarizing process should pay special attention is designated as the focal point. By maintaining the keywords and phrases from the original content, abstractive summarization (AS) can, on the other hand, summarizes the documents. It reduces some of the textual grammatical irregularities that extractive summarization produces (ES).

Although the summary appears to be accurate, it is actually quite repetitious. Additionally, it was not effective for large text documents [2] since a single fixed-length vector used to summarize the given text sequence significantly loses information. This extractive summarization approach would primarily have an impact on text summarization accuracy. The encoder-decoder neural network (NN) model has done incredibly well when used with the abstractive summarization technique for brief text [3]. The multilayered long short-term memory (LSTM) is utilized for a long input text sequence and remembers the long sequence of text for predicting the delicate words, effectively solving the problem. For some predefined datasets, these models were performing well. However, there are still significant research gaps and restrictions. The following list summarizes the main deficiencies and limitations: The word embedding effect is inappropriate because the input datasets vary in their levels of ambiguity, which further prevents semantic textual entailments from working. Similar to this, many NLP-based initiatives produce erroneous results due to a lack of contextual word representation. The important text summary models are shown in Figure 1. By employing these summarization models (Seq2Seq, BERT, Attention), the research aims to derive valuable insights from the COVID-19 dataset, making it more accessible and comprehensible for analysis and decision-making in the context of the pandemic.

We used the pre-trained language model known as the Bidirectional Encoder Representation of Transformers (BERT) [4], which is frequently implied in many natural language projects, in light of the aforementioned drawbacks. With a big data corpus as training, the BERT is well equipped to provide superior sequence word embedding. The semantic importance of text documents can be efficiently estimated using the vectors' similarity. For natural language processing (NLP) applications, Word2Vec (Word Vector) [5], Glove (Global Vector) [6], BERT [7], etc., are the most often utilized word embedding. These models will take into account a number of strategies to condense the textual information about coronavirus. By displaying the benefits and drawbacks of each model, it will compare the performance of the models. The attention neural networks will be used to construct an ensemble model. This research significantly advances the field of text association by offering a novel COVID-19 text summarizing model that has surpassed other prior experiments. The entire process involves considering taking the sentences from COVID-19 datasets and effectively retrieves the distributional semantics of the sentences from the techniques such as Word2Vec, Glove, and BERT and then apply hierarchical clustering to group the sentences based on their semantic similarity. This sort of approach has been pervasively used for the NLP tasks such as topic modeling, text summarization, information retrieval, and sentiment analysis. Hence, in this research work, we have proposed this ensemble model that has been designed to be efficient in terms of both memory occupancy and efficiency that provides reasonably good performance for the assigned task.

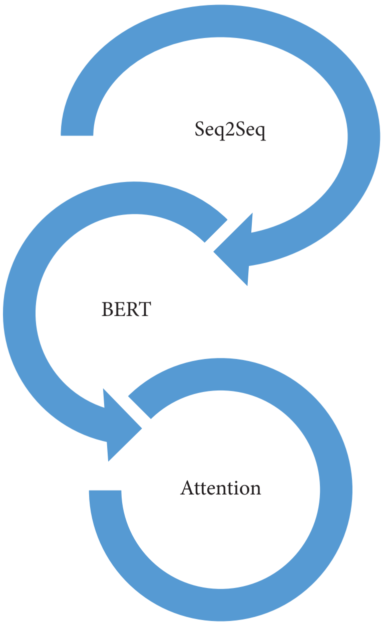


Figure 1: Prominent text summarization models.

1.1. Structure of the Paper. The rest of the paper is organized as follows. Section 2 critically reviews the literature on the extractive summarization process and highlights the critical text summarization proposals to motivate the present work. Section 3 critically analyses the TS mechanisms implemented through three prominent text summarization techniques and highlights their limitations. These text summarization mechanisms are BERT, Sequence-to-Sequence, and Attention Mechanisms. Section 4 presents the results obtained based on the three summarization techniques used in the paper and highlights the underlying differences through appropriate measures, such as Recall-Oriented Understudy for Gist Evaluation (ROUGE). Conclusions and directions for further research are presented in Section 5.

2. Related Works in Text Summarization and the Motivation

With the emergence of COVID-19, many research institutes such as Allen Institute for AI [8] had profusely accumulated coronavirus datasets. It is mainly to help research communities, particularly the general public, and to pervasively explore the meaningful insights from the COVID-19 dataset. The COVID-19 Open Research Dataset (CORD-19) Search [9] is deployed as an effective search engine that provides a semantic search platform to query the CORD-19 dataset. Likewise, Covidex is a multistage search operation designed to filter the various features related to the COVID-19 dataset. In this connection, the authors of [10] deployed an NLP-based medical inference engine (i.e., called WellAI) to accumulate medical-related concepts with appropriate ranking mechanisms and produce a structured list of concepts with high precision and recall scores. The Tmcovid tool [11] was effectively utilized to populate sufficient biorelated concepts and further disambiguate the mostly ambiguous terms. Later, with the advent of sequence-to-sequence models proposed by the authors of [12], it gained massive research attention for NN-based NLP systems and produced qualitative results with high precision.

Earlier, the research communities widely used LSTM-based approaches in applications such as image captioning, text categorization, entity classification, and speech recognition. LSTM is the alteration of the recurrent NN (RNN). LSTM has pervasively been used for effective text summarization processes and made text summarization possible, particularly for the abstractive summarization. It scores comparatively well on the extractive summarization. The authors [13] proposed a novel approach to predict the input's core parts and deeply apply the attention mechanism with suitable transformers to summarize and translate the given input effectively. The summarization process is mostly extractive because it can effectively detect the input's potential keywords through weight and ranking mechanisms [14]. The extractive summarization [15] is just a reproduction of the top–k-rank sentences. The document understanding conference (DUC)-2003 and DUC-2004 [16] competitions standardized the abstractive summarization and enabled practitioners to gather more popular new stories on divergent topics from different sources and later to analyze the stories for their summarization correctness.

In 2004, DUC-2004 recognized TOPIARY [16] for its attempt to couple both linguistics techniques and unsupervised algorithms in providing standard compressed results. Later, DUC-2004 was used to recognize some abstractive summarization processes. DUC-2004 was also used to formulate the conventional phrase table based on some machine translation approaches, compression using weighted tree transformation rules [17], and quasi-synchronous grammar approaches [18]. Latent semantic analysis (LSA) [19] is an algebraic learning algorithm that has been predominantly used in research fields such as information retrieval, text summarization, entity categorization, and image classification. As the appropriate culmination of statistical and algebraic approaches was taken, the LSA can potentially detect the words' inherent structure and their context by singular value decomposition (SVD) [20] through its input matrix and document representations. The conventional methods such as Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) [21] did not give a correct document matrix for the input. Hence, the results were not considered for further evaluation.

Both BoW and TF-IDF models usually required some external documents to calculate the sentence similarity and process the results precisely based on the document matrix generated for the supplemented document. Later, word embedding methods were introduced to find the context associated between every two pairs of words and able to match the words to their semantic roots. The word embedding tries to continuously learn the vector representation of the words and identify the syntactic and semantic inference through its neural network techniques. The advantage of the word embedding method is that it does not require any other external document for knowledge evaluation and infers the patterns based on the context given in the entire document. However, inferring the correct context for the words requires some unlabeled input data for establishing the semantic space. The word embedding method can precisely define each word's meaning in the processed input document through its semantic space and infer the correct contexts associated with every document pair. Recently, the word embedding method extended into sentence- and document-level embedding [22]. The deep NN methods have gained immense popularity in recent times, and it has been widely used in some applications such as text summarization, entity disambiguation, and fake news detection. However, recently, the deep neural network models have also been used for abstract and extractive summarization. The extractive query-oriented summarization model can create a feature space out of the term frequency generation. It develops the local word vectorization for each vocabulary in the input sentences. Likewise, the authors of [23] introduced the encoder-decoder model that increases the convolutional neural networks (CNN's) capabilities through its attentional model. The CNN algorithm has been used predominantly in image processing, but in recent years, its performance on sequence data analysis, such as named entity recognition (NER) and natural language processing (NLP), has become vibrant and made progressive attempts specifically in the field of artificial intelligence. This model effectively discards the full-sequence order while producing the input document's hidden representation and fixing the number of iterations based on the n-gram model principle. Similarly, the authors of [24] used the RNN-based sequence model for ES and top-k sentences were ranked based on the binary decision-making process. The authors of [25] attempted to use the attention mechanism to compute the query relevance based on sentence ranking, which converged randomly for every iteration. Table 1 lists some of the standard NLP methods and techniques used for text summarization.

3. Comparative Analysis of Extractive Text Summarization (ES) Methods

In this study, we selected three baseline models for effective extractive summarization, including BERT, sequence-to-sequence mechanism, and attention mechanism, and then present the ensemble approach to determine the variation in accuracy progress. Regarding the experimental outcomes, we noted variations in the baseline models. We also examined the modest variations that exist between the three baseline techniques that correspond to the NN translation model. A bidirectional gated recurrent unit (GRU)-RNN is always present in the encoder of the NN model [31], while a unidirectional GRU-RNN is stored in the decoder. The same hidden state is integrated by the encoder and decoder to the source hidden states, and the SoftMax layer is used to generate the target words from the extended vocabulary. The main motivation behind employing a GRU-NN encoder and decoder approach lies in the ability to train a unified model that operates on both source and target text sentences

Table 1: Some of the NLP methods and techniques used for summarization.

S. no	Model	Type	Architecture	Advantage
1	Word2Vec [26]	Prediction oriented	Log bilinear	Capable of representing the context words and most frequent words from the input document
2	Glove [27]	Count based	Log linear	Able to couple both Word2Vec and matrix factorization methods for identifying the co-occurrence list of words for the probable context generation
3	Fast text [18]	Prediction based	Log bilinear	Utilized for OOV words and unorthodox contents. It has been well trained in almost 157 languages
4	BERT [28]		LSTM	Uses the encoder and decoder for finding the hidden layer context for text generation
5	ELMo [29]	Prediction based	BiLSTM	Deals with polysemy tendency of words and eradicates the redundancy of co-reference words
6	PV-DBOW [30]	Prediction based	Log bilinear	It demands on word vectors and hence requires less memory for evaluation

simultaneously, accommodating varying lengths of input and output text sequences. In Figure 2, these encoders and decoders are integrated into various deep learning models, including Seq2Seq, Attention, and BERT. The bidirectional aspect is an integral part of the encoder's architecture, incorporating self-attention. The precise position and order of words within a sequence play a pivotal role in comprehending the overall meaning of sentences, especially in text summarization. In the encoder, word embeddings take into account both word positioning and sentence order. A complete description of each of the three baseline models, along with an explanation of their standard operating methods and pertinent empirical analysis, is provided in the parts that follow. This research project entails experimenting with the pre-trained Attention, Seq2Seq, and BERT models. The ranking of the final summation text of the input sentences was determined using an ensemble of these three summarization models, and the top N summarizations were gathered for performance comparison. For various degrees of the ROUGE score, the ensemble model and the baseline model of summarization have also been contrasted. The ensemble model for the task of text summarization is shown in Figure 2.

3.1. Extractive Summarization (ES) Using BERT. The extractive summarization is highly difficult for many NLP systems to understand, as noted by the authors of [32], but it has made good progress in recent years, thanks to the development of the BERT model, which provides improved embedding with transformer models. A decent summarizer should be able to scan the full text for intrinsic meanings and select sentences based on how the articles are internally embedded. The TextRank model [33] was chosen as the foundational strategy to guarantee its accuracy. On specific benchmarked references or any predetermined gold summary, the key problems for evaluating the text summarization are based. Finding the corpus needed to evaluate fresh information on unique subjects is becoming increasingly difficult. Therefore, the standard measure for the summary evaluation can be tested using the Recall-Oriented Understudy for Gist Evaluation (ROUGE-N) metric, which is accepted. Between the gold summaries and a few predetermined categories, the ROUGE-N can estimate the creation of N-grams. To efficiently summarize the content created by machines, the ROUGE-N would measure the words. The BERT model is particularly effective in parsing the meaning of the provided articles and papers and eliminates stop words, stem words with their root terms, and lowercase all text for simple transformation. Table 2 lists a few of the pre-trained BERT models.

To accomplish this objective, we tokenized the input material using the space package [40] and embedded the significant tokens using BERT through the sentence transformer package to maybe acquire some insights for the provided article/documents. The average tokens contained in the sentences are used to establish the document's standard mean, and the meaningful tokens are given more weight. In order to effectively disambiguate and identify each sentence in the article or document, we have additionally given each sentence a weight. The algorithm that determines the score for each article category using the binning technique would be completely responsible for the absolute labeling of the extracted summary. Also, it determines the exact match of the extraction summary through subsequent stages, which are essential. We significantly changed the BERT model to meet the needs of our extractive summarization in order to make it extremely effective.

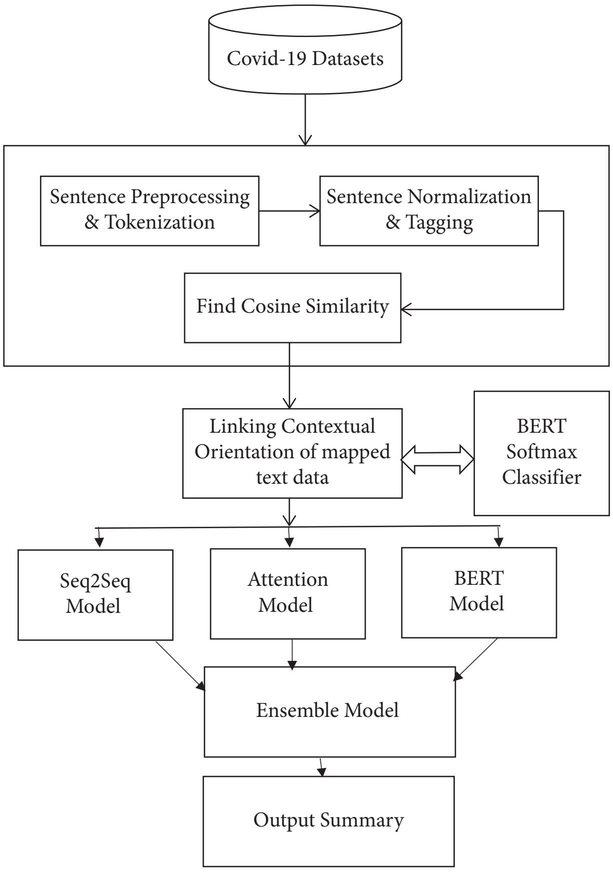


FIGURE 2: Architecture of the text summarization with ensemble model.

(i) Step 1: Load the COVID-19 related datasets and feed them into the BERT model.

(ii) Step 2: Find the cosine-similarity matrix between the two vectors C and D with equal dimensions of BERT hidden layers.

(iii) Step 3: Calculate each labeled token's probability as yielded by the dot product of C and the token represented in BERT's final hidden layers, followed by a SoftMax of the document's entire token.

(iv) Step 4. The final summary of the BERT model is computed using the token return probability after the document's end with calculated similarity Vector D.

When creating the summary for the given document, we first attempt to determine the sentence weight for each sentence in the documents using the dot product between C

Table 2: Some of the pre-trained BERT models for evaluation.

BERT models	Datasets used	Learning type	Task type	Accuracy rate/BERT (%)
ALBERT [28]	GLUE/SQuAD/RACE	Supervised/unsupervised	Question & answering	88.7/85.2
BLUEBERT [34]	PubMed/MIMIC-III	Supervised/unsupervised	Named entity recognition (NER)	89.2/86.9
BoostingBERT [35]	GLUE	Ensemble learning	NLP	82.93/80.72
SciBERT [36]	Scientific text	Supervised/unsupervised	NER	99.01/88.85
KnowBERT [37]	Wikipedia	Supervised/unsupervised	Relation extraction	89.01/89
BioBERT [38]	PubMed abstracts	Supervised/unsupervised	NER & QA	89.04/88.30
FinBERT [39]	Financial datasets	Supervised/unsupervised	Sentiment analysis	86

and D. After that, order the phrases from the highest to the lowest weight before selecting the top k for the summary. The link between the summarized text and the original input material is determined using the Recall-Oriented Understudy for Gist Evaluation (ROUGE), a common scoring algorithm. In essence, the precision calculation is done to guarantee the ROUGE-N accuracy rate. If it is determined that the trigram and the summary S overlap, the dot product of C should be discarded, and the remaining computed candidate sentences in D should be removed. The ROUGE score serves as a standardized method for evaluating the performance of text summarization and text translation models. Various ROUGE scale variations exist to gauge the degree of correspondence between a generated summary and its original reference summary, including ROUGE-N, ROUGE-L, ROUGE-S, and several others. This metric provides a reference or relative measurement that can be compared to human evaluations. In ROUGE-N, "N" represents N grams; this can be 1 or 2, denoting unigrams and bigrams respectively. ROUGE-L employs "L" to signify the longest common subsequence (LCS) of words that match between the candidate text and the reference summary, with a strong emphasis on preserving word order. When the preservation of word order in sentences is crucial, as is often the case in text summarization, the ROUGE-L score is utilized. The term gramme (N-gram) is indicated by the letter "N" in this sentence. The maximum length for position embedding in the original BERT model [41] is 512. We have overcome this restriction by incorporating a few extra position embeddings in other encoder settings. In order to possibly distinguish between distinct sentences in the imported document, we have additionally included some intermediary segment embeddings.

Table 3 depicts the performance of BERT model that we executed on the dataset COVID-19 and registered the total running time of every forward pass of the BERT model.

In comparison to the other two models discussed in this study, the BERT model summary used during the summarization process achieved 40% accuracy while using only 20% of the test data. Finally, the dense layer of the model summarizes the condensed summary of the input text while the dropout layer of the model prevents overfitting. During the process of developing the model with many rounds of epochs, we employed the Adam optimization strategy with the cross entropy loss function. In this study, the BERT model has been applied in two forms: BERT-base and BERT-large. With 110 million characteristics, the BERT base has 12 transformer layers and 12 attention layers. With 340 million parameters, the BERT big models contain 24 transformer layers and 16 attention layers. The first layer that accepts the input of max len is the input layer (512). This length was achieved by padding the input sentences. In order to prevent overfitting, the output of the transformer is sent as input to the drop layer. Finally, the activated dense layer provides the summary of our text input.

Sequence-to-sequence model with two encoder LSTM layers and two decoder layers is the other model employed in this study. Here, the input sentences were lengthened to a maximum of 30 before being processed through an embedding layer to create embedding word vectors for each word that was included in the input text. The output of the embedding layer is then transmitted through two LSTM layers—two encoding layers with padding input lengths of 300 and two decoding layers—before being decoded. The attention layer produces a compressed summary of the provided test text data list. The comparison results of various text summarizing techniques are shown in Table 4.

When compared to other text summarization models, the BERT model has done remarkably well, as seen in Table 4. ROUGE-1 has been used to conduct the evaluation while taking into account fundamental characteristics like count-vectorization, TF-IDF score, and Soft Cosine Similarity measure. The COVID-19 datasets were taken into account when the algorithms listed in Table 4 were being evaluated, and their accuracy rate was recorded for benchmarking.

3.2. Sequence-to-Sequence (Seq2Seq) Mechanism to Summarize Text.

Deep neural network models [42] have recently benchmarked their performance in text analytics, spanning several sectors. The recurrent neural network (RNN) model has mostly been employed for sequence modelling and language creation tasks. However, due to some expanding gradient concerns, the typical RNN model has had some trouble training the datasets for text summarization. The long short term memory (LSTM) model has typically been employed to address gradient difficulties, but it has not provided the appropriate level of judgment for text summarization. Additionally, the RNN-based computation experienced some problems locating previously hidden states and had problems with sequential dependency sequences. As a result, the RNN was unable to assess the memory and computation requirements of lengthy text document sequences. As a result, we used large collections of lengthy texts as the input for the sequence-to-sequence the deep

Table 3: The average running time of BERT model of a forward pass.

Models	ROUGE-1	ROUGE-2	ROUGE-L	Max sequence length	Parameter (M)
BERT-base	43.20	21.26	38.89	512	110
BERT-large	43.62	21.87	39.12	512	145

Table 4: Comparison result of text summarization algorithms.

Algorithms used	ROUGE-1	Count vectorizaiton	TF-IDF score	SoftCosine similarity
KNN	0.071	0.750	0.630	0.790
TextRank	0.001	0.720	0.660	0.850
SVM	0.065	0.732	0.650	0.690
Word2Vec	0.057	0.684	0.643	0.623
BERT	0.840	0.810	0.783	0.810

learning model and an attentional mechanism [43], with the intended output being the condensed summary. The developed model takes as input a big sentence of text and outputs a concise summary of it. Assume that the input text is made up of a succession of "I" words, such as T1, T2. The acronym TI was developed from a fixed-size vocabulary of the summary, which takes in T as input and produces the condensed text phrase "S" with length J, even if S is substantially smaller than "T" (J < I). The straightforward sequence-to-sequence model [17] for text summarization is shown in Figure 3.

> Encoder: The embedding layer has first converted each word in the input encoder into an embedding word vector for the distributional representation of the entire sentence. For all iterations, we processed the text in left-to-right and right-to-left directions using a bi-directional LSTM model [44].
>
> Decoder: The decoder receives the final word of the input sentence, eats it, and then uses a hidden layer unit to produce the output summary word. In the sequential processing of the text, the decoder provides the same word as input for producing the following word in a greedy manner.

The stepwise procedure from sequence-to-sequence (Seq2Seq) is illustrated below:

> Step 1: Let us assume the lengthy text "T" as the input to the encoder and the summary text "S" as the output of the decoder of the Seq2Seq model. Let the top "N" most likely words be v1, v2, v3, . . ., vn as per the decoder network output over the vocabulary V.
>
> TEXT (T): In the United States of America, the coronavirus death is 1 Million.
>
> Summary Text(S): *US COVID-19 death 1L*
>
> Step 2: The next possible word in the sequence is predicted if S1 has already occurred using the conditional probability formula P(S2|T, S1) by maximizing the probability of S1 and S2 occurring together using $P(S1, S2/T) = P(S1/T) * P\ (S2/T, S1)$
>
> Step 3: Similarly, the third possible word has been determined using the conditional probability $P(S3/T, S1, S2)$ by maximizing the probability of S_1, S2, and S3.

The above steps would be repeated until the end of a sentence is reached in the sequence of processing.

To summarize the COVID-19 related datasets, the selection of hyper parameter set "transformer_prepend" has been introduced and utilized the tensor2tensor library for effective filtering and categorization. The comparison of the most important hyper parameter differences has been laid out in Table 5.

3.3. Attention Mechanism for Text Summarization. Our model incorporates the attention mechanism [41] that enables the decoder to assign various weights and to review earlier words in the input sequence before generating the next word. The attention function of the decoder enables it to use contextual data pertaining to various input segments. Finally, the focus makes certain that the model employs several input segments with different weights, increasing the information coverage during the summarization phase [45]. When creating the relevant summary word in the output, the attention mechanism further concentrates on and remembers just specific passages from the input text. The attention model creates a context vector for each output it comes across rather than encoding the input sentence into a single, fixed-length context vector [46]. The attention mechanism takes into account every word in the summary output and generates just the most significant words from the input text by giving these words a higher weight. The attention mechanism [47] for condensing the content of the guidelines is shown in Figure 4.

Algorithm 1 exploits the step-by-step procedure towards accepting the text input and generates the condensed summary by applying the embedding and encoder decoder with an attention mechanism.

4. Experimental Analysis

The COVID-19 related guideline dataset was collected from various trusted sources [48–50] and authenticated sites such as Centers for Disease Control and Prevention (CDC), (https://www.cdc.gov/coronavirus/2019-ncov/index.html) Minister of Human Resource Development, Govt of India (MHRD) (https://hrm.mhrd.gov.in/home), and Indian Council of Medical Research (ICMR) (https://www.icmr.gov.in/). The

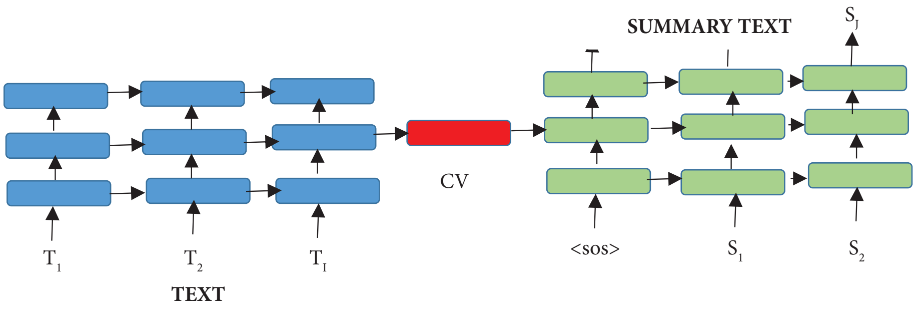


Figure 3: Simple sequence -to-sequence model for summarization of text.

Table 5: Selection parameter for hyperparameter settings.

Hyperparameter	TPU	Transformer prepend	Base_v1	Base_v3
Optimizer	Adafactor	Adam	Adam	Adam
Attention dropout	0.2	0.2	0.2	0.2
Learning rate	0.1	0.2	0.1	0.2
Relu_Dropout	0.1	0.1	0	0.1
Layer_preprocess_sequence	da	da	dan	da
Layer_postprocess_sequence	n	n	None	n
Learning_rate_constant	3	2	1	3

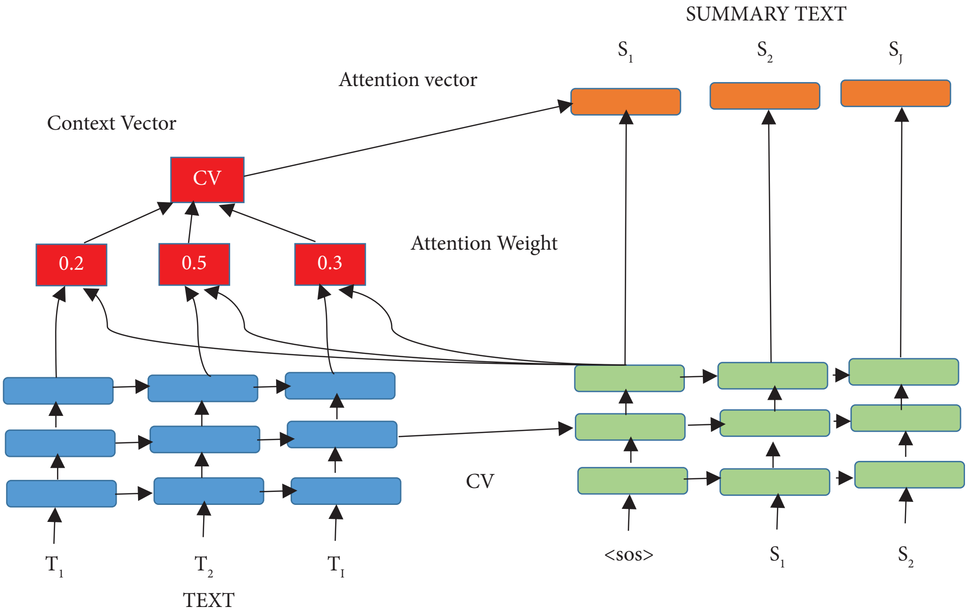


Figure 4: Attention mechanism for summarizing the guideline text.

frequently asked questions (FAQs) and their corresponding answers were also collected to construct the COVID-19 summarization dataset. In contrast, the question is viewed as a summary, and the corresponding answer is assumed as its lengthy sentence text. Analyzing the answer text and preparing the related question summary through the manual is time-consuming and leads to more ambiguity in the text summarization process.

Our objective is to summarize the lengthy guideline text using deep learning model-based techniques. The dataset consists of more than 500 guideline texts related to various information covering topics such as summary guideline texts, HTML links, categories, countries, cities, region, and GPS information. The initial data processing and data cleaning tasks were applied to the dataset to fine-tune the dataset suitable to build the model more effectively and efficiently. We used the Keras library [51–53] to remove the stop words, drop the duplication, and avoid the NA (not available) summary/text values. The unwanted symbol characters and punctuations were removed potentially without affecting the objective of the solution. A separate dictionary of words is also used to expand the contradictory words such as can't and couldn't. Special tokens such as <SOSTOK> and <EOSTOK> were added to the summary to

```
Input: The guideline data related to COVID-19 is the input to the deep learning network, initialization of attention weight
Output: Trained Model, Summary Text
Step 1: SourceDoc = Open(SoureDoc)
Step 2: Vocab = ExtractGuidelineVocab()
Step 3: Onehot = GetOneHotEncoding(Vocab)
Step 4: EmbeddingInput = GetEmbedding(onehot)
Step 5: ContextVector = Encoder(EmbeddingInput)
Step 6: DecoderInput = GetDecoderInput(ContextVector, AttentionWeight, SummaryInput)
Step 7: Training Phase
Step 8: TrainingEncoder = EncoderStack (EmbeddingInput, <SOStok>, <EOStok>)
Step 9: TrainingDecoder = Int_Decoder (LSTMStack, TrainingEncoder)
Step 10: TrainingDecoderOutput = Decoder (TtainigDecoder)
Step 11: For epochs in range (1,500) do
Step 12: Loss = MeasureLoss(CrossEntropy, TraingDecoderOutput, SummaryText)
Step 13: Return Model
Step 14: CallModel_Fit (TestText, TestSummary)
Step 15: Measure_Performance()
Step 16: Plot (measures)
```

Algorithm 1:Attention mechanism for summarization of text.

indicate the beginning position and the individual summary's end position. The plot shown in Figure 5 represents the frequency distribution of the words present in the summary and guideline text. The percentage or proportion of rare words is also estimated to fix our model building's words' that set the threshold for the frequent occurrence of words.

The model is trained with a sample training set, and its performance is tested with the validation split. The training phase has taken 90% of the dataset and the remaining 10% of the dataset is used for validation to evaluate the performance of the model. The model building part has the composition of the following layers to perform the deep learning task. The stacked encoder decoder with an attention mechanism has the following model summary.

(1) Embedding layer

(2) Encoder LSTM layer (1 to 3)

(3) Decoder LSTM layer

(4) Attention layer

(5) Dense layer

Our model will not learn the non-trainable parameters from weighted vectors of the embedding matrix. The check point facility in Keras helps us to save these best weights and has been used for early stopping of the model in 10 epochs. We have used the embedding layer to convert the integer sequence of words of text and summary into one-hot-vector method with their semantic meaning. The categorical cross entropy cost function is used for fine tuning the model. The epoch versus loss plot is shown in Figure 6.

During our training process, we evaluate the proposed model performance based on hold-out validation and intense training on the COVID-19 dataset. Then, we plot the major performances of the model through each training step, i.e., each epoch of an ensemble model tree. These learning curves help to review this model and diagnose the learning processes, such as overfit or underfit model. The underfitting models represent that the training dataset has not learned sufficiently and produces low training error values. On the other side, the overfitting model has learnt the model so well and produces more statistical details and other random fluctuations in the given training datasets.

4.1. Performance Assessment and Evaluation

4.1.1. ROUGE (Recall-Oriented Understudy for Gist Evaluation). ROUGE is a metric used for measuring the score/accuracy of the summarization task based on recall [54]. It evaluates the score by finding the relation between the number of overlapping (matched) words in the predicted and original summaries.

4.1.2. ROUGE Recall

$$\frac{\text{count of overlapping words}}{\text{count of total words in reference summary}}. \tag{1}$$

4.1.3. ROUGE Precision

$$\frac{\text{count of overlapping words}}{\text{count of total words in predicted summary}}. \tag{2}$$

In ROUGE-N, the value N refers to overlapping n-grams. The notational expression for obtaining the score can be written as follows:

$$\text{Rougn Score}\, n = \frac{\sum_o \sum_p \text{overlap}\left(\text{ngram}_p\right)}{\sum_o \sum_p \text{TotalCount}\left(\text{ngram}_p\right)}, \tag{3}$$

where "o" refers to the count of overlapping words present in the original and reference summaries and "p" refers to the count of the predicted/proposed set of summaries by algorithms.

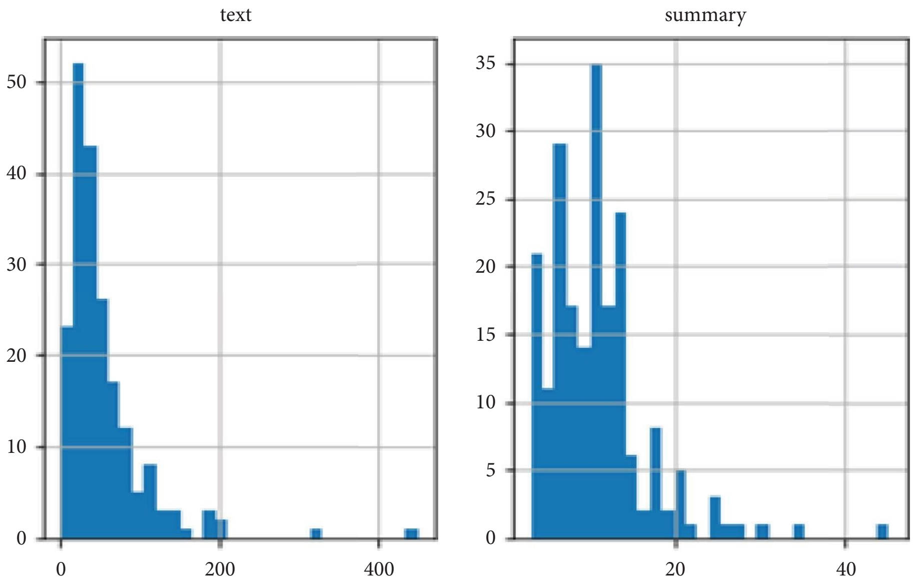


Figure 5: Frequency distribution of vocabulary of guideline text and summary.

Let us assume that we are calculating ROUGE-2, aka bigram matches. The numerator $\sum_o \sum_p$ loops through all bigrams in a single original summary and calculates the number of times an overlapping (matching) bigram is found in the candidate summary. This process of calculating the score is repeated for the overall reference summaries present in our test set [7, 55]. The denominator simply counts the total number of bigrams in all reference summaries. The ROUGE scores for the baseline, BERT attention model, and Seq2Seq pre-trained summarization models for Top 7 Guideline Texts are shown in Tables 6–8, respectively. Figure 7 represents the ROUGE score chart of BERT for Top 7 Guideline Texts. Figure 8 illustrates the ROUGE score chart of the Attention Model for Top 7 Guideline Texts. Figure 9 portrays the ROUGE Score chart of Seq2Seq Model for Top 7 Guideline Texts. The precision, recall, and F-measure scores of ROUGE-i have been, respectively, notated as RiP, RiR, RiF in the respective figures, whereas "Ri" refers to ROUGE score at the *i*th level of ROUGE and "*P*" refers precision refers recall and F refers F-measure.

Table 9 shows the average ROUGE score of three different models that we have built using the deep learning approach. Upon comparing the scores of such models, the BERT pretrained model outperforms in the process of summarization of the textual guidelines and generates the condensed summary of the COVID-19 dataset.

Figure 10 shows the details of the extractive text summary generated by the three baseline models.

4.1.4. Ensemble Approach of Text Document Summarization.

Finally, we integrated every model we created using the ensemble approach, which we usually employed for all kinds of machine learning tasks. The project represents experimental work using the Seq2Seq, Attention, and pre-trained BERT models. The ranking of the final summation text of the input sentences was determined using an ensemble of these three summarization models, and the top N summarizations were gathered for performance comparison. For various degrees of ROUGE score, the ensemble model and the baseline model of summarization have also been contrasted. The outcomes of the ensemble model for text summarization are shown in Table 10.

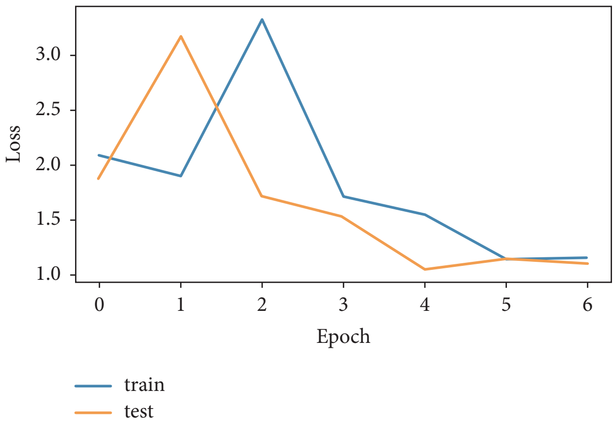


Figure 6: Epoch versus loss performance curve.

The different levels of ROUGE scores were evaluated through the correlation co-efficient between ROUGE scores and the reference summary. Figure 11 represents the comparison of the performance metrics of the ensemble model and baseline models.

Table 6: ROUGE scores -BERT model -top 7 guideline texts.

Sentence	R1F	R1P	R1R	R2F	R2P	R2R	RLF	RLP	RLR
1	0.056	0.045	0.071	0	0	0	0.059	0.05	0.071
2	0.625	0.714	0.556	0.571	0.667	0.5	0.625	0.714	0.556
3	0.05	0.03	0.143	0	0	0	0.059	0.036	0.167
4	0.098	0.056	0.4	0.051	0.029	0.25	0.105	0.061	0.4
5	0.098	0.06	0.273	0	0	0	0.077	0.049	0.182
6	0.029	0.019	0.067	0	0	0	0.04	0.029	0.067
7	0.143	0.088	0.375	0	0	0	0.162	0.103	0.375

Table 7: ROUGE scores -attention model -top 7 guideline texts.

Sentence	R1F	R1P	R1R	R2F	R2P	R2R	RLF	RLP	RLR
1	0.182	0.13	0.3	0	0	0	0.187	0.13	0.333
2	0.093	0.069	0.143	0	0	0	0.105	0.083	0.143
3	0.211	0.2	0.222	0	0	0	0.105	0.1	0.111
4	0.227	0.135	0.714	0.095	0.056	0.333	0.162	0.097	0.5
5	0.068	0.037	0.4	0.035	0.019	0.25	0.075	0.042	0.4
6	0.19	0.2	0.182	0	0	0	0.19	0.2	0.182
7	0.381	0.276	0.615	0.1	0.071	0.167	0.3	0.222	0.462

Table 8: ROUGE scores -Seq2Seq model -top 7 guideline texts.

Sentence	R1F	R1P	R1R	R2F	R2P	R2R	RLF	RLP	RLR
1	0.061	0.043	0.1	0	0	0	0.062	0.043	0.111
2	0.054	0.043	0.071	0	0	0	0.056	0.045	0.071
3	0.316	0.207	0.667	0.222	0.143	0.5	0.294	0.2	0.556
4	0.078	0.045	0.286	0	0	0	0.103	0.061	0.333
5	0.077	0.043	0.4	0.04	0.022	0.25	0.087	0.049	0.4
6	0.2	0.222	0.182	0	0	0	0.2	0.222	0.182
7	0.121	0.075	0.308	0	0	0	0.17	0.118	0.308

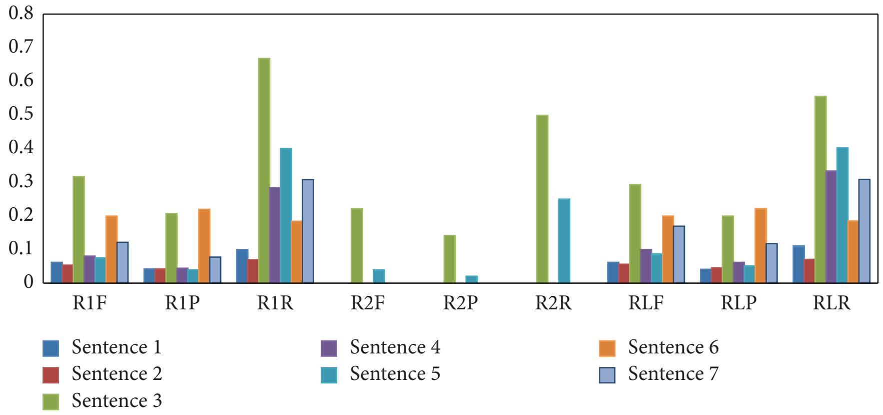


Figure 7: ROUGE score chart of summarizer with BERT for top 7 guideline texts.

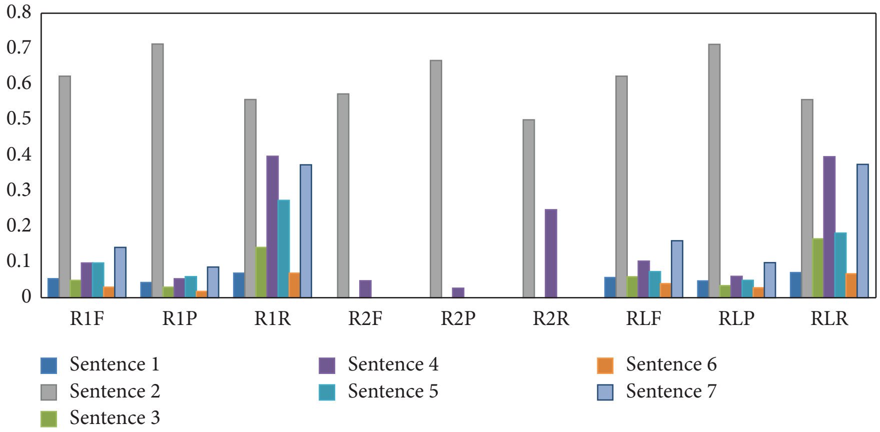


Figure 8: ROUGE score chart of attention model for top 7 guideline texts.

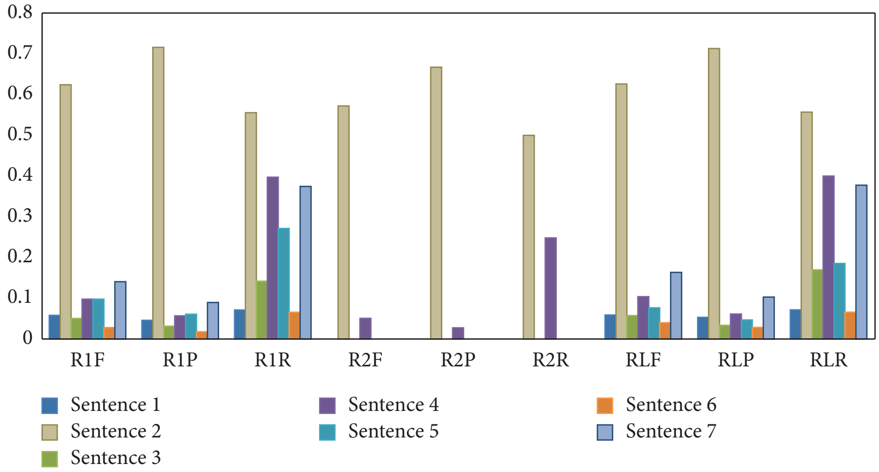
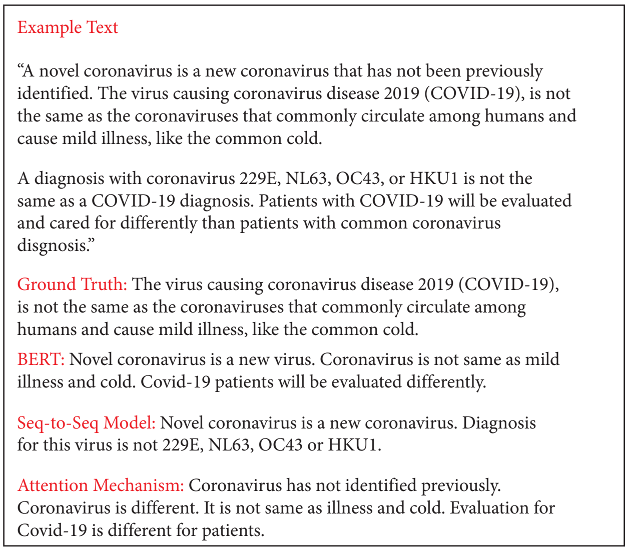


Figure 9: ROUGE score chart of Seq2Seq model for top 7 guideline texts.

Table 9: Average ROUGE score results for the baseline line models.

Model	R1F	R2F	RLF	Average
Baseline-Seq2Seq	0.135	0.213	0.140	0.196
Attention model	0.193	0.032	0.160	0.3865
BERT pre-trained model	0.157	0.88	0.161	0.40



Example Text

"A novel coronavirus is a new coronavirus that has not been previously identified. The virus causing coronavirus disease 2019 (COVID-19), is not the same as the coronaviruses that commonly circulate among humans and cause mild illness, like the common cold.

A diagnosis with coronavirus 229E, NL63, OC43, or HKU1 is not the same as a COVID-19 diagnosis. Patients with COVID-19 will be evaluated and cared for differently than patients with common coronavirus disgnosis."

Ground Truth: The virus causing coronavirus disease 2019 (COVID-19), is not the same as the coronaviruses that commonly circulate among humans and cause mild illness, like the common cold.

BERT: Novel coronavirus is a new virus. Coronavirus is not same as mild illness and cold. Covid-19 patients will be evaluated differently.

Seq-to-Seq Model: Novel coronavirus is a new coronavirus. Diagnosis for this virus is not 229E, NL63, OC43 or HKU1.

Attention Mechanism: Coronavirus has not identified previously. Coronavirus is different. It is not same as illness and cold. Evaluation for Covid-19 is different for patients.

Figure 10: Example of extractive text summarization.

Table 10: Results of ensemble model for text summarization.

Model	R1F	R2F	RLF	Average
Baseline- Seq2Seq	0.135	0.213	0.140	0.196
Base line attentional model	0.193	0.032	0.160	0.3865
BERT pre-trained model	0.157	0.88	0.161	0.40
Ensemble model	0.198	0.92	0.173	0.481

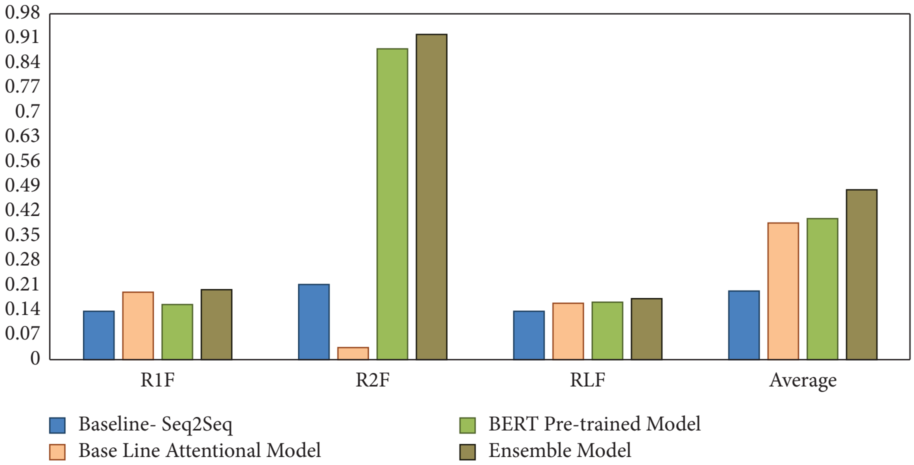


Figure 11: Comparison of the performance metrics of ensemble model and baseline models.

5. Conclusions

The datasets connected to COVID-19 have been efficiently summarized in this study using BERT models, Sequence-to-Sequence, and attention mechanisms. According to our analysis, the ensemble model fared quite well in the ROUGE examination. In order to produce the accurate summary for the loaded datasets, our ensemble model efficiently filters the semantic characteristics and extracts the words' implicit meaning. The main benefit of this suggested ensemble model is that it uses hierarchical clustering to connect related sentences and distributional semantics of the words for categorization. The integration of hierarchial clustering and distributional semantic approach creates a robust framework for text summarization and helps to gain the granular understanding of relationships between the sentences present in the COVID-19 dataset. These word embedding models enable to categorize the words into semantic clusters that reflect the appropriate meaning and context for the sentence. By employing the predefined threshold limit, this integrated technique facilitates the selection of the Top-k summaries and produces effective results.

5.1. Limitation. Even a large vocabulary size has not always helped the analysis in some cases. Similar to this, factual information was frequently produced improperly and with the inappropriate substitution of some popular tales for unusual words. This is considered the model's limitation.

5.2. Future Work. These tests were rigorously conducted using Google Colab and carried out performed on a single GPU resource. However, using fine-tuned models for big hyperparameters would not be suitable for efficient extractive synthesis [56, 57]. Additionally, rather than using domain-specific applications, we might search for an ensemble model that works for general extractive summarization. Only domain-specific datasets can be used with the suggested ensemble model. Similarly, we can attempt abstractive summarization for some datasets relevant to academia, which will yield positive outcomes for dropout analysis. Although we have not significantly reduced the size of the pre-trained model, using approaches such as pruning and quantization would have been very beneficial in this model.

Data Availability

The original contributions generated by this study are included in the article; further inquiries can be directed to the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

Open access publishing was facilitated by Victoria University, as part of the Wiley-Victoria University agreement via the Council of Australian University Librarians.

References

[1] M. Koupaee and W. Y. Wang, "Wikihow: a large scale text summarization dataset," 2018, https://arxiv.org/abs/1810.09305.

[2] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," 2017, https://arxiv.org/abs/1705.04304.

[3] F. Liu, J. Flanigan, S. Thomson, N. Sadeh, and N. A. Smith, "Toward abstractive summarization using semantic representations," 2018, https://arxiv.org/abs/1805.10399.

[4] Y. Liu, "Fine-tune BERT for extractive summarization," 2019, https://arxiv.org/abs/1903.10318.

[5] Y. Goldberg and O. Levy, "word2vec Explained: deriving Mikolov et al.'s negative-sampling word- embedding method," 2014, https://arxiv.org/abs/1402.3722.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, https://arxiv.org/abs/1301.3781.

[7] A. Savelieva, B. Au-Yeung, and V. Ramani, "Abstractive summarization of spoken and written instructions with BERT," 2020, https://arxiv.org/abs/2008.09676.

[8] C. Schoenick, P. Clark, O. Tafjord, P. Turney, and O. Etzioni, "Moving beyond the turing test with the allen ai science challenge," *Communications of the ACM*, vol. 60, no. 9, pp. 60–64, 2017.

[9] H. Kroll, J. Pirklbauer, J. Ruthmann, and W. T. Balke, "A semantically enriched dataset based on biomedical NER for the COVID19 open research dataset challenge," 2020, https://arxiv.org/abs/2005.08823.

[10] H. Yang, L. Luo, L. P. Chueng, D. Ling, and F. Chin, "Deep learning and its applications to natural language processing," in *Deep Learning: Fundamentals, Theory and Applications*, pp. 89–109, Springer, Cham, Switzerland, 2019.

[11] D. Su, Y. Xu, T. Yu, F. B. Siddique, E. J. Barezi, and P. Fung, "Caire-covid: a question answering and multi-document summarization system for COVID-19 research," 2020, https://arxiv.org/abs/2005.03975.

[12] M. Jayalakshmi, L. Garg, K. Maharajan et al., "Fuzzy logic-based health monitoring system for COVID'19 patients," *Computers, Materials & Continues*, vol. 67, no. 2, pp. 2431–2447, 2021.

[13] L. Garg, E. Chukwu, N. Nasser, C. Chakraborty, and G. Garg, "Anonymity preserving IoT-based COVID-19 and other infectious disease contact tracing model," *IEEE Access*, vol. 8, pp. 159402–159414, 2020.

[14] V. Kieuvongngam, B. Tan, and Y. Niu, "Automatic text summarization of COVID-19 medical research articles using BERT and GPT-2," 2020, https://arxiv.org/abs/2006.01997.

[15] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: a recurrent neural network based sequence model for extractive summarization of documents," *Proceedings of the AAAI Conference on Artificial Intelligence* in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, New York, NY, USA, February 2017.

[16] D. Zajic, B. Dorr, and R. Schwartz, "Bbn/umd at duc-2004: topiary," in *Proceedings of the HLT- NAACL 2004 Document Understanding Workshop*, pp. 112–119, Boston, MA, USA, May 2004.

[17] L. Wang, J. Yao, Y. Tao, L. Zhong, W. Liu, and Q. Du, "A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization," 2018, https://arxiv.org/abs/1805.03616.

[18] M. Yousefi-Azar and L. Hamey, "Text summarization using unsupervised deep learning," *Expert Systems with Applications*, vol. 68, pp. 93–105, 2017.

[19] S. T. Dumais, "Latent semantic analysis," *Annual Review of Information Science & Technology*, vol. 38, no. 1, pp. 188–230, 2004.

[20] A. G. Akritas and G. I. Malaschonok, "Applications of singular-value decomposition (SVD)," *Mathematics and Computers in Simulation*, vol. 67, no. 1-2, pp. 15–31, 2004.

[21] I. Sheikh, I. Illina, D. Fohr, and G. Linares, "Learning word importance with the neural bag-of- words model," in *Proceedings of the 1st Workshop on Representation Learning for NLP*, pp. 222–229, Berlin, Germany, August 2016.

[22] Y. Liu, C. Sun, L. Lin, and X. Wang, "Learning natural language inference using bidirectional LSTM model and inner-attention," 2016, https://arxiv.org/abs/1605.09090.

[23] T. Cohn and M. Lapata, "Sentence compression beyond word deletion," in *Proceedings of the 22nd International Conference*, Manchester, UK, August 2008.

[24] K. Woodsend and M. Lapata, "Automatic generation of story highlights," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 565–574, Uppsala, Sweden, July 2010.

[25] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," 2015, https://arxiv.org/abs/1509.00685.

[26] H. Liu, *Machine Learning: A Quantitative Approach*, CreateSpace Independent Publishing Platform, Carolina, CA, USA, 2019.

[27] N. Tomuro, S. Lytinen, and K. Hornsburg, "Automatic summarization of privacy policies using ensemble learning," in *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, pp. 133–135, New Orleans, LA, USA, March 2016.

[28] H. Choi, J. Kim, S. Joe, and Y. Gwon, "Evaluation of bert and albert sentence embedding performance on downstream nlp tasks," in *Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 5482–5487, IEEE, Milan, Italy, January 2021.

[29] Y. Peng, S. Yan, and Z. Lu, "Transfer learning in biomedical natural language processing: an evaluation of BERT and ELMo on ten benchmarking datasets," 2019, https://arxiv.org/abs/1906.05474.

[30] A. Fahfouh, J. Riffi, M. A. Mahraz, A. Yahyaouy, and H. Tairi, "PV-DAE: a hybrid model for deceptive opinion spam based on neural network architectures," *Expert Systems with Applications*, vol. 157, Article ID 113517, 2020.

[31] Z. Cao, W. Li, S. Li, F. Wei, and Y. Li, "Attsum: joint learning of focusing and summarization with neural attention," 2016, https://arxiv.org/abs/1604.00125.

[32] T. Bolukbasi, J. Wang, O. Dekel, and V. Saligrama, "Adaptive neural networks for fast test-time prediction," 2017, https://arxiv.org/abs/1702.07811.

[33] R. Mihalcea and P. Tarau, "Textrank: bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404–411, Barcelona, Spain, July 2004.

[34] Y. Peng, Q. Chen, and Z. Lu, "An empirical study of multitask learning on BERT for biomedical text mining," 2020, https://arxiv.org/abs/2005.02799.

[35] T. Huang, Q. She, and J. Zhang, "Boostingbert: integrating multi-class boosting into BERT for NLP tasks," 2020, https://arxiv.org/abs/2009.05959.

[36] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: a pretrained language model for scientific text," 2019, https://arxiv.org/abs/1903.10676.

[37] N. Zhang, S. Deng, X. Cheng et al., "Drop redundant, shrink irrelevant: selective knowledge injection for language pretraining," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI*, pp. 4007–4014, Beijing, China, August 2021.

[38] J. Lee, W. Yoon, S. Kim et al., "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[39] Z. Liu, D. Huang, K. Huang, Z. Li, and J. Zhao, "Finbert: a pretrained financial language representation model for financial text mining," in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 4513–4519, Beijing, China, January 2021.

[40] B. Srinivasa-Desikan, *Natural Language Processing and Computational Linguistics: A Practical Guide to Text Analysis with Python, Gensim, spaCy, and Keras*, Packt Publishing Ltd, Birmingham, UK, 2018.

[41] W. Kryściński, R. Paulus, C. Xiong, and R. Socher, "Improving abstraction in text summarization," 2018, https://arxiv.org/abs/1808.07913.

[42] A. Sanders, R. White, L. Severson et al., "Unmasking the conversation on masks: natural language processing for topical sentiment analysis of COVID- 19 Twitter discourse," 2020, https://www.medrxiv.org/content/10.1101/2020.08.28.20183863v2.

[43] S. U. Dan, Y. Xu, T. Yu, F. B. Siddique, E. Barezi, and P. Fung, "CAiRE-covid: A question answering and query-focused multi-document summarization system for covid-19 scholarly information management," 2020, https://arxiv.org/abs/2005.03975.

[44] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 27, pp. 3104–3112, 2014.

[45] A. See, P. J. Liu, and C. D. Manning, "Get to the point: summarization with pointer-generator networks," 2017, https://arxiv.org/abs/1704.04368.

[46] A. M. Rush, S. E. A. S. Harvard, S. Chopra, and J. Weston, "A neural attention model for sentence summarization," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, Lisbon, Portugal, September 2017.

[47] Z. Liang, J. Du, and C. Li, "Abstractive social media text summarization using selective reinforced Seq2Seq attention model," *Neurocomputing*, vol. 410, pp. 432–440, 2020.

[48] A. K. Bhardwaj, L. Garg, A. Garg, and Y. Gajpal, "E-learning during COVID-19 outbreak: cloud computing adoption in Indian public universities," *Computers, Materials & Continues*, vol. 66, no. 3, pp. 2471–2492, 2021.

[49] S. K. Narayanasamy, K. Srinivasan, Y.-C. Hu, S. K. Masilamani, and K.-Y. Huang, "A contemporary review on utilizing semantic web technologies in healthcare, virtual communities, and ontology-based information processing systems," *Electronics*, vol. 11, p. 453, 2022.

[50] X. Cai, S. Liu, L. Yang et al., "COVIDSum: a linguistically enriched SciBERT-based summarization model for COVID-19 scientific papers," *Journal of Biomedical Informatics*, vol. 127, Article ID 103999, 2022.

[51] N. K. Manaswi, "Understanding and working with Keras," in *Deep Learning with Applications Using Python*, pp. 31–43, Apress, Berkeley, CA, USA, 2018.

[52] S. K. Narayanasamy, K. Srinivasan, S. Mian Qaisar, and C.-Y. Chang, "Ontology-enabled emotional sentiment analysis on COVID-19 pandemic-related twitter streams," *Frontiers in Public Health*, vol. 9, Article ID 798905, 2021.

[53] C. Mallick, S. Das, and A. K. Das, "Evolutionary algorithm based summarization for analyzing covid-19 medical reports," *Understanding COVID-19: The Role of Computational Intelligence*, pp. 31–58, Springer, Berlin, Germany, 2022.

[54] T. Möller, A. Reina, R. Jayakumar, and M. Pietsch, "2020. COVID-QA: a question answering dataset for COVID-19," in *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020, Online, Association for Computational Linguistics*, July 2020.

[55] A. K. Das, B. Thumu, A. Sarkar, and S. Vimal, "Graph-based text summarization and its application on COVID-19 twitter data," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 30, pp. 513–540, 2022.

[56] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Advances in Neural Information Processing Systems*, pp. 5998–6008, Curran Associates, Inc, New York, NY, USA, 2017.

[57] R. Nallapati, B. Zhou, C. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to- sequence rnns and beyond," 2016, https://arxiv.org/abs/1602.06023.