



A novel centroid based sentence classification approach for extractive summarization of COVID-19 news reports

Sumanta Banerjee¹ · Shyamapada Mukherjee¹ ·
Sivaji Bandyopadhyay¹

Received: 17 August 2022 / Accepted: 28 February 2023 / Published online: 24 March 2023

© The Author(s), under exclusive licence to Bharati Vidyapeeth's Institute of Computer Applications and Management 2023

Abstract A COVID-19 news covers subtopics like infections, deaths, the economy, jobs, and more. The proposed method generates a news summary based on the subtopics of a reader's interest. It extracts a centroid having the lexical pattern of the sentences on those subtopics by the frequently used words in them. The centroid is then used as a query in the vector space model (VSM) for sentence classification and extraction, producing a query focused summarization (QFS) of the documents. Three approaches, TF-IDF, word vector averaging, and auto-encoder are experimented to generate sentence embedding that are used in VSM. These embeddings are ranked depending on their similarities with the query embedding. A Novel approach has been introduced to find the value for the similarity parameter using a supervised technique to classify the sentences. Finally, the performance of the method has been assessed in two different ways. All the sentences of the dataset are considered together in the first assessment and in the second, each document wise group of sentences is considered separately using fivefold cross-validation. The proposed method has achieved a minimum of 0.60 to a maximum of 0.63 mean F1 scores with the three sentence encoding approaches on the test dataset.

Keywords Sentence classification · Extractive text summarization · Query focused summarization

1 Introduction

A steep increase can be found in the volume of web content over the years, including different websites, blog posts, social media posts, and news publications flooding the Internet with textual materials. The Information Retrieval (IR) and Information Extraction (IE) techniques are useful tools to search and extract contents from the Internet depending on user's choice.

Text summarization techniques are mainly of two types, abstractive and extractive [1, 2]. The abstractive summarization presents the document's important materials in a different and condensed fashion through an intricate procedure [3]. In contrast, the extractive approach identifies and picks key portions of the text for summarization [4–6]. The extractive summarizations are mainly of two types (i) generic [7] and (ii) query focused [8]. The generic summarization is performed considering the general topics of the document. Whereas the query focused summarization (QFS) [9] considers the query as the main topic to extract text relevant to that. The vector space model (VSM) is an IR method that represents documents/sentences using vectors on the vector space and evaluates their proximity to decide their relationship. VSM ranks a set of documents/sentences [10] based on their proximity to a piece of contextual text called *query*. The centroid based summarization [11, 12] method clusters documents and finds a centroid representation for each class. Then the centroids are used to classify the forthcoming documents.

Different news readers may be concerned about different impacts of COVID-19 such as infection, death,

✉ Sumanta Banerjee
sumanta_rs@cse.nits.ac.in
Shyamapada Mukherjee
shyama@cse.nits.ac.in
Sivaji Bandyopadhyay
sivaji.cse.ju@gmail.com

¹ Computer Science and Engineering, National Institute of Technology Silchar, Silchar, Assam 788010, India

economy, livelihood, education, etc. This work finds patterns in the words of the COVID-19 news-sentences chosen by an individual reader. Then using the pattern, it can classify the sentences from forthcoming reports to extract a summary for that individual. In this work some digitally published news articles on COVID-19 are collected from “The Hindu” daily. The sentences of those articles are labeled by three volunteers. Each of the sentences are labeled with either 1 (*positive*) or 0 (*negative*). A centroid or query is formed from the frequencies of the constituent words in the train set. Finally VSM is applied to filter sentences depending on their cosine similarity to the query on the vector space. The methodology is elaborated in section 3 and a flow diagram is shown in Fig. 1. The following are the contributions of this paper:

- Prepared a gold standard dataset by processing and labeling 4988 unique sentences from COVID-19 news for the experiments.
- Proposed a technique to get the centroid or query from the positive class sentences.
- Compared three approaches for generating suitable sentence embeddings.
- Proposed a technique to find a threshold parameter that defines the decision boundary to classify the sentences.

The organization of the rest of the paper is as follows: A review of the literature has been presented in section 2. In Sect. 3 the methodologies involving the acquisition, pre-processing, and labeling of the data are discussed. It also introduces the techniques for query formation and QFS. Experimental results are reported in Sect. 4. A discussion on obtained results is presented in Sect. 5. Finally, Sect. 6 concludes the paper.

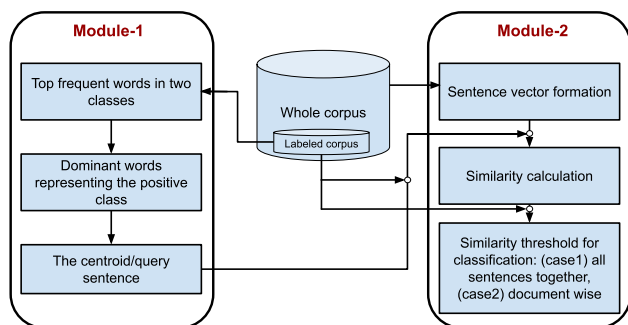


Fig. 1 Flow diagram of the work. **Module-1:** Finding the centroid/query from labeled sentences, **Module-2:** Sentence classification using the centroid

2 Related works

There are existing works where researchers have reported methods to find a succinct form of documents from various sources by extracting salient sentences [13]. The famous centroid based multi-document summarization technique introduced in [11] has been one of the promising techniques in this area. Another centroid based technique by Radev et al. [12] presents a sentence ranking and extraction based summarization. The news-summarization tool NewsInEssence [14] used the techniques in [11, 12] for document-clustering and sentence extraction to generate QFS. An interesting centroid based approach [15] used topic wise key-phrases to classify sentences and documents. The popular word2vec word-embedding technique which considers cooccurrences of words is used to summarize documents [16]. The centroid based model in [17] followed the law of universal gravitation to classify a document. A generic summarization technique is presented in [18] that ranks sentences depending on linguistic and statistical features of the words/terms. The Drag Pushing and Large Margin Drag Pushing [19, 20] provides a solution to the common inductive bias or model misfits with normalized centroid to classify sentences. A similar classification approach is presented in [21] that works on adjustments of the model. The Generalized Cluster Centroid based Classifier (GCCC) in [22] combines the powers of the K-nearest-neighbor and the Rocchio classifiers for text categorization. The Border-Instance-based Iteratively Adjusted Centroid Classifier in [23] focused on the border instances while computing the class centroids. The multiclass classifier in [24] relies on inter-class and inner-class term distributions to compute the term weights to find the classification of documents. An unsupervised extractive multi-document summarization has been presented in [25] that finds relevant sentences by comparing the sentence embeddings with document centroid. Researchers have also explored hybrid techniques where the centroid based technique is combined with Random Indexing [26] and k-means clustering [27] for multi-document summarization.

The popular sentence extraction based approaches other than centroid based approaches are discussed below. A Support Vector Regression (SVR) based model in [28] finds sentence scores depending on query features for extraction. The method introduced in [29] scores sentences with Probabilistic modeling Relevance and coverage for ranking and extraction. The authors in [8] reported a manifold-ranking based approach for QFS. A similar theme clustering and sentence ranking based approach has been introduced in [30]. The method in [31] followed a combination of different sentence embedding and selector functions for document summarization. Authors in [32] have selected salient sentences through distinguishing discriminative topics covered by each sentence. A sentence embedding based on the embedding of

POS tags, bi-grams, and tri-grams is introduced in [33]. A graph based microblog clustering algorithm for the selection of important sentences is presented in [34]. A similar graph based approach for QFS has been presented in [35]. A Bayesian topic modeling based supervised approach has been proposed in [36].

The proposed method presents a statistical approach involving the frequent words in both the positive and the negative classes to form the positive class centroid, which is unavailable in the literature. Moreover, the supervised technique designed to calculate the sentence classification threshold parameter is a new technique presented in this method.

3 Methodology

This section elaborates on the method in a step-wise fashion. The first step of the proposed approach prepares the required data for experiment. In the subsequent steps, the query is formed out of the train set of labeled sentences. Then, for each sentence, a vector is generated. In the final step, the similarities between each sentence and the query are computed which is followed by setting the threshold parameter of similarities to classify sentences.

3.1 Dataset building

Generally, for performance assessment, experiments are carried out on a standard dataset to compare the outcome with state-of-the-art. However, there are cases where a dataset suitable for a specific experiment is unavailable

and it becomes necessary to prepare a dataset for the current requirement [37]. The dataset in this work [38] is prepared by collecting 3308 documents from “The Hindu” archive from 23rd January to 24th April 2020. The title and main text of each news are combined to make a sample. During preprocessing, the URLs, punctuations (except ‘.’), digits, contracted words (like *isn't, we've, they'll*), apostrophes, and stop-words (like *I, we, you, he, they*) are removed. Finally, word stemming is performed using the porter stemmer technique available in nltk library [39]. Out of the 3308 collected documents containing 57668 sentences, the first 300 documents having 4988 unique sentences are labeled. The whole corpus and the labeled subset have 24868 and 6189 unique words respectively. The labeling is done with 1 (relevant) or 0 (irrelevant) (Table 1) for positive and negative classes respectively. The whole corpus 3308 documents are used to train the word2vec [40] model that generates word vectors for the WVA (section 3.3.2) and the auto-encoder approach (section 3.3.3).

Three research scholars (*acknowledged in the Acknowledgement section*) have volunteered to annotate the sentences as label1, label2, and label3 respectively (Table 1). They labeled each sentence with 1 and 0 if the sentence contains any of the below information and if it does not, respectively.

- Spread of infection in a geographical area.
- Death due to COVID-19 pneumonia and respiratory failure.
- A number of persons suspected to be infected with COVID-19.

Table 1 First 10 labeled sentences; *column-2* sentences, *column-3, 4 and 5* are given labels by annotators, *column-6 (final label)* is the majority vote of the labels in previous three columns

Sl No.	Sentences	Label1	Label2	Label3	Final label
1	Health counters opened at Kochi airport for prevention of Corona virus	1	1	0	1
2	Passenger surveillance has commenced in major airports across the country and in Kochi too where special health counters have been opened near the immigration desk for preventing the entry of Novel Corona Virus Disease nCoV to the country	1	1	0	1
3	Though the Thiruvananthapuram international airport does not figure on the list it has made arrangements to implement the protocol if directed	0	0	0	0
4	On Wednesday officials of the Airport Health Organisation visited the airport and briefed the airline operators and immigration officials on the standard operational procedures SOP for early isolation of those showing symptoms of the disease	0	0	0	0
5	We are ready to carry out the thermal screening of those coming from the suspected areas	0	0	0	0
6	We will activate the SOP once the directive is received Airport Director C.V. Ravindran said	0	0	0	0
7	Steps taken to prevent spread of Corona virus says Sriramulu	0	0	0	0
8	Minister for Health and Family Welfare B. Sriramulu has said that measures have been taken to prevent the spread of Corona virus from China to India	1	0	1	1
9	After his hospital stay programme in Chitradurga Mr. Sriramulu told presspersons here that teams had been deployed at all airports to screen those coming from China for Corona virus	0	0	0	0
10	He added that they are also closely monitoring Indians who are staying in China	0	0	0	0

- A number of persons infected or reinfected with COVID-19.
- A number of persons admitted to medical center due to COVID-19 symptoms.
- A number of persons suffering from pneumonia.
- Some of them are improving or cured.
- Health related updates from governments or WHO on COVID-19.

Among the 4988 sentences, 1308 and 3680 are labeled with '1' and '0' in the final label. There are sentences in the corpus that are difficult to categorize due to the priority and amount of information they carry. Hence, there are mixed opinions among the annotators for those sentences. The final label of each sentence is found by taking *majority vote* among the given three labels. The following are examples of such sentences:

1. Roadmap for research No specific treatment or vaccine against the virus exists and World Health Organization has repeatedly urged countries to share data in order to further research into the disease.
2. World Health Organization had earlier given the virus the temporary name of 2019 nCoV acute respiratory disease and China National Health Commission this week said it was temporarily calling it novel coronavirus pneumonia or NCP.
3. The virus has killed more than 1000 people infected over 42000 and reached some 25 countries with the World Health Organization declaring a global health emergency.

The first example is considered as relevant as two out of three annotators (1 and 2) labeled it as '1'. But for the second example, it is considered as irrelevant as the labels 2 and 3 are '0'. The third example has some important quantities and events in it, so it is labeled '1' by all three annotators. The first annotator labeled 1238 (label1), the second annotator labeled 1337 (label2), and the third annotator labeled 1349 (label3) sentences as '1'. The given labels in label1, label2, and label3 are the same as the final label (irrespective of '1' or '0') in 4814, 4919, and 4903 cases respectively. There are 1148 and 3516 cases where the label is unanimously given 1 and 0 respectively, and 325 cases where the opinions differed. The interrater agreement score is 0.8881 calculated using Fleiss Kappa [41] technique.

3.2 Module-1: forming the query

The frequency of a word determines its significance in an article and the importance of a sentence depends on the relative position of words with different significance in it [42].

The frequency of query terms decides the importance of a sentence in the Term Frequency (TF) technique [43], where the weightage becomes higher with a higher frequency of query terms. However, in this case, which sentences are in *positive class* and in *negative class* is already known. The task here is to collect the significant words that can represent the positive class. Since the frequently occurring words in a document can play a significant role in classifying the document [44], the plan was to find the frequently used words from both the classes in the train set. The frequent words from the positive class are collected and the frequent words common in both classes are dropped from the set of collected words. Then the collected frequent words are used to form the *query* such that the query would be close to the positive class and distant from the negative class. To restrict the length of the query within the length of the longest sentence (say *max_len*) in the dataset, only top frequent *max_len* words are picked from both classes. The 5 queries that have come from 5 train sets of the 5-step cross-validation are given below.

1. Death confirm first posit number south korea total die toll february quarantin wuhan tuesday year neg outsid read citi monday
2. Death confirm posit first number die neg itali toll total hubei wuhan year february tuesday south outsid organ read
3. death confirm posit number first die toll wuhan total hubei south tuesday neg february itali korea year provinc read monday organ
4. Death confirm first number posit die wuhan toll february total iran south korea hubei itali neg provinc organ ship outsid novel
5. Death confirm first number posit die toll total wuhan hubei south korea itali provinc neg organ ship person outsid sunday

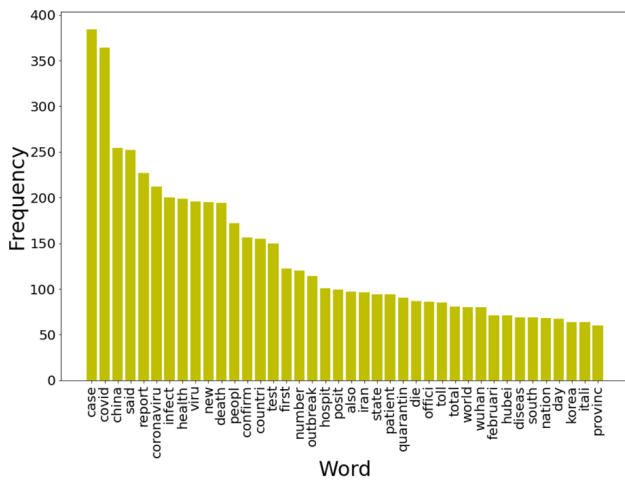
In this experiment, it is assumed that the sentences in the news with more similar words or lower semantic distance with the formed query have a higher chance of carrying significant information. The set of words in the obtained query is as per the Eq. 1.

$$query = W(P) - W(N), \quad (1)$$

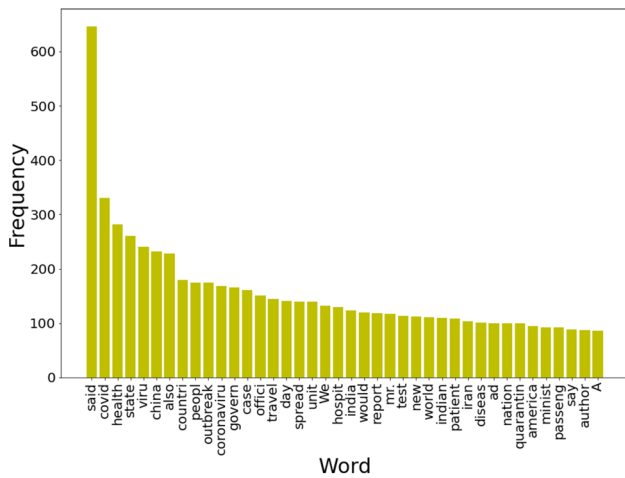
where $W(P)$ and $W(N)$ are sets of top *max_len* frequent words (Fig. 2) in positive and negative classes respectively.

3.3 Module-2: sentence classification

In this stage, each sentence is represented using a vector based on its constituting words. The uniqueness of a



(a)



(b)

Fig. 2 Most frequent words in **a** positive and **b** negative set of sentences (to avoid congestion only 40 words are shown)

sentence vector keeps it at its appropriate position on the vector space in relation to others depending on the lexical and semantic properties of its words. As per [45], it is very important to learn the representation of the data to design an efficient classifier. The approaches in [46, 47] have introduced standard deviation and relevance frequency based term weighting techniques depending on the distribution of the terms in a corpus. The most popular neural network based models to learn word embeddings are word2vec and Global Vectors (GloVe) (compared in [48]). In this work, the TF-IDF and word2vec are preferred for the generation of word embeddings.

Following are the three approaches that are investigated in this work for sentence vector generation.

3.3.1 Term frequency—inverse document frequency (TF-IDF) based approach

TF-IDF [12] is a statistical measure computed as the product of two different statistics Term Frequency (TF) and Inverse Document Frequency (IDF) [49] to calculate the weightage score of a word in a document depending on its frequency in that document and occurrence in all other documents. In this work, a sentence is considered as a document and the embedding of a sentence is generated using the TF-IDF scores of its words. In TF-IDF, the TF ranks the candidate sentences in higher the frequency of query terms, higher the rank, fashion. The normalized term frequency (*TF*) is calculated with equation 2.

$$TF = \frac{TF_{s,t}}{n}, \tag{2}$$

where $TF_{s,t}$ is the frequency of term t in a sentence s and n is the number of words in s .

In case of IDF, it gives higher weights to the rare query terms than the frequent query terms. If the number of sentences containing t is sf_t and there is N number of total sentences, then the IDF is calculated with Eq. 3.

$$IDF = \log \frac{N}{sf_t}, \tag{3}$$

and the TF-IDF vector of sentence s is calculated with Eq. 4.

$$\left[\frac{TF_{s,1}}{n} \times \log \frac{N}{sf_1}, \frac{TF_{s,2}}{n} \times \log \frac{N}{sf_2}, \dots, \frac{TF_{s,W}}{n} \times \log \frac{N}{sf_W} \right], \tag{4}$$

where W is the total number of words in the vocabulary.

3.3.2 Word vector averaging (WVA) based approach

In this approach, [50] all the word vectors of a sentence are averaged to get the sentence vector. As per Le [51] it may be unsuitable where word ordering is important. This approach is suitable in this case as it concerns only the presence of specific words in a sentence. Suppose there are n words in a sentence and each one is embedded with a vector of size m . It forms an $n \times m$ matrix, where each row represents a word. Now, the average of all the values in each of the m columns will be the sentence vector of size m . Let, a specific sentence s has n words (W_1, W_2, \dots, W_n) and the word vector $W_i \in \mathbb{R}^m$ is represented by $(x_{(i,1)}, x_{(i,2)}, \dots, x_{(i,m)})$, then the vector for the sentence s is given in Eq. 5

$$\left(\frac{1}{n} \sum_{i=1}^n x_{(i,1)}, \frac{1}{n} \sum_{i=1}^n x_{(i,2)}, \dots, \frac{1}{n} \sum_{i=1}^n x_{(i,m)} \right). \tag{5}$$

3.3.3 Auto-encoder based approach

The auto-encoder [52] is an unsupervised learning model implemented with neural networks. Its architecture has two sequential layers, i.e., Encoder and Decoder [53]. In the training phase, the input to this network is encoded to a low dimensional latent space which is reconstructed by the decoder to the output (Fig. 3). When the reconstruction error is the least the encoder part may be used to compress an input.

It is worth mentioning that the popular models for language processing, specifically sentence embedding, and semantic understanding tasks use RNN based units [54, 55] that captures not only the constituting words but also their sequence in a part of the text (document/ paragraph/ sentence), which is not the case in this problem. This problem demands inspection of the exact or similar words in a sentence for a possible match with the query.

Hence, the auto-encoder model is implemented here using Convolutional Neural Network (CNN) layers. The CNN layers along with Maxpooling layers help in the compression of the input while preserving the important features. The auto-encoder in Fig. 3 takes a sentence as input represented as $X \in \mathbb{R}^{100 \times 100}$. In the first stage, the input matrix passes through the encoder part which is having a set of layers consisting of 4 Convolutional 2D layers with *relu* as activation function, interwoven with 3 Maxpooling layers condensing the input data into 25×25 matrix. Finally, a Flatten layer followed by a Dense layer with 100 neurons and *relu* as activation function producing the compressed low dimensional *latent space* of the input sentence. In the second stage, the latent space goes into the decoder part starting with a Dense layer with *relu* as the activation function followed by a Reshape layer producing a 25×25 matrix. It is followed by 3 Convolutional 2D layers with *relu* as the activation function interwoven with 2 Up-sampling layers. Finally, ending with a Flatten and Dense (sigmoid as activation function)

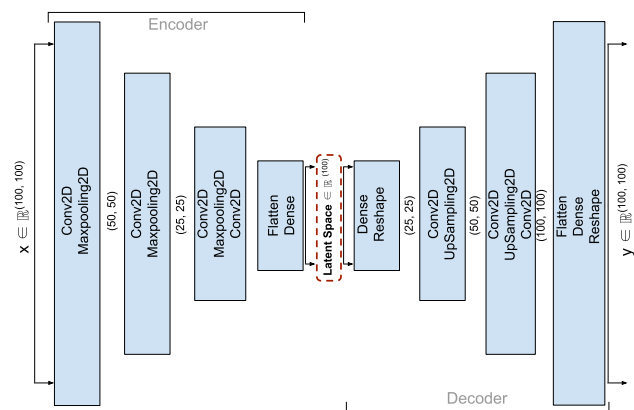


Fig. 3 Details of the auto-encoder based model

and Reshape layers producing the regenerated matrix $Y (Y \in \mathbb{R}^{100 \times 100})$ of the sentence.

3.3.4 Determining the cosine similarity threshold for classification

The key parameter defining the decision boundary for sentence classification is the threshold value of similarity between the query/centroid vector and each sentence vector. The learning of the best value of this parameter is done through a supervised way where the outcomes (*F1 score*) of a range of values for this parameter are calculated and the one with the best outcome is selected.

At first, the news documents are divided into 80% (240 documents) and 20% (60 documents) for training and testing, respectively. Then, the cosine similarity between the query and each sentence in the train set is determined. Then, the threshold value is initialized at **threshold_init**. The sentences having a similarity value equal to or above the threshold are labeled with 1 and the rest are labeled with 0. Next, the assigned labels are compared with the ground truth and the F1 score is calculated. This process is repeated by incrementing the threshold value by 0.01 each time until it reaches **threshold_limit**. Finally, the threshold value responsible for the highest F1 score is considered as the parameter to be used to classify sentences. The best threshold value determination is done for the below two cases:

case1: The true and predicted labels of all the sentences across all the train set documents are compared to calculate the F1 score.

case2: The true and predicted labels of the sentences in each document are compared to calculate the F1 scores for all 240 train set documents. Finally, the mean of the 240 F1 scores is considered to determine the best threshold value.

The design of this sentence classifier may be considered unique in its category. The identifiers used in Algorithm 1 which is followed to calculate case1 and 2 best thresholds are explained below.

Let, there are N sentences across all the M documents. V is the set of N sentence-vectors and $v_i \in V, 1 < i < N$. L and PL are the sets of ground truth and predicted sentence labels respectively. The algorithm takes M, N , and L as inputs. D is the set of similarity values between the query and each sentence, and $d_i \in D, (1 < i < N)$. $T_{all_vs_F1}$ and $T_{doc_vs_F1}$ are the sets of threshold and corresponding F1 score pairs for **case1** and **case2** respectively. The algorithm outputs θ_1 and θ_2 are the best thresholds for case1 and case2. The initial threshold **threshold_init** = 0.6 for WVA and auto-encoder approaches and **threshold_init** = 0.01 for TF-IDF approach. **threshold_limit** = 0.9 for WVA and auto-encoder approaches and **threshold_limit** =

0.03 for TF-IDF based approach. The function **threshold_for_max_F1()** returns the threshold for the highest F1 score from the lists $T_all_vs_F1$ and $T_doc_vs_F1$. The measures such as Precision (Eq. 6), Recall (Eq. 7), F1_score (Eq. 8) and cosine_similarity (Eq. 9) are calculated as follows:

$$Precision = \frac{TP}{TP + FP}, \tag{6}$$

$$Recall = \frac{TP}{TP + FN}, \tag{7}$$

$$F1_score = 2 \times \frac{Precision \times Recall}{Precision + Recall}, \tag{8}$$

where TP, TN, FP, and FN denote True Positive, True Negative, False Positive, and False Negative, respectively.

$$cosine_similarity(\vec{X}_1, \vec{X}_2) = \frac{\vec{X}_1 \cdot \vec{X}_2}{\sqrt{\sum x_{1i}^2} + \sqrt{\sum x_{2i}^2}}, \tag{9}$$

$x_{1i} \in \vec{X}_1, x_{2i} \in \vec{X}_2$ and $1 < i < vector\ length$.

Algorithm 1 Algorithm for determining the cosine similarity threshold value for classification

```

Input: L, M, N
Output:  $\theta_1, \theta_2$ 
while threshold  $\leq$  threshold_limit do
  if  $d_i \geq$  threshold then
     $pl_i \leftarrow 1$ 
  else
     $pl_i \leftarrow 0$   $\triangleright pl_i \in PL, 1 < i < N$ 
  end if
   $T\_all\_vs\_F1 \leftarrow F1\_score(L, PL)$ 
  for each document  $t_j$  in M do
     $T\_doc\_vs\_F1 \leftarrow F1\_score(Lt_j, PLt_j)$ 
     $\triangleright$  Set of actual labels is  $Lt_j$  and predicted labels is  $PLt_j$ 
  end for
end while
 $\theta_1 \leftarrow threshold\_for\_max\_F1(T\_all\_vs\_F1)$ 
 $\theta_2 \leftarrow threshold\_for\_max\_F1(T\_doc\_vs\_F1)$ 

```

3.4 Extractive summarization

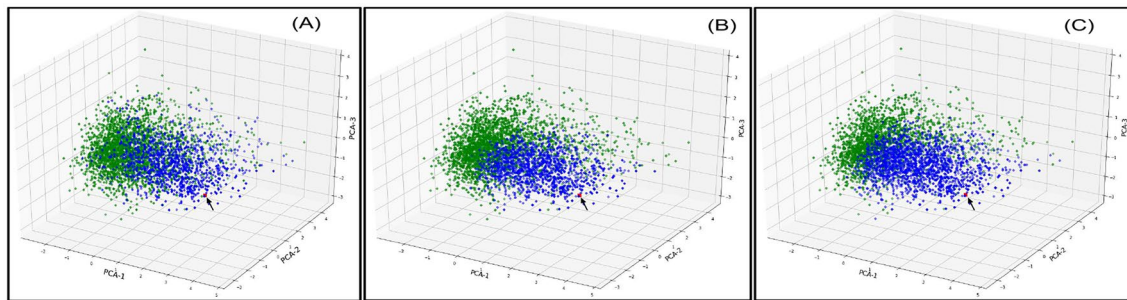
Finally, the trained threshold parameter (Sect. 3.3.4) is used to identify the important sentences in a document. Then depending on the similarity with the centroid sentences are ranked and the top 10% are selected as the summary of the document.

4 Experimental results

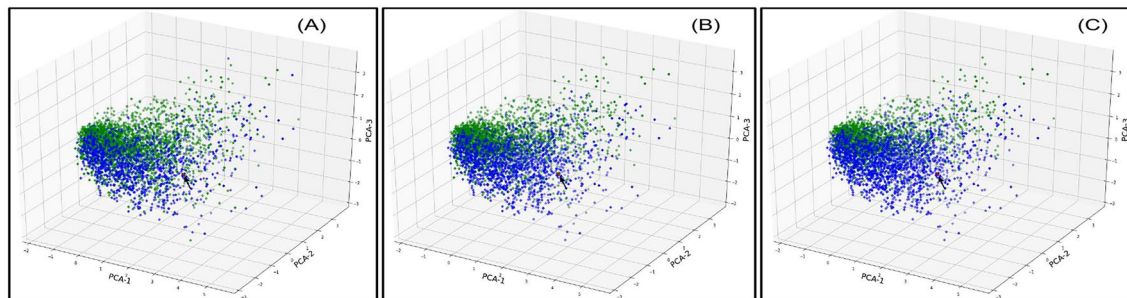
In this work, Three sentence representation approaches (Sects. 3.3.1, 3.3.2, 3.3.3) are examined, and the results are compared. Then, a threshold parameter is determined that classifies a sentence based on its cosine-similarity with the positive class centroid (Sect. 3.3.4). The cosine-similarity values for the train set sentences range from 0.0 to 0.40 for TF-IDF, from 0.11 to 0.97 for WVA, and from 0.36 to 0.92 for auto-encoder approaches (found in step5 of 5-fold cross-validation). The 3-dimensional (3-D) view of train and test set sentence vectors are shown in Figs. 4, 5, respectively. The dimensions of the vectors are reduced using Principal Component Analysis (PCA) to plot the data. The 3D plots give a good perception of the concept. It can be easily observed from the leftmost plots (marked A) of each row that the query (red square) is closer to the positively labeled sentences (blue diamonds) than the negatively labeled sentences (green diamonds). The classification can be visually perceived by comparing the figures for case1 (B) and case2 (C) in each row with the leftmost (A) figure (data with original labels). The classification boundary drawn by the similarity threshold value can also be compared for case1 and case2 for all the sentence embedding techniques. The best threshold of the cosine-similarities is obtained by comparing the F1 scores for different threshold values and picking the best one. The best threshold values that are found after this process are 0.75 (for case1) and 0.69 (for case2) for WVA, 0.65 (for case1), 0.62 (for case2) for auto-encoder approaches and 0.01 (for both cases) for TF-IDF approach. Different threshold values and corresponding F1 scores are depicted in Fig. 6A for case1 and in Fig. 6B for case2. The accuracy measure (equation 10) is also used to find the share of the correct prediction by this method. [56].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{10}$$

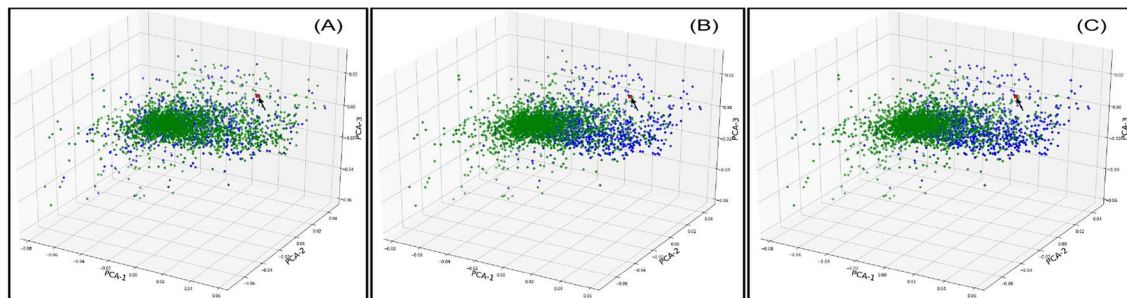
The k-fold cross-validation ($k = 5$) is used here to evaluate the sentence-embedding approaches. At first, the labeled 300 documents are divided into five folds, each having 60 documents. In each step of the five-step process, one fold is considered as the test, and the rest is considered as the train set. The outcomes of all the steps are averaged to get the final result. The sentence share in train and test sets in the first step 3826 and 1162, in the second step 4035 and 953, in the third step 4132 and 856, in the fourth step 4100 and 888, and in the fifth step 3859 and 1129. In the mean result, the WVA returns 0.80 and 0.74 accuracy scores on train and test data, respectively, and 0.64 as the F1 score for both, in case1. It returns 0.77 and 0.75 as the accuracy on train and test data, respectively, and 0.47 as the F1 score for both, in case2 (Table 2). The auto-encoder based approach



(a) Using WVA based approach



(b) Using auto-encoder based approach



(c) Using TF-IDF based approach

Fig. 4 Plotting the classified **train** sentence vectors, (Tables 2, 3, 4, step5) based on (a) actual labels, (b) predicted labels case1, and (c) predicted labels case2 (query is in red square, positive and negative

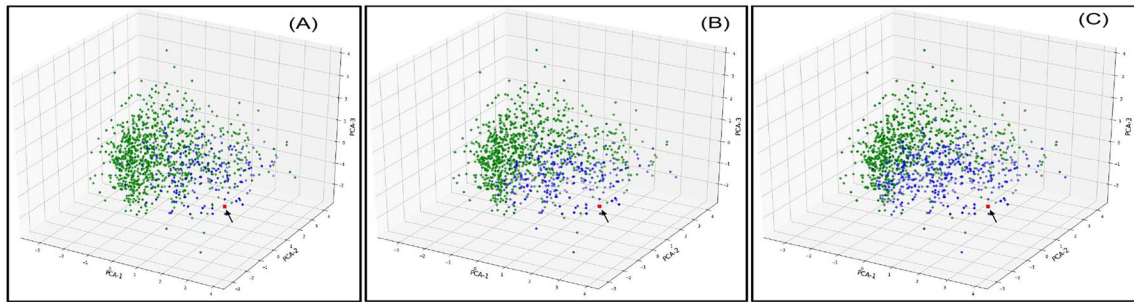
data points are in blue and green diamonds respectively); for convenience an arrow-head is used to indicate the query (Color figure online)

has shown 0.78 and 0.77 as accuracy and, 0.64 and 0.63 as F1 scores for train and test data, respectively, in case1. It has shown 0.50 and 0.51 as F1 scores and 0.74 and 0.72 as accuracy for train and test data, respectively, in case2 (Table 3). The TF-IDF based approach returns 0.62 and 0.61 as F1 scores and 0.78 as accuracy for both train and test data in case1. It has shown 0.44 and 0.46 as F1 scores, and 0.74 and 0.75 as accuracy for train and test data, respectively, in case2 (Table 4). The data that are presented in this section

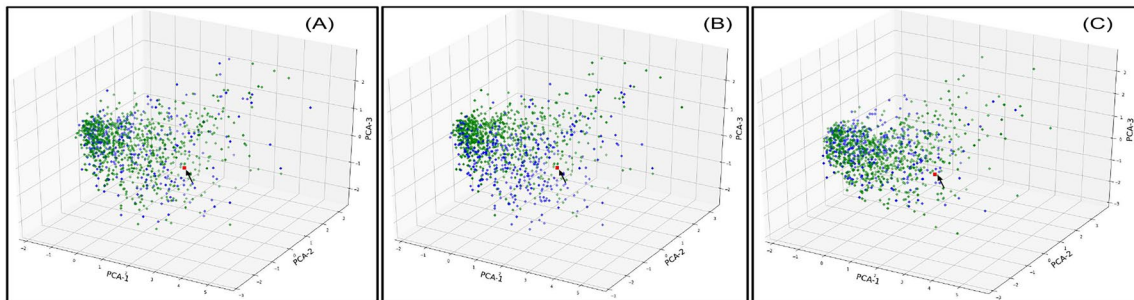
and plotted in Figs. 4, 5, 6 are from the step5 of the fivefold cross-validation process.

5 Discussion

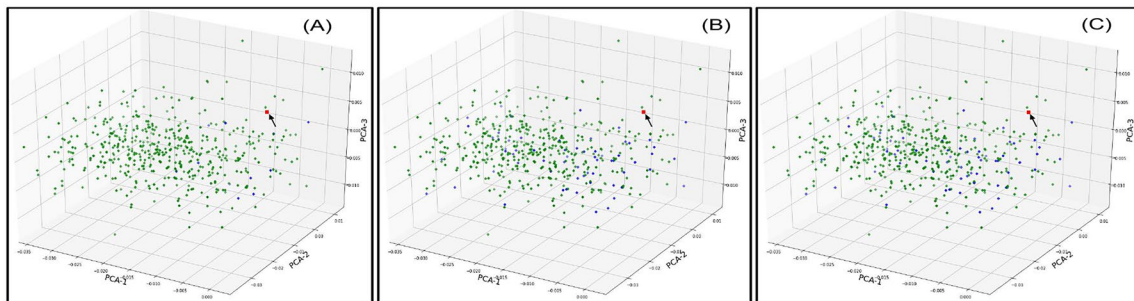
The data in this work is not highly sensitive to false-positive and false-negative predictions, so the accuracy metric (Eq. 10) gives a good performance measurement of the model. The dataset is imbalanced as it has 1308 (26%) in the positive and 3680 (74%) samples in the negative class. So,



(a) Using WVA based approach



(b) Using auto-encoder based approach



(c) Using TF-IDF based approach

Fig. 5 Plotting the classified **test** sentence vectors, (Tables 2, 3, 4, step5) based on (a) actual labels, (b) predicted labels case1, and (c) predicted labels case2 (query is in red square, positive and negative

data points are in blue and green diamonds respectively); for convenience an arrow-head is used to indicate the query (Color figure online)

the precision and recall values are very useful to evaluate the proposed method.

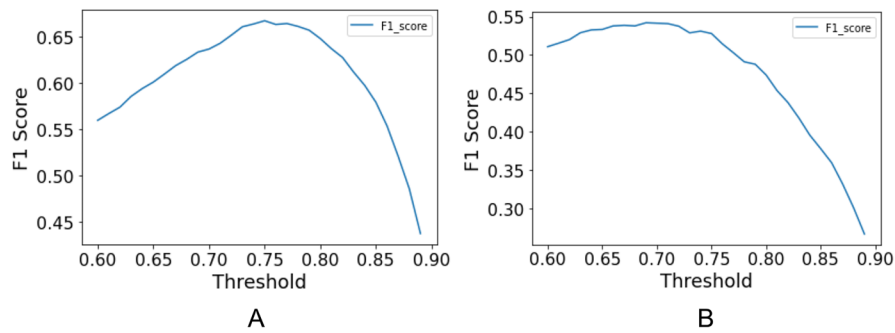
For every approach, the recall score is higher than the precision score (Tables 2, 3, and 4). That simply means that the technique correctly identifies most of the positive samples from all the positive samples in a given (test) set. This is an impressive result shown by the technique, despite being trained with a dataset that is imbalanced and contains more negative than positive samples. On the other hand, the precision score tells what share of the captured positive samples is correct. In this measure, all three approaches have

scored almost similarly well. It is observed that the results for case1 are better than case2 in all the parameters. The possible reason for that is the experiment is done for individual documents each having a different ratio of positive and negative class sentences in case2 unlike the case1 where all the sentences across all the documents are considered together.

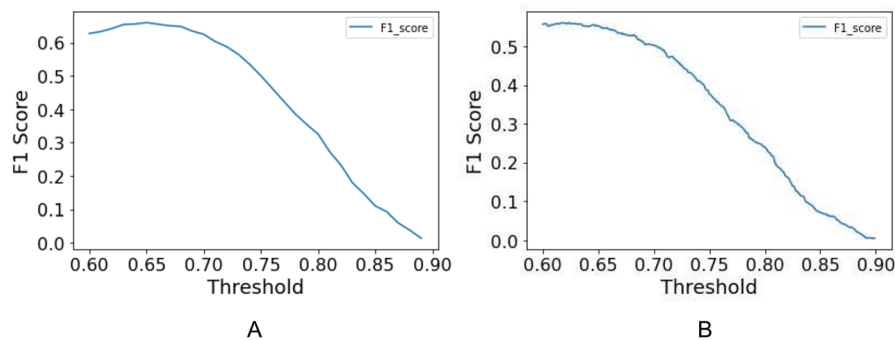
The auto-encoder approach has shown the best performance considering the recall and F1 scores in both cases. The WVA and TF-IDF approaches came second and third, respectively. The possible reasons behind the results are the following. The word2vec model utilizes word cooccurrence

to find embeddings that support the word semantics. The Convolution and Maxpooling layers in the auto-encoder capture the inter-word context among the word embeddings and condense them into a vector that is a good semantic representation of the sentence (see 3.3.3). In the WVA approach, the word embeddings in a sentence are simply averaged to get its vector representation. That makes this approach less efficient in capturing the inter-word context present in a sentence, as compared to the previous one (see 3.3.2). The TF-IDF is a statistical approach that represents a sentence with a vector of the length of the vocabulary. Each real number in a vector is calculated using the count of each word in that sentence and also in all the sentences in the set

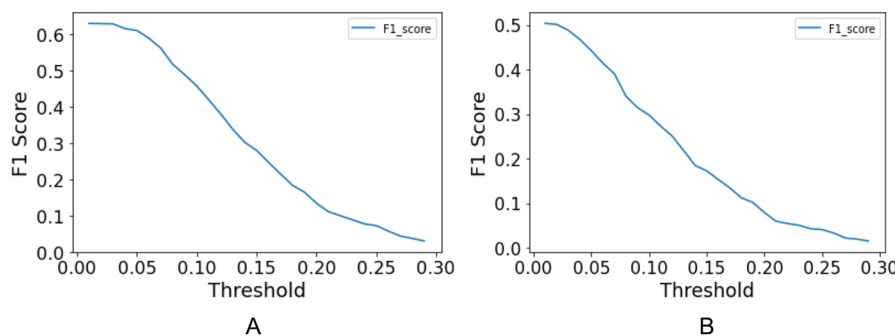
3.3.1. So, it is the least effective in putting the inter-word context of a sentence in its vector. Hence, the relationship between the centroid/query sentence and each sentence in the test set is reflected accordingly for the three sentence representation techniques, which in turn affected the determination of the similarity threshold parameter (see 3.3.4). Moreover, It is expected that the performance of the neural network based models word2vec (responsible for generating word embeddings) and the auto-encoder will improve with a dataset containing a higher amount of labeled samples. These are the possible rationale for the performance scores of three sentence representation techniques that play a significant role in efficacious classification of sentences. The



(a) Using WVA based approach



(b) Using auto-encoder based approach



(c) Using TF-IDF based approach

Fig. 6 Threshold versus F1 score plot considering (A) case1 and (B) case2, using the three approaches (Tables 2, 3, 4, step5)

Table 2 Outcome of WVA approach

	On train data				On test data			
<i>Case1</i>								
Step	F1	Accu	Prec	Rec	F1	Accu	Prec	Rec
1	0.67	0.83	0.65	0.68	0.59	0.77	0.67	0.53
2	0.63	0.81	0.61	0.66	0.67	0.77	0.63	0.71
3	0.62	0.79	0.60	0.64	0.67	0.82	0.60	0.76
4	0.62	0.79	0.57	0.68	0.67	0.80	0.63	0.73
5	0.66	0.79	0.62	0.71	0.57	0.57	0.49	0.70
Avg	0.641	0.802	0.611	0.676	0.636	0.748	0.603	0.687
<i>Case2</i>								
Step	F1	Accu	Prec	Rec	F1	Accu	Prec	Rec
1	0.51	0.77	0.46	0.63	0.52	0.71	0.51	0.62
2	0.47	0.74	0.44	0.59	0.51	0.67	0.46	0.70
3	0.44	0.77	0.46	0.48	0.41	0.78	0.41	0.50
4	0.41	0.77	0.41	0.46	0.52	0.78	0.51	0.63
5	0.52	0.79	0.54	0.58	0.39	0.79	0.35	0.46
Avg	0.469	0.766	0.463	0.546	0.471	0.748	0.447	0.582

Bold values represent the results of the 5 steps of the 5-fold cross-validation technique applied to test the performance of each of the sentence embedding approaches

Table 3 Outcome of auto-encoder based approach

	On train data				On test data			
<i>Case1</i>								
Step	F1	Accu	Prec	Rec	F1	Accu	Prec	Rec
1	0.64	0.79	0.56	0.77	0.62	0.74	0.56	0.71
2	0.64	0.81	0.59	0.70	0.62	0.76	0.63	0.61
3	0.63	0.78	0.56	0.70	0.70	0.84	0.65	0.76
4	0.63	0.77	0.53	0.77	0.65	0.75	0.54	0.82
5	0.66	0.78	0.59	0.75	0.54	0.78	0.41	0.77
Avg	0.640	0.783	0.567	0.738	0.626	0.774	0.557	0.733
<i>Case2</i>								
Step	F1	Accu	Prec	Rec	F1	Accu	Prec	Rec
1	0.51	0.76	0.45	0.67	0.56	0.71	0.50	0.68
2	0.47	0.79	0.43	0.56	0.57	0.76	0.54	0.66
3	0.49	0.68	0.42	0.68	0.51	0.76	0.44	0.70
4	0.46	0.72	0.40	0.63	0.55	0.67	0.45	0.86
5	0.56	0.74	0.50	0.74	0.36	0.70	0.30	0.53
Avg	0.499	0.736	0.441	0.656	0.508	0.722	0.447	0.685

Bold values represent the results of the 5 steps of the 5-fold cross-validation technique applied to test the performance of each of the sentence embedding approaches

centroid based sentence classification method is important where the query sentence used in QFS is generated from a set of labeled sentences.

6 Conclusion

This article presents an important contribution to the literature on centroid based extractive summarization. Each sentence in the corpus is represented using a vector on the

VSM to perform sentence classification experiments. Results of the three sentence-vector generation approaches are compared in Tables 2, 3, and 4. A supervised approach has been presented to learn the two threshold parameters for case1 and 2 to classify COVID-19 news sentences. Results show that the WVA approach exhibits the best performance in case1 and the auto-encoder approach performs well in both cases. In future work, the dataset will be extended by labeling more samples in the corpus. The state-of-the-art transformer models may be used as a sentence encoder. Features such as

Table 4 Outcome of TF-IDF based approach

	On train data				On test data			
<i>Case1</i>								
Step	F1	Accu	Prec	Rec	F1	Accu	Prec	Rec
1	0.63	0.80	0.59	0.68	0.56	0.73	0.57	0.55
2	0.61	0.80	0.59	0.63	0.65	0.77	0.64	0.66
3	0.60	0.76	0.53	0.69	0.66	0.81	0.59	0.74
4	0.61	0.77	0.55	0.70	0.63	0.77	0.58	0.69
5	0.63	0.75	0.56	0.72	0.53	0.79	0.42	0.72
Avg	0.616	0.777	0.563	0.682	0.606	0.776	0.560	0.673
<i>Case2</i>								
Step	F1	Accu	Prec	Rec	F1	Accu	Prec	Rec
1	0.46	0.76	0.44	0.57	0.49	0.71	0.48	0.54
2	0.42	0.75	0.40	0.50	0.51	0.74	0.50	0.64
3	0.43	0.73	0.41	0.53	0.42	0.79	0.40	0.49
4	0.40	0.74	0.39	0.48	0.54	0.75	0.50	0.68
5	0.50	0.75	0.49	0.60	0.36	0.78	0.33	0.46
Avg	0.443	0.747	0.424	0.533	0.462	0.754	0.443	0.561

Bold values represent the results of the 5 steps of the 5-fold cross-validation technique applied to test the performance of each of the sentence embedding approaches

sentence position, length, and key term frequency may also be explored in forming a better classification parameter.

Acknowledgements The authors would like to acknowledge the department of CSE and the IoT Laboratory, NIT Silchar for providing the research atmosphere and infrastructure necessary for this work. Thanks to the research scholars, Dr. Manosree Chatterjee (Department of Biotechnology, NIT Durgapur), Sarbani Purkayastha, and Yagnyaseene Sengupta (Department of CSE, NIT Silchar) for their help and support in labeling the data used in this research. The news reports that are used for the experiments are downloaded from "The Hindu" archive <https://www.thehindu.com/archive/>.

Data Availability The datasets generated during and/or analysed during the current study are available in the IEEE DataPort repository, at <https://dx.doi.org/10.21227/g4z2-ab91>.

References

- Mahalleh ER, Gharehchopogh FS (2022) An automatic text summarization based on valuable sentences selection. *Int J Inform Technol* 14(6):1–7
- Banerjee S, Mukherjee S, Bandyopadhyay S, Pakray P (2023) An extract-then-abstract based method to generate disaster-news headlines using a DNN extractor followed by a transformer abstractor. *Inform Proces Manag* 60(3):103291
- El-Kassas WS, Salama CR, Rafea AA, Mohamed HK (2021) Automatic text summarization: A comprehensive survey. *Expert Syst Appl* 165:113679
- Yadav AK, Singh A, Dhiman M, Kaundal R, Verma A, Yadav D, et al (2022) Extractive text summarization using deep learning approach. *Int J Inform Technol* 14(5):1–9
- Gambhir M, Gupta V (2017) Recent Automatic Text Summarization Techniques: A Survey. *Artif Intell Rev* 47(1):1–66
- Das D, Martins A (2007) A survey on automatic text summarization. Technical Report. 12
- Sahni A, Palwe S (2018) Topic Modeling on Online News Extraction. *Intell Comput Inform Commun* 673:611–622
- Cai X, Li W (2012) Mutually Reinforced Manifold-Ranking Based Relevance Propagation Model for Query-Focused Multi-Document Summarization. *IEEE Trans Audio Speech Lang Process* 20(5):1597–1607
- Baumel T, Cohen R, Elhadad M (2016) Topic Concentration in Query Focused Summarization Datasets. *Proceedings of the AAAI Conference on Artificial Intelligence*. 30:2573–2579
- Melucci M (2008) A basis for information retrieval in context. *ACM Transact Inform Syst*. 26(3):1–41
- Radev DR, Hatzivassiloglou V, Mckeown KR (1999) A Description Of The Cidr System As Used For Tdt-2. In: *In DARPA Broadcast News Workshop*
- Radev D, Jing H, Styś M, Tam D (2004) Centroid-based summarization of multiple documents. *Inf Process Manage* 11(40):919–938
- Mandal S, Singh GK, Pal A (2021) Single document text summarization technique using optimal combination of cuckoo search algorithm, sentence scoring and sentiment score. *Int J Inf Technol* 13(5):1805–1813
- Radev D, Otterbacher J, Winkel A, Blair-Goldensohn S (2005) NewsInEssence: Summarizing online news topics. *Commun ACM* 10(48):95–98
- El-Ghannam F, El-Shishtawy T (2014) Multi-Topic Multi-Document Summarizer. *International J Comput Sci Inform Technol* 5(6):77–90
- Rossello G, Basile P, Semeraro G (2017) Centroid-based Text Summarization through Compositionality of Word Embeddings. In: *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*. Valencia, Spain: Association for Computational Linguistics; p. 12–21
- Liu C, Wang W, Tu G, Xiang Y, Wang S, Lv F (2017) A new Centroid-Based Classification model for text categorization. *Knowledge-Based Syst*. 136:15–26
- J G, M K, V M, J C (1999) Summarizing Text Documents: Sentence Selection and Evaluation Metrics. In: *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference*

- on Research and Development in Information Retrieval, August 15–19, 1999, Berkeley, CA, USA; p. 121–128
19. Tan S (2008) An improved centroid classifier for text categorization. *Expert Syst Appl* 35(1–2):279–285
 20. Tan S (2007) Large margin DragPushing strategy for centroid text categorization. *Expert Syst Appl* 33(1):215–220
 21. Tan S, Wang Y, Wu G (2011) Adapting centroid classifier for document categorization. *Expert Syst Appl* 38(8):10264–10273
 22. Pang G, Jiang S (2013) A generalized cluster centroid based classifier for text categorization. *Inf Process Manage* 49(2):576–586
 23. Wang D, Wu J, Zhang H, Xu K, Lin M (2013) Towards enhancing centroid classifier for text classification-A border-instance approach. *Neurocomputing* 101:299–308
 24. Guan H, Zhou J, Guo M (2009) A Class-Feature-Centroid Classifier for Text Categorization. In: *Proceedings of the 18th International Conference on World Wide Web*. New York, NY, USA: Association for Computing Machinery; p. 201–210
 25. Lamsiyah S, El Mahdaouy A, Espinasse B, El Alaoui Ouatik S (2021) An unsupervised method for extractive multi-document summarization based on centroid approach and sentence embeddings. *Expert Syst Appl* 167:114152
 26. Sahni N, Aggarwal T, Chatterjee N (2020) Random Indexing and Centroid Based Technique for Multi Document Summarization. In: *ICDSMLA 2019*; p. 246–252
 27. Manh HC, Le Thanh H, Minh TL (2019) Extractive Multi-document Summarization using K-means, Centroid-based Method, MMR, and Sentence Position. In: *Proceedings of the Tenth International Symposium on Information and Communication Technology*; p. 29–35
 28. Ouyang Y, Li W, Li S, Lu Q (2011) Applying regression models to query-focused multi-document summarization. *Inf Process Manage* 47(2):227–237
 29. Luo W, Zhuang F, He Q, Shi Z (2013) Exploiting relevance, coverage, and novelty for query-focused multi-document summarization. *Knowl-Based Syst* 46:33–42
 30. Cai X, Li W (2013) Ranking Through Clustering: An Integrated Approach to Multi-Document Summarization. *IEEE Trans Audio Speech Lang Process* 21(7):1424–1433
 31. Templeton A, Kalita J (2018) Exploring sentence vector spaces through automatic summarization. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE; p. 55–60
 32. Gao Y, Xu Y, Huang H, Liu Q, Wei L, Liu L (2019) Jointly Learning Topics in Sentence Embedding for Document Summarization. *IEEE Trans Knowl Data Eng* 32(4):688–699
 33. Yang K, He H, Al-Sabahi K, Zhang Z (2019) EcForest: Extractive document summarization through enhanced sentence embedding and cascade forest. *Concurr Comput* 31(17):e5206
 34. Geng F, Liu Q, Zhang P (2020) A time-aware query-focused summarization on evolving microblog stream via sentence extraction. *Digital Commun Netw.* 6(3):389–397
 35. Bhaskar P, Bandyopadhyay S (2010) A Query Focused Multi Document Automatic Summarization. In: *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation*; p. 545–554
 36. Li J, Li S (2013) A novel feature-based bayesian model for query focused multi-document summarization. *Transact Assoc Comput Linguistics.* 1:89–98
 37. Bui DDA, Del Fiol G, Hurdle JF, Jonnalagadda S (2016) Extractive text summarization system to aid data extraction from full text in systematic review development. *J Biomed Inform* 64:265–272
 38. Banerjee S. Salient sentence extraction dataset from COVID-19 news reports. *IEEE Dataport*. Available from: <https://dx.doi.org/10.21227/g4z2-ab91>
 39. Loper E, Bird S (2002) NLTK: The Natural Language Toolkit. In: *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*. ETMTNLP '02. USA: Association for Computational Linguistics; p. 63–70
 40. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient Estimation of Word Representations in Vector Space. In: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2–4, 2013, Workshop Track Proceedings*
 41. Nichols TR, Wisner PM, Cripe G, Gulabchand L (2010) Putting the Kappa Statistic to Use. *Qual Assur J* 13(3–4):57–61
 42. Luhn H (1958) The Automatic Creation of Literature Abstracts. *IBM J Res Develop.* 2(2):159–165
 43. Sang-Woon Kim JMG (2019) Research paper classification systems based on TF-IDF and LDA schemes. *HCIS* 9(1):1–21
 44. Kalra V, Kashyap I, Kaur H (2022) Improving document classification using domain-specific vocabulary: hybridization of deep learning approach with TFIDF. *Int J Inform Technol* 14(5):1–7
 45. Bengio Y, Courville A, Vincent P (2013) Representation Learning: A Review and New Perspectives. *IEEE Trans Pattern Anal Mach Intell* 35(8):1798–1828
 46. Lertnattee V, Theeramunkong T (2004) Effect of term distributions on centroid-based text categorization. *Inf Sci* 158:89–115
 47. Lan M, Tan CL, Su J, Lu Y (2009) Supervised and Traditional Term Weighting Methods for Automatic Text Categorization. *IEEE Trans Pattern Anal Mach Intell* 31(4):721–735
 48. Naili M, Habacha A, Ben Ghezala H (2017) Comparative study of word embedding methods in topic segmentation. *Proc Comput Sci.* 12(112):340–349
 49. Arroyo-Fernández I, Méndez-Cruz CF, Sierra G, Torres-Moreno JM, Sidorov G (2019) Unsupervised sentence representations as word information series: Revisiting TF-IDF. *Comput Speech Lang.* 56:107–129
 50. Kenter T, Borisov A, de Rijke M (2016) Siamese CBOW: Optimizing Word Embeddings for Sentence Representations. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. vol. 1; p. 941–951
 51. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: *International conference on machine learning*. PMLR; p. 1188–1196
 52. Hinton GE, Salakhutdinov RR (2006) Reducing the Dimensionality of Data with Neural Networks. *Science* 313(5786):504–507
 53. Bowman SR, Vilnis L, Vinyals O, Dai A, Jozefowicz R, Bengio S (2016) Generating Sentences from a Continuous Space. In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics; p. 10–21
 54. Conneau A, Kiela D, Schwenk H, Barrault L, Bordes A (2017) Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9–11, 2017*; p. 670–680
 55. Arora S, Liang Y, Ma T (2017) A simple but tough-to-beat baseline for sentence embeddings. In: *International conference on learning representations; 5th International Conference on Learning Representations, ICLR 2017*
 56. Kowsari K, Jafari Meimandi K, Heidarysafa M, Mendu S, Barnes L, Brown D et al (2019) Text Classification Algorithms: A Survey. *Information.* 10(4):150

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.