Expert Systems With Applications 186 (2021) 115867

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A weighted word embedding based approach for extractive text summarization

Ruby Rani [*,1], Daya K. Lobiyal

School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi, India

ARTICLE INFO

Keywords:
Word Vectors
Diversity
Sentence Embedding
Text Clustering
Word Mover's Distance

ABSTRACT

Automatic text summarization (ATS) is a method to condense a long size text document into abridging form by enveloping all the primary information and central theme. Numerous ATS models have already prospected in this direction. However, many of those do not capture the semantic features and latent meanings of the text documents. In this paper, we present a weighted word vector representation method concerning TF-IDF for ATS. The proposed model is a prospective method for huge data on the internet that can catch all possible semantic meanings from the text along with the statistical and linguistic features. The proposed word vectors help to strengthen the diversity of the generated summary by discriminating semantically dissimilar sentences. Besides, we evaluate the proposed model on news articles taken from DUC 2007 dataset using the ROUGE summary evaluation metric. Moreover, we compare the proposed model against the four state-of-the-art summarization models and observe that our proposed approach outperforms among all the baselines and able to produce coherent, meaningful, diverse, and least redundant summaries.

1. Introduction

Millions of electronic text documents are increasing enormously over the internet. Several text analytics techniques obtain pertinent learning from these documents. Particularly, text summarization is an adequate mechanism to discover and extract relevant sentences from the input document. It is a prosperous tool for big size data problems. Besides, text summarization reduces the size of input documents without maltreating the primary purpose and content. In literature, multiple techniques have been suggested (Akter et al., 2017; AR, n.d.; Carbonell & Goldstein, 1998; Cheng & Lapata, 2016; Ganesan, Zhai, & Han, 2010; García-Hernández et al., 2008; Nallapati, Zhai, & Zhou, 2017; Nallapati, Zhou, Gulcehre, & Xiang, 2016; Neto, Freitas, & Kaestner, 2002; Nomoto & Matsumoto, 2001; Patil, Bewoor, & Patil, 2014; Rani & Lobiyal, 2020; Shivakumar & Soumya, 2015). The manual summary generation demands substantial time and human efforts. Recently, automatic text summarization (ATS) reduces the interpretation time, intervention of human efforts and remains convenient in different fashions: accommodates added knowledge inside the same expanse; solves question-answering problems; benefits against valuable article selection; exhibits honest opinions of products in commercial abstracting services; news articles, search engine results (Aggarwal, 2018), etc. Additionally, ATS generated summaries are less biased than human-generated summaries. Hence, we realize that improvement in text summarization is an influential attempt to ascertain.

There are two different approaches to create text summaries automatically: extractive and abstractive. The very first method extracts a subset of meaningful sentences as a part of a text summary from the document describing the central theme of it. They are simple to understand and easy to implement (Aggarwal, 2018) and utilize statistical features like term frequency, sentence position, title words, cue phrases, etc. to select a subset of sentences. The later approach paraphrases the original sentences using natural language generation techniques to prepare a summary, as identical to a human-like fashion. The abstractive methods are precise, rational, and articulate; their structure is more complicated and complex (Ganesan et al., 2010; Nallapati et al., 2016). In the case of the cardinality of the input documents, the ATS models are single-document, and multi-document (Saggion & Poibeau, 2013). Single document and multi-document text summarization systems construct summaries for a single designated document and multiple documents, respectively.

It is well-known that supervised machine learning and neural

* Corresponding author.
E-mail addresses: ruby73_scs@jnu.ac.in (R. Rani), lobiyal@gmail.com (D.K. Lobiyal).
[1] ORCID: 0000-0003-1257-8478.

https://doi.org/10.1016/j.eswa.2021.115867
Received 13 June 2020; Received in revised form 30 June 2021; Accepted 2 September 2021
Available online 8 September 2021

network approaches are good in generating qualitative summaries, but they required a large size of training data and significant time. Recent studies show that unsupervised methods of automatic summary generation methods are mainly concerned about statistical attributes while neglecting the semantic features and latent meaning of the text document. Thus, it is challenging to construct an ATS system that can extract semantics from text documents as well as generate an optimal summary. Recently, Mohd, Jan, and Shah (2020) have addressed such challenges in their work. In their work, they produce high-quality summaries using distributed semantic model by capturing semantics of the text units. They used Big-Vector (Jan & Khan, 2018), similar to bag-of-word approach, to measure semantic similarities between text units. Ren et al. (2019) proved that BOW approach does not consider the two main factors, that are information of word ordering and degree of each word, during sentence generation. To the best of our knowledge, it has been observed that assigning high weight to each important words gives an successful outcomes in BOW approaches. Authors in Gupta, Kanchinadam, Conathan, and Fung (2020) have shown that the weighted word embeddings compute the task-specific importance of a word unlike to BOW which computes the word importance based on its frequency. Inspired from the research work proposed by Mohd et al. (2020), we enhance the Word2Vec distributed word representation model and generate efficient weighted word vectors to capture better semantics. The novelty of our approach is to consider these two factors such as information of word ordering and degree of each word, during sentence generation in order to generate informative summary. However, deep-learning based algorithms easily and efficiently produce informative summary but require large pre-trained dataset. Currently, we are working on problems where small dataset is available. Nowadays, it is a challenging task for researchers and academician working on problems with small dataset.

In this paper, we discuss the latent semantic relationships pertained by the text document that helps to produce a semantically coherent, diverse, least redundant, compressed, and quality summary. Distributed vectors of each word capture the related words carrying the same meaning in the same context from the corpus. Therefore, we employ distributional representation vectors to perceive the hidden semantic meanings from the text. The experimental outcomes show that semantic features enhance the quality, diversity of the generated summary, and maintain the coherence of the final summary by minimizing the redundancy of irrelevant content. The contribution is summarized as follows.

- We generate weighted word vectors from the amalgamation of word vectors and the TF-IDF term weighting scheme. The proposed model utilizes the weighted word vectors to construct the sentence embedding.
- We apply the sentence embedding based K-means clustering algorithm to form the cluster of semantically closer sentences together.
- We propose a sentence ranking algorithm for each cluster that assigns prescribed weights to the sentences.
- To produce a diverse, relevant, and compressed summary, we also propose a summary generation algorithm that eliminates irrelevant sentences; selects diverse sentences by exploiting word vectors; constructs condensed summaries.
- For a diversity-rich summary, we used semantic feature-based word mover's distance (WMD) to identify diverse sentences that helps in finding dissimilarity between sentences.
- To evaluate our proposed scheme, we simulate the experiments over DUC 2007 dataset to generate a machine summaries. Besides, we employ Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score, an extractive summary evaluation toolkit that compares the machine-generated summaries against the manual summaries. The authors also investigate the effectiveness of the proposed model against the state-of-the-art summarization systems and (Mohd et al., 2020) model. From the results, we observe that the weighted semantic attributes notably improve the performance of the proposed summarizer.

Organization. The remaining sections of the paper are organized in the following way. Section 2 addresses the related work. Section 3 demonstrates the background models. In section 4, we propose a document vector based ATS model. Section 5 discusses the experimental outcomes of the proposed model and its comparison against the state-of-the-art. Section 6 concludes the proposed work.

2. Related work

In this section, we concentrate on extractive text summarization methods focusing on different features such as statistical, linguistic, and semantic. There have discussed various methods using statistical features like term frequency, cue phrases, title similarity, and sentence position. In 1950, Luhn (1958) coined the idea of ATS and extended by Edmundson and Wyllys (1961) by incorporating statistical features such as word frequency, frequency of title word in a sentence, cue phrases, and sentence position in the input document. Machine learning techniques have been widely utilized in the field of extractive text summarization by exploiting the already trained data to develop the system. In 2008, Wong, Wu, and Li (2008) employed Support Vector Machines (SVM) and Naive-Bayes approaches based on some properties such as appropriateness, content, topic, and events discussed in an input text document to generate its summary. Fattah (2014), in 2004, proposed a multi-document text summarization scheme based on a combination of statistical features including frequency of a term in the whole document, title similarity, position of the sentence, and cue-phrases. He used the three different classifiers: naive-Bayes, maximum entropy, and SVM techniques, and showed competitive results against existing approaches (Fattah, 2014). Some supervised machine learning techniques involved in extractive text summarization are discussed in (Cao, Wei, Dong, Li, & Zhou, 2015; Cheng & Lapata, 2016; Nallapati et al., 2017). The other important feature considered by researchers is the linguistic feature, such as parts of speech (POS) tags, noun, verb, adverb, adjective, and co-occurrence of words. In 1988, Mann and Thompson (1988) suggested a model that fetched the coherent text units from the main document based on Rhetorical Structure Theory (RST). In 2004, Bellare et al. (2004) have proposed a WordNet-based text summarization model for generic documents, such as news articles and biographies. They also utilized Principle Component Analysis (PCA) for sentence selection from the computed matrix. In 2011, Gupta, Pendluri, and Vats (2011) proposed a shallow linguistic feature-based model to exploit the lexical chains to improve the connectedness, cohesive, and lexicographical relations among sentences. From the above discussion, we observed that linguistic features based summaries need more processing, external resources, and are language-dependent. Numerous hybrid models integrate the features: statistical, linguistic, and semantic. In 2011, authors utilized complex networks, syntactic dependencies, and linguistic knowledge to propose an extractive text summarization model for Brazilian Portuguese. They applied different novel metrics like diversity, betweenness, closeness, and vulnerability to fetch significant sentences from the input text document. They also observed that the diversity achieved better performance compared to complex networks (Amancio, Nunes, Oliveira Jr, & Costa, 2012). One of the state-of-the-art methods for text analysis is the TextRank (Fakhrezi, Bijaksana, & Huda, 2021; Jain, Borah, & Biswas, 2020). Based on TextRank algorihtm, Paper (Ferreira et al., 2013) suggested a text summarization method, in which sentences are treated as vertices while their relations are treated as edges. The sentences with higher degrees become part of the summary formation. In 2014, Ferreira et al. (2014) addressed a graph-based clustering approach to produce a diverse and relevant summary. In this approach, they mainly focus on the co-reference resolution and discourse analysis.

A meaningful summary enriches in semantic meaningful text units

fulfills users' requirements. Recent works as we discussed earlier ensure the originality of the above statement. The second state-of-the art method, other than TextRank, is Lexical Semantic Analysis (LSA) that analyzing the text by considering relationship between the set of documents (Anjaneyulu, Sarma, Reddy, Chander, & Nagaprasad, 2019; H. Gupta & Patel, 2021). Gong and Liu (2001) employed Latent Semantic Analysis (LSA) to generate a topic rich summary. In 2004, Mihalcea and Tarau (2004) have proposed TextRank, a graph-based extractive algorithm that works on the lexical similarities present between two sentences. Several neural networks based ATS models have also been proposed those perform better than the traditional methods. In 2014, Kågebäck, Mogren, Tahmasebi, and Dubhashi (2014) proposed the first neural network-based continuous vectors for automatic text summarization. The results of these vectors were better than the existing unsupervised and supervised ATS models. In 2016, Gu, Lu, Li, and Li (2016) suggested a sequence-to-sequence learning technique named as COPYNET that segments the input text during given intervals. In 2018, a neural network model enriched with semantic relevance was used to generate semantically meaningful summaries for Chinese text documents (Ma et al., 2018). The deep learning models require large size training data to improve their efficiency and also have high time complexity (Dong, 2018). Different studies show that word embedding is an emerging technology to find the semantic relationship among words based on their vector representation efficacy. Word embeddings are the distributed representation vectors carrying semantic relations of a word for other words in a document without requiring any external resource. Additionally, these models also capture the syntactic relations without intervening any lexical and linguistic investigation. This technology is a shallow two-layer network and thus is less complex than deep learning methods and has less time complexity(Mikolov, Chen, Corrado, & Dean, 2013). It generates the pre-trained word embeddings from large training corpus in the least computational overhead. Here, we discuss recent research work done in the field of text summarization using word embedding. In 2017, Jain, Bhatia, and Thakur (2017) proposed an extractive text summarization method based on word embedding that classifies sentences as relevant and non-relevant for a summary generation. Tohalino and Amancio (2018) proposed a graph-based multi-layer extractive text summarization system for multi-documents. They considered the numerous network measurements like the degree of sentences, the strength of a sentence, shortest path, page rank, absorption time, accessibility, etc. to assign weights to the sentences in the network. They analyzed that by keeping the intra-layer and inter-layer apart have a great impact on text summarization results. Mao, Yang, Huang, Liu, and Li (2019) proposed an extractive text summarization model for single documents. They utilized supervised and unsupervised learning to weight the sentences and to find relations between sentences. They also validated that the usage of prior knowledge may also help in important sentence selection. Recently Mohd et al. (2020) employed word embedding to summarize the text documents but have limited results. Hailu, Yu, and Fantaye (2020) discussed a bi-objective framework for word embedding based ATS and their evaluation. In this work, they extract Top-n salient sentences based on word embeddings to form a reference summary and evaluate systems summaries against reference summaries. This paper extends Mohd et al. (2020) work by considering some improvements in word embedding, diversity, and redundancy.

3. Background

3.1. Distributed word vector representation

Generally, word embeddings (Word2Vec) are the methods to represent individual words by a real-valued vector in a predefined high-dimensional vector space. The word with syntactic and semantic proximity reside near in vector space (Le & Mikolov, 2014; Mikolov et al., 2013). Word2Vec framework consists of two model architectures: Continuous Bag of Words (CBOW) and skip-gram. Both model architectures predict the target variable that is again a word for the current word(s). Both frameworks describe the idea of learning of different words weights using the neural network model to produce word vectors. The CBOW predicts the probability of the current word for a given context by ignoring history and future words. The given size of context varies in its size that it may be a single word or a combination of words. In contrast, the Skip-gram predicts the context for a given word. The large size of context improves the resultant word vector quality but with an increase in computational cost (Mikolov et al., 2013) (Fig. 1).

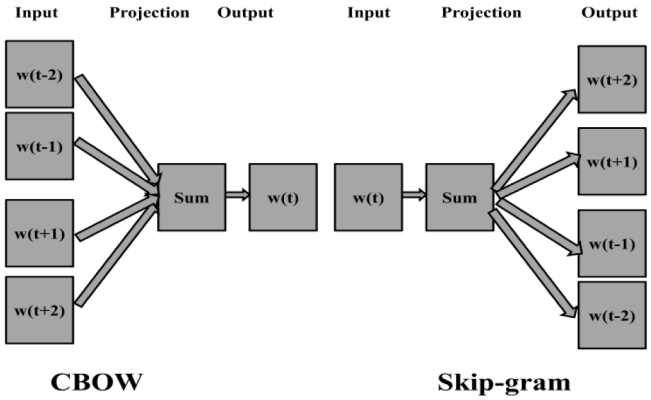

Fig. 1. Distributed Word Vector Representation Training Frameworks (Mikolov et al., 2013).

3.2. Word Mover's Distance

In 2015, Kusner, Sun, Kolkin, and Weinberger (2015) have presented a method named Word Mover's Distance (WMD) to measure the dissimilarity between two text documents. It is an extension of Earth Mover's Distance method, highly uses in transportation problems (Gottschlich & Schuhmacher, 2014), image retrieval (Rubner, Tomasi, & Guibas, 2000), and histograms (Ling & Okada, 2007). WMD learns the semantic meanings of words from local co-occurrences in sentences using Word2Vec and Glove. These embeddings extract the latent semantic relationships by performing vector operations on word vectors. For e.g., V(Berlin) - V(Germany) + V(France) = V(Paris). WMD assumes that distance between two embedded word vectors is semantically meaningful and applies this property to represent text document as a weighted point cloud of word vectors. This cloud calculates the minimum cumulative distance between word vectors from two text documents as shown in Fig. 2. Gensim yields an outstanding word mover's distance library implemented over word embeddings. The WMD is highly useful in text summarization to predict similarity among sentences in a given document for summary formation.

4. Proposed model

In this section, we discuss our proposed ATS system. Algorithm 1 illustrates the semantic news summarizer. Precisely, our proposed

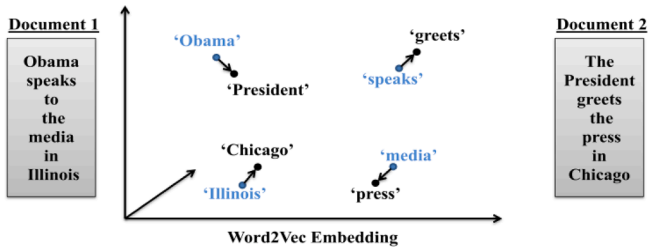

Fig. 2. An illustration for Word Mover's Distance (WMD) method where it measures the minimum aggregate length traveled by all significant words (mentioned in bold) from document 1 to document 2 for an ideal match.

model comprises of following steps as shown in Fig. 3.

Algorithm 1: Semantic News Summarizer
Input: News Document D
Output: News Summary NS.
1) begin
2) Preprocess the input news document D by applying the following functions as discussed in section 4.1.

- *Sentence Segmentation*
- *Tokenization*
- *Punctuation Removal*
- *Stopword Removal*
- *Stemming*

3) Compute distributed word vectors for each word as explained in section 4.2.
For each sentence $s_i \in S$ in Tokenized corpus do
For each word $w_j \in s_i$ do
$WV \leftarrow wv_j$
4)(a) Calculate weighted word vector using term weight of each word in the input document D as described in section 4.3.
For each word $w_j \in D$ do
Calculate TF-IDF of each word w_j
Calculate weighted word vector for each word using Eq.1.
4)(b) Calculate weighted word embedding and sentence embedding by combining word vector and term weight of each word in the input document D.
For each sentence $s_i \in S$ in Tokenized corpus do
Calculate sentence embedding for each s_i using Eq.2.
5) Calculate linguistic and statistical features weights of each sentence in the input document D as described in section 4.4.
For each sentence $s_i \in S$ in Tokenized corpus do
Call Linguistic_Features(s_i) #Calculate Linguistic Features of each sentence
Calculate Statistical_Features(s_i) # Calculate Linguistic Features of each sentence
6) Partition the sentences among semantic clusters as explained in section 4.5.
Call k_means(N, Sent_Emb) #N denotes number of clusters to create
7) Calculate the weight of each sentence in the input document D as described in section 4.6.
For each sentence $s_i \in S$ in Tokenized corpus do
Call SWRA(s_i)#SWRA is a sentence weighting algorithm
8) Compute semantic importance of each sentence for the input document D as described in section 4.7.
For each sentence $s_i \in S$ in Tokenized corpus do
Call RedR(s_i)#Calculate redundancy rate (RedR)of each sentence
Call Diversity(s_i) #Compute diversity of each sentence
9) Generate input news document D summary(NS) for the given compression ratio CR as described in section 4.7.
While CR(Summ) $< \varphi$ Repeat step 10 #φ denotes the input compression ratio
10) Select semantically rich and relevant sentences using step 8 to generate NS.
11) While $i <$ |NS| Repeat step 14:
12) Arrange final summary NS into chronological order in accordance with the original input document D.
23) END

4.1. Data preprocessing

In an automatic text summarization process, it requires to apply preprocessing steps to remove inconsistencies, noise, and irrelevant data from the input text. It comprises of the following steps:

- *Sentence Segmentation:* We utilize the NLTK sentence segmentation function to divide the input text document into the collection of sentences represented as S={s}.
- *Tokenization:* Further, we tokenize each sentence into words. We use the NLTK word tokenizer to segment two words based on space.
- *Punctuation Marks Removal*: We remove punctuation symbols such as '?', '|', '!' , ';' , '.' , ',', '-', '_','“ ”', ':-', etc. from the text document. It helps in minimizing the computational cost.
- *Stop words Removal*: The removal of uninformative words helps in dimension reduction (Rani & Lobiyal, 2018a; Rani & Lobiyal, 2018b; Rani, & Lobiyal, 2020b). E.g., for, of, in, on, his, her, etc.
- *Stemming:* In our current work, we apply the NLTK stemmer in our present work.

4.2. Distributed word vectors generation

To capture the semantic relationships exist among words, we use Word2Vec (Mikolov et al., 2013), a word embedding model to generate distributed word vectors. It is a nice and simple to use architecture to capture the semantic and syntactic similarity between texts as explained in section 3.1. We use Google News pre-trained word vectors dataset trained using the Word2Vec model. Here, we use the skip-gram architecture of the Word2Vec model to create word embeddings for the vocabulary set of the input text summarization dataset.

4.3. Weighted word vectors generation

Various research works [49]-[51] show that by incorporating the TF-IDF, a term weighting method with word representation vectors creates efficient distributed word vectors and representation learning. This makes them easy in interpretation, and captures the exact distributed semantic relations of the language. There are different ways to weigh the word vectors as addition, multiplication (Elsaadawy et al., 2018), concatenation (Mohd et al., 2020), etc. In our present work, we combine TF-IDF and Word2Vec outcomes using multiplication. To achieve sentence embedding, we multiply the TF-IDF score and word vector of each word in each sentence as shown in Eq. (1). We perform this function for all the words in a sentence. Further, the weighted embedding algorithm accumulates multiplication results of all words of a sentence and divide by the number of words in the sentence using Eq. (2). The TF-IDF helps to map the sentence to a distributed semantic vector with the difference that most frequent words have a lower impact on the result. Lastly, we obtain a vector for all sentences in the corpus and this is known as average sentence embedding.

$$WWE(w) = TF - IDF(w) * WV(w) \tag{1}$$

$$WAE(w) = \frac{\sum_{w \in s}^{n} WWE(w)}{n} \tag{2}$$

Here, $WWE(w)$ denotes the weighted word embedding of word w in a sentence. $WAE(w)$ depicts the weighted average embedding of a sentence.

4.4. Feature extraction

Here, we assign weights to all sentences in each formed cluster depending on linguistic and statistical features.

4.4.1. Linguistic features

This section discusses the linguistic features extracted using the NLTK library.

- *Noun Phrase and Verb Phrase:* In an input text document, a sentence possesses either of the noun or verb phrases are indispensable and assigned a higher weight (Kulkarni & Apte, 2002). The weight of noun and verb phrases in each sentence is calculated using Eq. (3).

$$NVP(s) = \frac{\#(\text{NounsandVerbsPhrasesins})}{|s|} \tag{3}$$

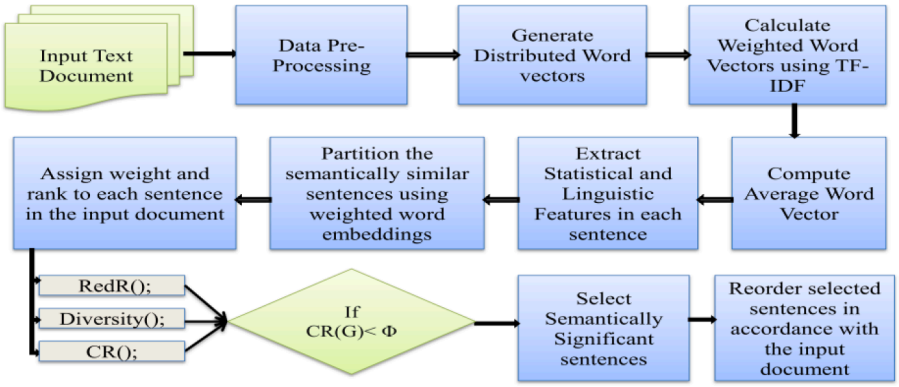

Fig. 3. Flow Diagram for Proposed Model.

Here, |s| represents the number of words in a sentence after pre-processing and #(x) represents the number of item x.

- *Proper Noun:* It refers to an entity directly and their existence in the sentence makes it more significant (Ferreira et al., 2013).
- *Cue Phrases:* These are the connectors like however, but, hence, etc. in sentences. Thus, if a cue-phrase exists at the beginning of a sentence then it is dependent on the preceding sentence and included in summary [50].

4.4.2. Statistical features

The statistical features including sentence length, sentence position, term frequency, etc. also play a crucial role in sentence selection in the proposed ATS system. The following are the statistical features considered by our system.

- *Sentence Length:* (Edmundson & Wyllys, 1961) suggested that sentences with large lengths are more significant. Consequently, our proposed model considers sentence length as an important feature for sentence weight assignment. The number of words in a sentence after pre-processing is taken as the length of a sentence and denoted by |s|.
- *Sentence Position:* Sentences exist in the beginning and at the end of the text document are more informative (McCreadie, Macdonald, & Ounis, 2018). Hence, the weight to a sentence is calculated in terms of sentence position by Eq. (4).

$$P(s) = \frac{s_j}{|S|} \tag{4}$$

Here,$P(s)$, s_j and $|S|$ represent sentence position, j^{th} sentence and length of sentence respectively.

- *Term- Frequency-Inverse-Sentence-Frequency (TF-ISF):* TF is the frequency of a word in the whole document. ISF is the number of sentences that carry the word. Precisely, the sum of TF-ISF values of all words present in a sentence post preprocessing is the weight of the sentence computed by Eq. (5) and Eq. (6).

$$TF - ISF = TF(w, D) \times ISF(w) \tag{5}$$

$$ISF(w) = \frac{|S|}{\log(SF(w)) + 1} \tag{6}$$

Here, $|S|$ is the number of sentences in text document and $SF(w)$is the number of sentences with word w.

- *Cosine Similarity:* The cosine similarity method is used to measure the similarity between two text units. Here, we apply this method to measure the cosine similarity between two sentences. The average cosine similarity (CS_{Avg}) of the sentence 's_m' is computed using the Eq. (7).

$$CS_{Avg}(s_m) = \frac{\sum_{m=1, m\neq n}^{|S|} CS(s_m, s_n)}{|S|} \tag{7}$$

where $CS(s_m, s_n)$ calculates the similarity between two sentences s_m and s_n.

4.5. Partitioning sentences using sentence embedding

In this step, we apply the K-means (Hartigan & Wong, 1979) text clustering algorithm to partition semantically closer sentences into the same cluster. The traditional K-means algorithm works on bag-of-words (BOW) but in our current work, we feed average sentence embedding to the algorithm. The main advantage of using average sentence embedding is that it captures the semantic dissimilarity between sentences. The next input parameter it takes is the number of clusters 'k' to be formed which we determine by the size of the summary. The clustering algorithm returns the clusters formed based on the semantic closeness as an output.

4.6. Sentence weighting and ranking algorithm (SWRA)

The SWRA performs the following tasks such as calculates the sentence weight for each sentence in the text document, normalizes the sentence weight using its length, sorts all sentences in decreasing order of their weights and selects the top 't' sentences from each cluster. The cumulative weight of a sentence is denoted by $Weight(s)$and calculated using Eq. (8).

$$Weight(s) = \sum_{s \in S} \frac{LinguisticfeaturesandStatisticalfeatures}{Numberoffeatures} \tag{8}$$

4.7. Summary generation Algorithm(SGA)

The SGA is the decisive algorithm for the summary generation that picks the sentences and assembles these picked sentences in the same chronological order following the original text document. The final summary should be qualitative, informative, coherent, and condensed in nature. Thus, this algorithm follows the three properties including redundancy removal, diversity, and compression ratio discussed as follows:

Redundancy Removal (RedR): Sometimes, sentences written by the author are not informative, and hence, such irrelevant sentences need to be eliminated for a summary generation. Our proposed ATS system removes redundancy by scrutinizing sentences based on its weight. Our text summarizer inspects such sentences and eliminates from inclusion in summary formation. Redundancy rate of a sentence is denoted by $RedR(s)$ and calculated by Eq. (9).

$$RedR(s) = \frac{1}{Weight(s)} \tag{9}$$

Diversity: A summary provides all possible information about the text to the reader. An informative summary covers different sentences addressing diverse topics. Therefore, the summary should include one sentence from the two sentences providing the same information but written in different forms. In present work, we utilize the Word Mover's Distance (WMD) (Gottschlich & Schuhmacher, 2014) method as described in section 3 for this purpose. The WMD method computes the semantic similarities of each sentence against all sentences in the document and creates a semantic similarity score matrix. The diversity function filters a sentence amongst all sentences based on its semantic similarity score. The diversity method ensures to discard sentences with high semantic similarity concerning the sentence weight. Consequently, the generated summary covers a variety of informative topics from the parent document. It is calculated using Eq. (10).

$$Diversity(S_m) = \frac{\sum_{n=1, S_m \neq S_n}^{|S_c|} wmd(S_m, S_n)}{|S_c|} \tag{10}$$

Here, $Diversity(S_m)$ represents the diversity of sentence S_m and $|S_c|$ is the number of sentences in cluster c.

Compression Ratio (CR): Time taken during the reading of a text document is directly proportional to the length of the text document. The main objective of the ATS system is to convert a text into the condense form. Thus, the compression ratio (CR) is one of the most important constraints in a summary generation and shrinks the size of the summary. The CR is calculated using Eq. (11).

$$CR(Summ) = \frac{|Summ|}{|D|} \tag{11}$$

Here, $CR(Summ)$ represents the compression ratio of summary, $|D|$ denotes the length of the input text document, and $|Summ|$ is the length of the summary.

5. Experimental section

In this section, we discuss the dataset used to perform the experiments, evaluate our proposed model against reference summaries and compare our model against baselines.

5.1. Dataset

In our present work, we use a benchmark dataset named the main task from Document Understanding Conference3 (DUC 2007) collected from the ACQUAINT corpus (Vorhees & Graff, 2008). National Institute of Standards and Technology (NIST) conducts a series of DUC summarization tasks since 2001 and has collected a large size text dataset. The NIST utilizes the DUC 2007 dataset for the automatic text summary evaluation task. The DUC 2007 dataset consists of news articles from various newspapers such as the Central News Agency (Taiwan), New York Times, Los Angeles Times-Washington Post, and Xinhua News Agency. The dataset comprises 45 distinct topics and 45 different documents where all 45 documents discuss all 45 topics. In our proposed work, we combine all the 45 documents of the same topic into a single document. The NIST evaluators have created four different reference summaries of approximately 250 words size for each topic and considered as the ground truth. Table 1 summarizes the characteristics of the dataset.

Table 1
Test Dataset Size Characteristics.

	DUC-2007 Dataset
Nature of Dataset	News Articles
#of Topics	45
# of Articles per Topic	45
# of Total Articles	2025
#of sentences	68,244
# of Words	560,644

5.2. Baselines

To evaluate our proposed model, we consider following state-of-the-art baselines and Mohd et al. (2020) model.

1. *Mudasir's model:* In (Mohd et al., 2020), the authors proposed a word embedding approach for text summarization to extract the semantics of the document as discussed. The distribution representation model follows the distribution hypothesis principle in which words appearing in the nearby surroundings bear the same meanings.
2. *TextRank*: In our current work, we have used an unsupervised, graph-based text summarization model named as TextRank. It is an extractive summary generation method which exploits the document structure to fetch meaningful text units. It builds a graph of the captured text units in an unsupervised manner. In TextRank, vertex denotes the sentence while the edge between two sentences denotes the established relationship between them. It utilizes the principle of voting and assigns higher votes to the vertex with more in-degrees. Hence, the vertex with higher votes gets higher rank (Mihalcea & Tarau, 2004; Erkan & Radev, 2004).
3. *LSA*: We have also utilized latent semantic analysis (LSA) as a baseline for text summarization. In LSA, the singular vector decomposition (SVD) retrieves the sentences rich in semantic information. The LSA generates a summary with a wider aspect and less redundancy (Ozsoy, Alpaslan, & Cicekli, 2011).
4. *LexRank*: The LexRank is also an unsupervised, graph-based method for a summary generation. This method utilizes the graph's centrality to select sentences. In this method, other sentences recommend a sentence followed by the selection of a sentence with a high recommendation (Erkan & Radev, 2004).

Word embedding (Mohd et al., 2020)based automatic summarization is the recently proposed model that we have chosen for the evaluation of our proposed work. Additionally, we compare our proposed model against the three state-of-the-art baselines as TextRank, LSA, and LexRank. *S*everal studies show that these baselines have been widely used for comparative analysis purposes. We used 'sumy' to implement all the three baselines and are openly available. Thus, it is easy to reproduce the same results again. Moreover, All the baseline models follow different algorithms, and thus, evaluation of our proposed model over these baselines is in-depth and meticulous.

5.3. Summary evaluation

As described in Table 1, each topic contains 45 topics, so, in our proposed work, we combine 45 documents of each topic into a single document and generate a summary for every respective topic. We have obtained summaries for respective comparison models and our proposed approach. We produced summaries of different sizes by restricting the compression ratio as 5%, 15%, 25%, and 50% of the text document for a substantial evaluation. The ROUGE-N score metric measures the effectiveness of the produced summaries discussed in this section.

ROUGE Score: ROUGE is considered as a gold standard toolkit for automatic summary evaluation. ROUGE compares the machine-generated summaries with the manual summaries (reference summaries). It evaluates summaries at a different level of granularity such as ROUGE-1, ROUGE-2, and ROUGE-L which gives results in terms of Precision (P), Recall (R), and F-score (F). ROUGE1(2) are the common unigrams or bi-grams between machine summary and reference summary; ROUGE-L is the Longest Common Sub-sequence (LCS) between the machine and the reference summaries, and ROUGE-SU applies the skip-grams and unigrams for evaluation (Lin, 2004). ROUGE score for

different granularities is calculated using Eq. (14):

$$ROUGE-N = \frac{\sum_{summ}(\sum_{c} CountCommon(c))}{\sum_{summ}(\sum_{c} Count(c))} \quad (14)$$

$where, summ \in Ref\_Summ$

Here c denotes the length of n-grams, *Ref_Summ* is the reference summary, $CountCommon(c)$ represents the number of common words between machine summary and reference summary and $Count(c)$ depicts the number of n-grams present in the reference summary. In the present work, we have generated three types of summaries, i) very short, ii) average length, and iii) lengthy. In the very short summaries, we consider two compression rates at 5% and 15%. The average length summaries and lengthy summaries take a compression rate of 25%, and 50% respectively for a summary generation. We have also computed the average metrics and macro-averaged scores for different types of summaries concerning all the given 45 topics as shown in further tables and figures.

Tables 2 and 3 give the average metrics for very short summaries at compression rates 5%, and 15% towards all articles concerning 45 topics. Experimental results show that our proposed approach performs better than the compared state-of-the-art methods. The average metrics such as Precision, Recall, and F-score score higher than the comparison models. The average Precision value of ROUGE-1 is 27%, ROUGE-2 is 12%, and Rouge-L is 27%. Moreover, we can see from Tables 2 and 3 that average Recall values over different metrics increase with the increase in summary length. At some points, our model shows lower recall, it may be because of either the shorter length of summary of the negligence of significant statistical features. With the increase in the compression rate from 5% to 15%, Macro-Average F-score values slightly decrease due to a fall in the overall precision score of different metrics. However, the Macro-Averaged F-score values at 22% and 23% compression rates are greater against the comparison models as shown in Tables 2 and 3 respectively. Hence, these results show that the presented method has a competitive efficiency against the state-of-the-art.

In the case of average length summary generation at a 25% compression rate, as shown in Table 4, the proposed approach can generate a more informative summary than comparison models. However, due to the lesser precision value, the proposed model has a lesser F-score for ROUGE-1 than LSA. In the case of the Macro-Average F-score, our model has a higher score than comparison methods and shows competitive results against LSA. From the results, we deduce that with the increase in summary size, our model captures more semantic relations and can generate a coherent summary.

Table 5 shows the experimental results for lengthy summaries generated at a 50% compression rate. From the results, we notice that the proposed model achieves better F-score against all state-of-the-art and Mohd et al. (2020) Models. However, the model (Mohd et al., 2020) shows higher recall and thus has a better F-score for ROUGE-2

Table 2
Average ROUGE Score values at Compression 5% rate.

		5%			
ROUGE-Score	Metric	Our Model	TextRank	LexRank	LSA
ROUGE-1	*F-score*	0.3	0.25	0.26	0.22
ROUGE-2		0.1	0.06	0.07	0.04
ROUGE-L		0.27	0.22	0.21	0.18
Macro-Average		0.22	0.18	0.18	0.15
ROUGE-1	*Precision*	0.27	0.29	0.27	0.21
ROUGE-2		0.12	0.08	0.07	0.05
ROUGE-L		0.27	0.24	0.23	0.18
Macro-Average		0.22	0.2	0.19	0.15
ROUGE-1	*Recall*	0.31	0.22	0.25	0.23
ROUGE-2		0.09	0.05	0.07	0.04
ROUGE-L		0.27	0.19	0.2	0.14
Macro-Average		0.22	0.15	0.17	0.14

Table 3
Average ROUGE Score values at Compression 15% rate.

		15%			
ROUGE-Score	Metric	Our Model	TextRank	LexRank	LSA
ROUGE-1	*F-score*	**0.26**	0.25	0.25	0.22
ROUGE-2		0.09	0.07	0.07	0.06
ROUGE-L		0.29	0.23	0.23	0.19
Macro-Average		0.21	0.18	0.18	0.16
ROUGE-1	*Precision*	**0.2**	0.2	0.2	0.16
ROUGE-2		0.07	0.05	0.05	0.04
ROUGE-L		0.23	0.18	0.18	0.14
Macro-Average		0.17	0.14	0.14	0.11
ROUGE-1	*Recall*	**0.39**	0.35	0.35	0.35
ROUGE-2		0.13	0.11	0.11	0.09
ROUGE-L		0.39	0.32	0.32	0.31
Macro-Average		0.3	0.26	0.26	0.25

than the proposed approach. The Macro-Average Precision, Recall, and F-score values are 0.21, 0.52, and 0.28 respectively. The Macro-Average F-score for TextRank, LexRank, LSA, and Word2vec are 0.17, 0.19, 015, and 0.18 respectively. It is noticeable from the results that the F-score increases as we increase the length of the summary. These results show that the proposed model has competitive efficiency against the state-of-the-art. These results illustrate that the hypothesis of the inclusion of semantic features has a remarkable role in enhancing the summary quality and thus holds. In all experiments, our model has consistent performance in a good summary generation over a 50% compression rate. We can conclude from the results that semantic features improved the quality of the summary generated by our proposed model. Moreover, our model achieves better Precision and F-score values than other baseline models while Recall values increase from very short length summaries to lengthy summaries. However, this score is not enough due to the removal of statistically important attributes.

Figs. 4, 5, and 6 show the F-score values at different compression rates (5%, 15%, 25%, and 50%) for ROUGE-1, ROUGE-2, and ROUGE-L evaluation metrics respectively. The figures show the experimental results performed for all documents concerning 45 different topics. These results compare the ROUGE Score metrics outcomes of the proposed approach against the state-of-the-art. It is evident from the figures that the proposed approach collects semantically important sentences and generates a coherent and meaningful summary.

Statistical Testing. Apart from experimental results, we have also performed a statistical paired *t*-test to prove that outcome of the proposed model is significant. To prove this, we have assumed the reverse hypothesis named the null hypothesis. The null hypothesis states that our proposed method did not achieve significant results against baseline methods. The paired *t*-test was conducted on 45 summaries samples in which these samples results from the proposed model and baseline algorithms paired significantly. From the statistical results, it is evident that the proposed model selected more effective semantically rich sentences than the baseline models. To affirm whether the results yield by our proposed algorithm is just a coincidence or these are statistically significant for the considered baseline methods, we performed a statistically significant test. In testing, we assumed a significance level of 5%. In other words, we exercised 30 samples with a 95% confidence level. Initially, the null hypothesis we assumed fails and an alternative hypothesis holds which, implies the outcome of our proposed model is significantly different from the baseline model. Hence, our model produces effective results. P-value, as shown in Table 6, depicts the percentage of observations of our model against the considered baselines (TextRank, LSA, LexRank, and Word2Vec). From the F-score values for ROUGE-1, ROUGE-2, and ROUGE-L, we analyze that the compression rates 0.05%-0.5% taken into account help to generate a coherent, less-redundant, diverse summary with main concentration on semantic attributes.

Table 4
Average ROUGE Score values at Compression 25% rate.

25%						
ROUGE-Score	Metric	Our Approach	TextRank	LexRank	LSA	Word2Vec
ROUGE-1	*F-score*	0.28	0.25	0.25	0.34	0.2
ROUGE-2		0.08	0.07	0.06	0.07	0.05
ROUGE-L		0.24	0.22	0.22	0.2	0.18
Macro-Average		0.2	0.18	0.18	0.2	0.14
ROUGE-1	*Precision*	0.21	0.17	0.17	0.34	0.13
ROUGE-2		0.06	0.047	0.04	0.07	0.03
ROUGE-L		0.18	0.15	0.15	0.2	0.12
Macro-Average		0.15	0.12	0.12	0.2	0.09
ROUGE-1	*Recall*	0.42	0.44	0.44	0.34	0.44
ROUGE-2		0.12	0.15	0.15	0.08	0.13
ROUGE-L		0.34	0.4	0.4	0.2	0.4
Macro-Average		0.29	0.33	0.33	0.21	0.32

Table 5
Average Rouge Score values at Compression 50% rate.

50%						
ROUGE-Score	Metric	Our Approach	TextRank	LexRank	LSA	Word2Vec
ROUGE-1	*F-score*	**0.34**	0.21	0.21	0.17	0.15
ROUGE-2		**0.22**	0.16	0.16	0.13	0.23
ROUGE-L		**0.29**	0.14	0.2	0.15	0.17
Macro-Average		0.28	0.17	0.19	0.15	0.18
ROUGE-1	*Precision*	0.27	0.13	0.13	0.1	0.11
ROUGE-2		0.13	0.1	0.11	0.08	0.14
ROUGE-L		0.22	0.12	0.12	0.09	0.11
Macro-Average		0.21	0.12	0.12	0.09	
ROUGE-1	*Recall*	0.47	0.5	0.56	0.55	0.24
ROUGE-2		0.67	0.39	0.31	0.29	0.79
ROUGE-L		0.43	0.18	0.52	0.5	0.45
Macro-Average		0.52	0.36	0.46	0.45	

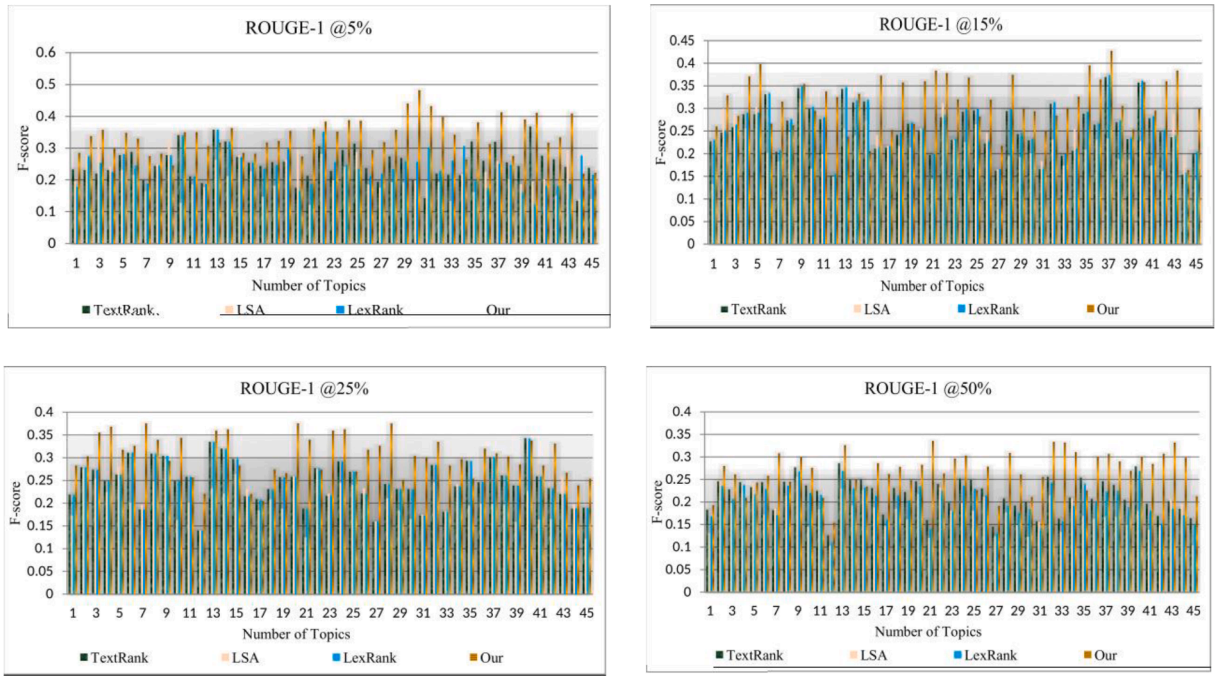

Fig. 4. F-scores metric for all given 45 topics in the DUC-2007 dataset at different compression rates for ROUGE-1.

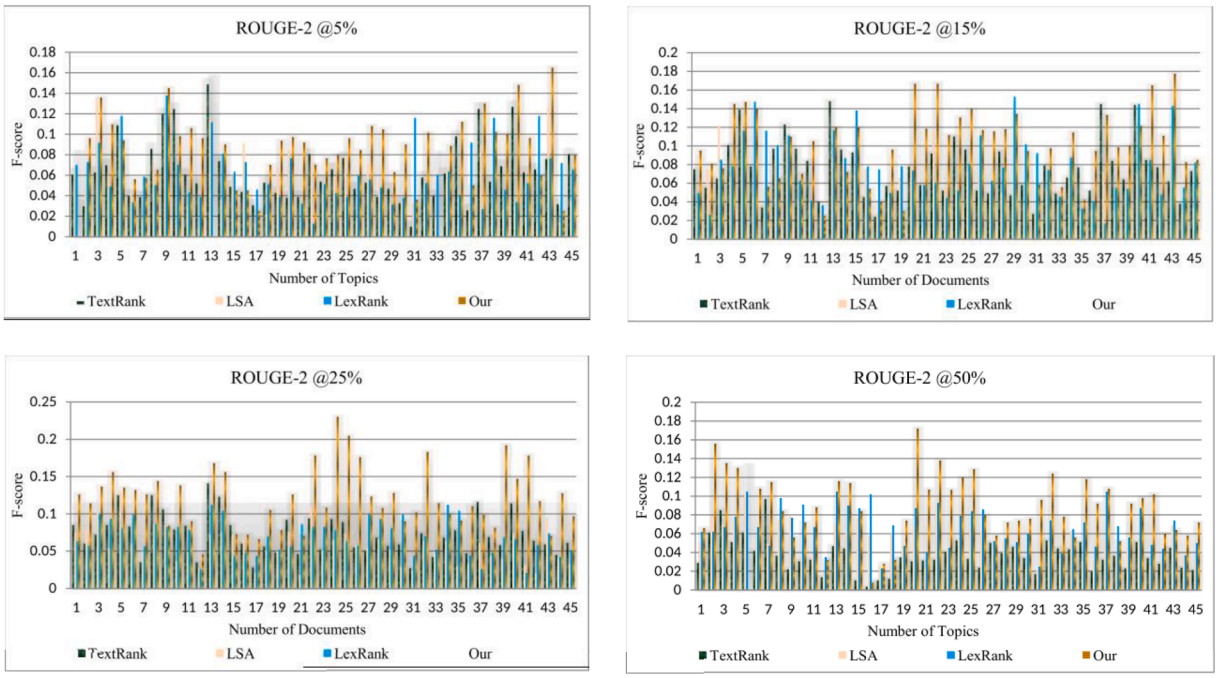

Fig. 5. F-scores metric for all given 45 topics in the DUC-2007 dataset at different compression rates for ROUGE-2.

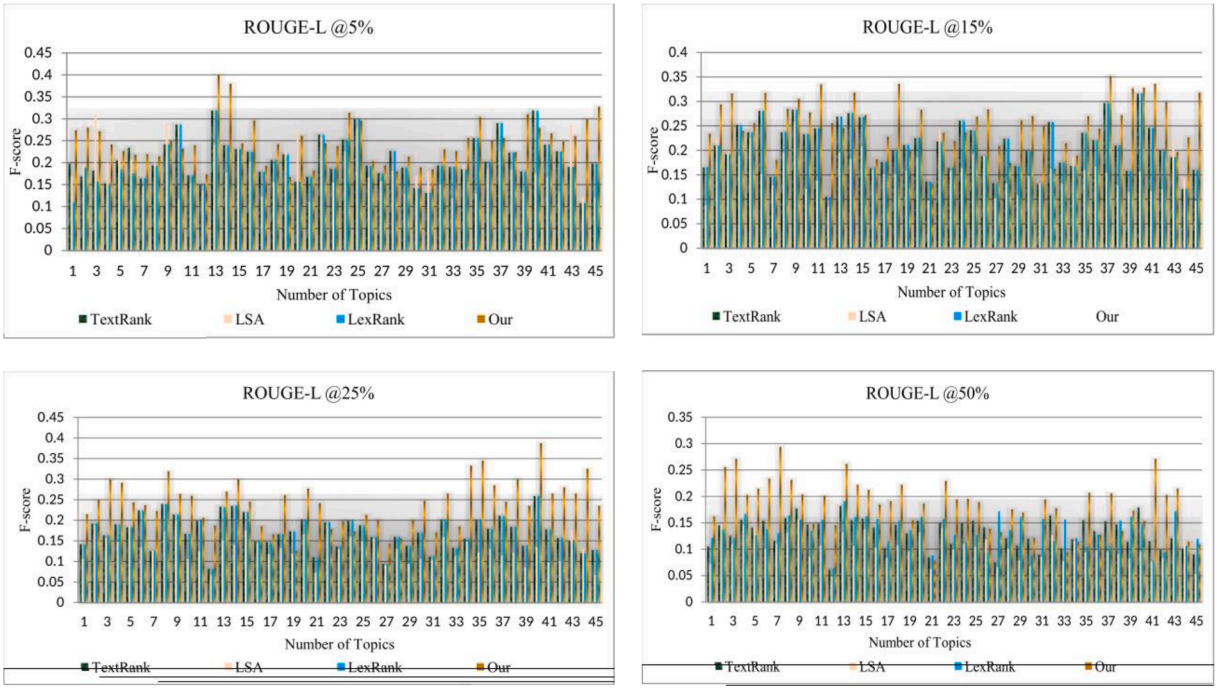

Fig. 6. F-scores metric for all given 45 topics in the DUC-2007 dataset at different compression rates for ROUGE-L.

6. Discussions and future research directions

6.1. Weighted word embedding system

Our proposed work extends the Mohd et al. work, which address the semantic and latent feature by considering the word embedding on BoW. To improve the efficiency, we proposed weighted word embedding for the same task. However, there has been chance to speed up the efficiency by twisting the word embedding mechanism, which is open problem to discuss.

6.2. Adoption of deep learning

Recently, Deep learning algorithms has shown a tremendous applications in many areas. Several Extractive text summarizer model has been discussed that uses deep learning algorithms (Al-Sabahi, Zuping, & Nadher, 2018; Dong, Shen, Crawford, van Hoof, & Cheung, 2018; Jadhav & Rajan, 2018; Nallapati et al., 2017; Narayan, Cohen, & Lapata, 2018; Wu & Hu, 2018; Zhang, Lapata, Wei, & Zhou, 2018; Zhong, Liu, Wang, Qiu, & Huang, 2019). It is well-known that deep learning is an advance approach that require large trained dataset in order to perform

Table 6
Paired *t*-test p-values for F-score under the ROUGE-Score evaluation measure for the proposed model against baseline algorithms.

P-Value under T-Test				
Metric	Our vs. TextRank	Our vs. LSA	Our vs. LexRank	Our vs. Word2Vec
ROUGE-1	0.0065713	0.000809	0.0055152	0.004372
ROUGE-2	0.0099519	0.0032617	0.0099543	0.031531
ROUGE-L	0.0367819	0.0051711	0.0324533	0.005441

the dedicated task. However, there have discussed many work that used deep learning approach to extract the semantic and latent feature of text in a given document, which are considered as the better approach but requires large trained dataset. To the best of our knowledge, deep learning algorithms gives inadequate results as compare to the word-embedding based approach for small dataset. Therefore, constructing a deep learning mechanism to extract features on small dataset is still a challenging task.

7. Conclusion

In our current work, we have presented a weighted word embedding based extractive ATS model to collect the semantic information from the text for a qualitative summary generation. Simultaneously, we have also focused on statistical and linguistic features. The sentence embedding computed by weighted word vectors helps K-means text clustering to form semantic clusters of sentences that exist in the corpus. To generate an informative summary, we have considered three main characteristics: the least redundancy, high diversity, and high compression rate. To enhance the summary diversity, we applied the semantic WMD method to help in determining the semantic closeness among sentences. From the experimental results and evaluations, we have analyzed that our model produces qualitative, diverse, least redundant, and compressed summary. From the results, we have concluded that semantic attributes play a crucial role in improving the summaries precision value and hence produced a consistently good quality summary. As stated in the evaluation and comparative analysis, our proposed model is more appropriate, reliable, and scalable.

CRediT authorship contribution statement

Ruby Rani: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Daya K. Lobiyal:** Investigation, Supervision, Project administration.

Declaration of Competing Interest

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

References

Aggarwal, C. C. (2018). Text Summarization. In Machine Learning for Text (pp. 361–380). Springer.

Akter, S., Asa, A. S., Uddin, M. P., Hossain, M. D., Roy, S. K., & Afjal, M. I. (2017). In *An extractive text summarization technique for Bengali document (s) using K-means clustering algorithm* (pp. 1–6). IEEE.

Al-Sabahi, K., Zuping, Z., & Nadher, M. (2018). A hierarchical structured self-attentive model for extractive document summarization (HSSAS). *IEEE Access, 6*, 24205–24212.

Amancio, D. R., Nunes, M. G. V., Oliveira, O. N., Jr, da Costa, L., & F.. (2012). Extractive summarization using complex networks and syntactic dependency. *Physica A: Statistical Mechanics and Its Applications, 391*(4), 1855–1864.

Anjaneyulu, M., Sarma, S., Reddy, P. V. P., Chander, K. P., & Nagaprasad, S. (2019). In *Topic Oriented Multi-document Summarization Using LSA, Syntactic and Semantic Features* (pp. 487–502). Springer.

AR, M. K. (n.d.). Text Summarization using Neural Networks and Rhetorical Structure Theory.

Bellare, K., Sarma, A. Das, Sarma, A. Das, Loiwal, N., Mehta, V., Ramakrishnan, G., & Bhattacharyya, P. (2004). Generic Text Summarization Using WordNet. In LREC.

Cao, Z., Wei, F., Dong, L., Li, S., & Zhou, M. (2015). Ranking with recursive neural networks and its application to multi-document summarization. *Twenty-ninth AAAI conference on artificial.* intelligence.

Carbonell, J., & Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 335–336).

Cheng, J., & Lapata, M. (2016). Neural summarization by extracting sentences and words. *ArXiv Preprint.* ArXiv:1603.07252.

Dong, Y. (2018). A survey on neural network-based summarization methods. *ArXiv Preprint.* ArXiv:1804.04589.

Dong, Y., Shen, Y., Crawford, E., van Hoof, H., & Cheung, J. C. K. (2018). Banditsum: Extractive summarization as a contextual bandit. *ArXiv Preprint.* ArXiv:1809.09672.

Edmundson, H. P., & Wyllys, R. E. (1961). Automatic abstracting and indexing—survey and recommendations. *Communications of the ACM, 4*(5), 226–234.

Elsaadawy, A., Torki, M., & Ei-Makky, N. (2018). In *A Text Classifier Using Weighted Average Word Embedding* (pp. 151–154). IEEE.

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

Fakhrezi, M. F., Bijaksana, M. A., & Huda, A. F. (2021). Implementation of Automatic Text Summarization with TextRank Method in the Development of Al-Qur'an Vocabulary Encyclopedia. *Procedia Computer Science, 179*, 391–398.

Fattah, M. A. (2014). A hybrid machine learning model for multi-document summarization. *Applied Intelligence, 40*(4), 592–600.

Ferreira, R., de Souza Cabral, L., Freitas, F., Lins, R. D., de França Silva, G., Simske, S. J., & Favaro, L. (2014). A multi-document summarization system based on statistics and linguistic treatment. *Expert Systems with Applications, 41*(13), 5780–5787.

Ferreira, R., de Souza Cabral, L., Lins, R. D., e Silva, G. P., Freitas, F., Cavalcanti, G. D. C., … Favaro, L. (2013). Assessing sentence scoring techniques for extractive text summarization. Expert Systems with Applications, 40(14), 5755–5764.

Ganesan, K., Zhai, C., & Han, J. (2010). Opinosis: A graph based approach to abstractive summarization of highly redundant opinions.

García-Hernández, R. A., Montiel, R., Ledeneva, Y., Rendón, E., Gelbukh, A., & Cruz, R. (2008). In *Text summarization by sentence extraction using unsupervised learning* (pp. 133–143). Springer.

Gong, Y., & Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 19–25). ACM.

Gottschlich, C., & Schuhmacher, D. (2014). The shortlist method for fast computation of the earth mover's distance and finding optimal solutions to transportation problems. *PloS One, 9*(10).

Gu, J., Lu, Z., Li, H., & Li, V. O. K. (2016). Incorporating copying mechanism in sequence-to-sequence learning. *ArXiv Preprint.* ArXiv:1603.06393.

Gupta, H., & Patel, M. (2021). In *Method Of Text Summarization Using Lsa And Sentence Based Topic Modelling With Bert* (pp. 511–517). IEEE.

Gupta, P., Pendluri, V. S., & Vats, I. (2011). In *Summarizing text by ranking text units according to shallow linguistic features* (pp. 1620–1625). IEEE.

Gupta, S., Kanchinadam, T., Conathan, D., & Fung, G. (2020). Task-optimized word embeddings for text classification representations. *Frontiers in Applied Mathematics and Statistics, 5*, 67.

Hailu, T. T., Yu, J., & Fantaye, T. G. (2020). A Framework for Word Embedding Based Automatic Text Summarization and Evaluation. *Information, 11*(2), 78.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics), 28*(1), 100–108.

Jadhav, A., & Rajan, V. (2018). Extractive summarization with swap-net: Sentences and words from alternating pointer networks. In *In Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 142–151).

Jain, A., Bhatia, D., & Thakur, M. K. (2017). In *Extractive text summarization using word vector embedding* (pp. 51–55). IEEE.

Jain, D., Borah, M. D., & Biswas, A. (2020). Fine-Tuning Textrank for Legal Document Summarization: A Bayesian Optimization Based Approach. In *In Forum for Information Retrieval Evaluation* (pp. 41–48).

Jan, R., & Khan, A. A. (2018). Emotion Mining Using Semantic Similarity. *International Journal of Synthetic Emotions (IJSE), 9*(2), 1–22.

Kågebäck, M., Mogren, O., Tahmasebi, N., & Dubhashi, D. (2014). Extractive summarization using continuous vector space models. In *In Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)* (pp. 31–39).

Kulkarni, A. R., & Apte, M. S. S. (2002). An automatic text summarization using feature terms for relevance measure. Dec.

Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. (2015). From word embeddings to document distances. In *In International conference on machine learning* (pp. 957–966).

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *In International conference on machine learning* (pp. 1188–1196).

Lin, C.-Y. (2004). *Rouge: A package for automatic evaluation of summaries*. Text Summarization Branches Out.
Ling, H., & Okada, K. (2007). An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(5), 840–853.
Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development, 2*(2), 159–165.
Ma, S., Sun, X., Li, W., Li, S., Li, W., & Ren, X. (2018). Query and output: Generating words by querying distributed word representations for paraphrase generation. *ArXiv Preprint*. ArXiv:1803.01465.
Mann, W. C., & Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse, 8*(3), 243–281.
Mao, X., Yang, H., Huang, S., Liu, Y., & Li, R. (2019). Extractive summarization using supervised and unsupervised learning. *Expert Systems with Applications, 133*, 173–181.
McCreadie, R., Macdonald, C., & Ounis, I. (2018). Automatic ground truth expansion for timeline evaluation. In *In The 41st international acm sigir conference on research & development in information retrieval* (pp. 685–694).
Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. *In Proceedings of the 2004 conference on empirical methods in natural language processing*.
Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. ArXiv Preprint ArXiv:1301.3781.
Mohd, M., Jan, R., & Shah, M. (2020). Text document summarization using word embedding. *Expert Systems with Applications, 143*, Article 112958.
Nallapati, R., Zhai, F., & Zhou, B. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *In Thirty-First AAAI Conference on Artificial Intelligence*.
Nallapati, R., Zhou, B., Gulcehre, C., & Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. *ArXiv Preprint*. ArXiv:1602.06023.
Narayan, S., Cohen, S. B., & Lapata, M. (2018). Ranking sentences for extractive summarization with reinforcement learning. *ArXiv Preprint*. ArXiv:1802.08636.
Neto, J. L., Freitas, A. A., & Kaestner, C. A. A. (2002). In *Automatic text summarization using a machine learning approach* (pp. 205–215). Springer.
Nomoto, T., & Matsumoto, Y. (2001). A new approach to unsupervised text summarization. In *In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 26–34).
Ozsoy, M. G., Alpaslan, F. N., & Cicekli, I. (2011). Text summarization using latent semantic analysis. *Journal of Information Science, 37*(4), 405–417.
Patil, M. S., Bewoor, M. S., & Patil, S. H. (2014). A hybrid approach for extractive document summarization using machine learning and clustering technique. *International Journal of Computer Science and Information Technologies, 5*(2), 1584–1586.
Rani, R., & Lobiyal, D. K. (2018a). Automatic Construction of Generic Stop Words List for Hindi Text. *Procedia Computer Science Elsevier Journal*, 1–7.
Rani, R., & Lobiyal, D. K. (2018b). In *Social Choice Theory Based Domain Specific Hindi Stop Words List Construction and Its Application in Text Mining* (pp. 123–135). Springer.
Rani, R., & Lobiyal, D. K. (2020). an extractive text summarization approach using tagged-LDA based topic modeling. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-020-09549-3
Rani, R., & Lobiyal, D. K. (2020b). Performance Evaluation of Text-Mining Models with Hindi Stopwords Lists. Journal of King Saud University-Computer and Information Sciences.
Ren, H., Zeng, Z., Cai, Y., Du, Q., Li, Q., & Xie, H. (2019). In *A weighted word embedding model for text classification* (pp. 419–434). Springer.
Rubner, Y., Tomasi, C., & Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision, 40*(2), 99–121.
Saggion, H., & Poibeau, T. (2013). Automatic text summarization: Past, present and future. In Multi-source, multilingual information extraction and summarization (pp. 3–21). Springer.
Shivakumar, K., & Soumya, R. (2015). Text summarization using clustering technique and SVM technique. *International Journal of Applied Engineering Research, 10*(12), 28873–28881.
Tohalino, J. V., & Amancio, D. R. (2018). Extractive multi-document summarization using multilayer networks. *Physica A: Statistical Mechanics and Its Applications, 503*, 526–539.
Vorhees, E., & Graff, D. (2008). AQUAINT-2 Information-retrieval text: Research collection. Linguistic Data Consortium.
Wong, K.-F., Wu, M., & Li, W. (2008). Extractive summarization using supervised and semi-supervised learning. In Proceedings of the 22nd international conference on computational linguistics (Coling 2008) (pp. 985–992).
Wu, Y., & Hu, B. (2018). Learning to extract coherent summary via deep reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32).
Zhang, X., Lapata, M., Wei, F., & Zhou, M. (2018). Neural latent extractive document summarization. ArXiv Preprint ArXiv:1808.07187.
Zhong, M., Liu, P., Wang, D., Qiu, X., & Huang, X. (2019). Searching for Effective Neural Extractive Summarization: What Works and What's Next. ArXiv Preprint ArXiv:1907.03491.