Iranian Journal of Science and Technology, Transactions of Electrical Engineering
https://doi.org/10.1007/s40998-024-00751-9

RESEARCH PAPER

Stacked Denoising Variational Auto Encoder Model for Extractive Web Text Summarization

Madhuri Yadav[1] · Rahul Katarya[1]

Received: 18 May 2023 / Accepted: 28 August 2024

Abstract
Extracting and concatenating distilled content from a corpus into a summary is a technique known as extractive summarization. In recent days, extractive summarization of web text has become popular due to the wide usage of social media. Hence various researches have been conducted on extractive summarization of web text, but the processing of huge amounts of web text and understanding the context is difficult due to the requirement of a lot of storage and time. To solve this issue, the continuous bag of words text vectorization model has been used that reduce the processing time by producing a distributed combination of words in vector arrangement. Moreover, the polysemous words are unable to be captured, which makes extraction difficult. Hence a novel Hierarchical Attention pointer Stacked Denoising Variational Autoencoder Model has been proposed in which the SDVAE model forms latent distribution for contextualized words and bidirectional attention mechanism extracts keywords and features from sentences thereby capturing polysemic words. Furthermore, the summary is obtained with dangling anaphora whereas antecedent morphological expression and verb referents are not considered in the summary. Hence a novel Multilayered Competitive Probable Modular Perception Model has been proposed in which the competitive layer scores the sentence and the scored sentences are ranked using string kernel and class conditional probability thereby considering the antecedent morphological expression and then, Graph based Quadruplicate Lexicon Summarization is used that forms quadruplicate lexicon chain in graphical format to eliminate dangling anaphoric expressions. The experimental results obtained show that the proposed model has achieved a comparatively high accuracy of 98.3% and recall, precision, and F-measure of 98%.

Keywords Continuous bag of words · Extractive summarization · Hierarchical Attention · Modular perception model · Stacked denoising variational autoencoder model

1 Introduction

Text resources have proliferated on the web as a result of the cyclopean use of digital devices combined with Internet accessibility. The most common large collections of freely accessible information are different versions of websites and online platforms. On the internet, there is a multitude of textual information. Web text encompasses a wide range of contexts/domains and is constantly updated by emerging multi-dimensional information (Abualigah et al. 2020; Hossain and Hoque 2020; El-Kassas, et al. 2021). People are not interested in reading a long piece of text and as a result, typically skip crucial sections of it. This has boosted the need for text summarization which is a method that makes key information extraction easier from a document shortly and straightforwardly.

In the current emergent information era, text summarization has evolved into a critical and relevant engine for supporting and illustrating text material. People find it much more challenging to physically summarize lengthy texts (Al-Maleh and Desouki 2020; Kumar et al. 2021; Madhuri and Ganesh Kumar 2019). Data mining is used to describe the process of dealing with large amounts of raw data and extracting meaningful information from it using

✉ Rahul Katarya
rahulkatarya@dtu.ac.in

Madhuri Yadav
madhuriyadav.me@gmail.com

[1] Big Data Analytics and Web Intelligence Laboratory, Department of Computer Science and Engineering, Delhi Technological University, New Delhi, India

Published online: 13 September 2024

an algorithm (Ghodratnama, et al. 2020; Kanapala et al. 2019). The methods that fall under the area of text summarization are extractive approaches and abstractive approaches. The extractive procedures choose a few lines or phrases from the original text, whereas the abstractive ways create a summary by building a semantic structure or semantic tree and then using the natural language generation methodology (Shirwandkar and Kulkarni 2018; Song et al. 2019).

The extractive text summarization technique is further separated into two types: supervised and unsupervised learning. Supervised learning requires a person to identify the sentences in the original training text and a learning data set while the unsupervised technique does not necessitate the need for a human for summary generation (Mao, et al. 2019; Alami et al. 2019). To construct a summary, text summarization techniques employ natural language-generating techniques. Extraction-based summarization is used by the majority of text synthesis tools. Topic identification, interpretation, summary creation, and summary evaluation are the primary challenges in text summarization. Important tasks in extraction-based summarizing include locating relevant phrases and using them to select sentences for the summary. All extraction-based summarizers do three distinct activities, (i) gathering key text components and saving them as an intermediate representation, (ii) grading text sentences based on that representation, and (iii) creating a summary by selecting several phrases.

Traditional scoring methods, such as sentence length, sentence position, and TF-IDF-based features, incorporate feature engineering as a necessary and labor-intensive task. Best n, maximal marginal relevance, and global selection are some of the methods for selecting sentences for the summary (Mishra et al. 2019; Siautama and Suhartono 2021; Mao, et al. 2010). However, these existing techniques for extractive text summarization have issues in the extraction of features due to the consideration of decontextualized sentences with the core meaning of the word only, lack coverage, and have redundancy issues (Wang, et al. 2013). Text summarization methods based on neural networks have improved substantially in recent years. Extracting semantic characteristics with neural networks for extractive summarization has gotten more attention. In the realm of natural language processing, these semantic latent characteristics are beneficial(Zhou, et al. 2020). Deep learning-based approaches (Suleiman et al. 2019; Doğan and Kaya 2019; Magdum and Rathi 2021; Shini et al. 2021) have been used to tackle the issues in traditional text summarization techniques. However, during extractive text summarization, some issues have been noticed such as the dangling anaphora problem and the inability to capture polysemy (Steinberger et al. 2007; Li et al. 2021). Hence, it is necessary to develop a novel solution to tackle the aforementioned issues. This work focuses on filling in the gaps in text summarization, with particular attention on web-specific feature extraction, web-specific data processing, sentence scoring, ranking difficulty, redundancy, and dangling anaphora resolution in web material.

1.1 Major Contribution

The following are the major contributions provided by this paper:

- In the proposed work, the CBoW text vectorization model has been introduced to eliminate the delay in processing large-scale data from websites.
- We propose a Hierarchical Attention pointer Stacked Denoising Variational Auto Encoder Model for efficient feature extraction.
- We propose a Multilayered Competitive Probable Modular Perception Model for scoring and ranking sentences.
- A Graph-based Quadruplicate Lexicon Summarization has been proposed to produce a summary by resolving the dangling anaphora issue.

Our research hypothesizes that the proposed model will eliminate processing delays, improve feature extraction, enhance sentence scoring and ranking in text summarization, and effectively address dangling anaphora issues. We aim to empirically verify these hypotheses, anticipating significant advancements in the field of text summarization.

1.2 Paper Structure

The content of the paper is structured as follows: In the next section the literature survey has been discussed, Sect. 3 discusses the process of extractive summarization, Sect. 4 discusses the proposed approach, Sect. 5 discusses experimental results and their comparison; finally, Sect. 6 concludes the paper.

2 Literature Survey

Extractive summarization is a process in which we try a sentence using a strategy and then select the most important sentences to form the output sentence. Our primary goal when summarizing a document is to produce a summary that encapsulates the document's overall conclusion. In this endeavor, extractive summarization identifies the phrases or paragraphs that accurately and precisely convey the significance of the content. Elbarougy et al. (Elbarougy et al. 2020) suggested a graph-based system in which the text is represented as a graph, with the sentences as the vertices. A modified PageRank algorithm is used for each node with a starting score equal to the number of nouns in this phrase. The cosine similarity between phrases is used to create a final summary that incorporates sentences with more information and is well related to one another. The three main stages of text summarization are pre-processing, feature extraction, and graph formation, followed by summary extraction using the Modified PageRank approach. The Modified PageRank method uses a variable number of iterations to discover the number that produces the best summary results, and the extracted summary is based on compression ratio, which takes into consideration reducing redundancy based on sentence overlapping. However, ranking score determination takes huge time and becomes expensive. El-Kassas et al. (El-Kassas, et al. 2020) presented “EdgeSumm,” revolutionary graph-based architecture based on four described algorithms. The first technique builds a new text graph model representation from the supplied content. The next two algorithms search the constructed text graph for sentences to include in the proposed summary. When the final candidate summary exceeds the user-specified threshold, the fourth strategy is utilized to select the most important sentences. EdgeSumm combines a variety of extractive ATS techniques (including graph-based, statistical-based, semantic-based, and centrality-based methods) to maximize their benefits and minimize their drawbacks. EdgeSumm is unsupervised, global (not limited to a particular topic), and does not need any training data. However, dangling anaphora problems are not focused on graph-based approaches hence additional resolution techniques are required. Patel et al. (Patel et al. 2019) provided a multi-document summary to ensure enough content coverage and information variety. The fuzzy model is used by the statistical feature-based approach to deal with the erroneous and ambiguous feature weight. In addition to this technique, redundancy elimination using cosine similarity is offered. On the DUC 2004 dataset, the proposed technique is compared to DUC participant systems and other summarizing systems such as TexLexAn, ItemSum, Yago Summarizer, MSSF, and PatSum using the ROUGE measure. However, sentence ordering is difficult using this model. Manjari et al. (Manjari, et al. 2020) employ a method that creates an extractive summary of the information depending on the user’s query, data was collected from numerous websites all over the internet. Selenium web scraping is also covered. For text summarization, the Term Frequency–Inverse Document Frequency (TF-IDF) method is used. This method is original and effective for producing summaries in response to user requests. However, other web-scraping techniques and summarization techniques are required to enhance the performance of text summarization.

Chatterjee et al. (Chatterjee and Yadav 2019) presented a text-summarizing approach that combines latent semantic analysis with random indexing. In addition, tests to compare the results to several relevant baseline approaches. A hybrid strategy based on random indexing and latent semantic analysis known as LSA-RI tries to reduce the amount of time necessary for SVD method matrix computations. The relative improvement in outcomes above the baseline LSA-based strategy demonstrates the usefulness of the hybrid method devised. However, this technique is unable to determine multiple meanings of a word and hence it is challenging to compare documents. Hernández-Castañeda et al. (Hernández-Castañeda, et al. 2020) provided a new keyword detection strategy for the ATS task that takes advantage of semantic information. This method (Uçkan and Karcı 2020) not only improves coverage by clustering sentences to identify the primary subjects in the source document, but it also improves precision by finding keywords within the clusters. The solution offered performed better than earlier methods using a common collection, according to the findings of this study’s experiments. However, the representation of the produced summary is redundant without any verb referents.

This paper (Gangathimmappa et al. 2023) proposed a DLCLS–MQO model for cross-lingual multi-document summarization. it enables queries in one language summarization to another and uses deep learning and meta-heuristics for superior results. The author (Joshi et al. 2023) proposed a deep Summ novel extractive summarization method using topic modeling and word embedding. It combines topic vectors and sequence networks to enhance summary quality and accuracy. Authors in Ma et al, (2304) proposed an impression GPT that improves radiology report generation by levering large language models (LLMs) within dynamic prompt an iterative optimization algorithm. It achieves state-of-the-art results on medical datasets, binding the gaps between LLMs and domain-specific language processing. Similarly, this paper provides an alternative approach (Luo 2303), which explores ChatGPT’s effectiveness in evaluating factual consistency in text summarization. In (Ghadimi et al. 2023) the author focused on sentence embedding and feature learning using a submodular convolution network.

Authors of Mao, et al. (2010) a maximum marginal relevance-based pre-trained BERT model is used which tackles the redundancy however dangling anaphora issue is still not focused. Authors of Elbarougy et al. (2020) take huge time for ranking score determination and in El-Kassas, et al. (2020), the dangling anaphora problem is not focused on graph-based approaches hence additional resolution techniques are required. Sentence ordering is difficult in Patel et al. (2019) and (Manjari, et al. 2020) requiring other web-scraping techniques and summarization techniques to enhance the performance of text summarization. (Chatterjee and Yadav 2019) is unable to determine multiple meanings of a word and (Hernández-Castañeda, et al. 2020) forms a summary with redundant sentences. In (Gangathimmappa et al. 2023) model focused on multi-document and multi-lingual summary generation, but did not focus on short-length text. In(Joshi et al. 2023), the author emphasized redundancy but did not address the dangling anaphoric issue. The authors of Ma et al, (2023) concentrated on domain-specific summarization using large language models but did not give attention to concerns such as redundancy and dangling anaphora. (Luo 2023) exhibits limitations such as favoring lexically similar options false reasoning, and incomplete instruction understanding. In (Ghadimi et al. 2023) the author focused on the issues of redundancy but did not address the dangling anaphora issue. Hence, there is a need to develop a novel model to solve the issues in the aforementioned techniques.

3 Summarization Methodology

Extractive summarization is the process of finding the important and most significant sentences of the text. However, the existing techniques are having issues in extracting features from text and in producing relevant summaries. The following phases are involved in the extractive summarization approach:

3.1 Preprocessing

This is a step that plays an important role in summarization. The following tasks are done that preprocess the original web text by performing tokenization, stop word removal, and lemmatization((Abualigah et al. 2020; El-Kassas, et al. 2021)). To provide an efficient processing of web text, a novel CBoW Text Vectorization Model has been introduced. This model converts the tokens formed from the Tokenization process into text vectors to form a combination of distributed representations of words which makes the processing of huge web text easier in minimum time.

3.2 Processing

3.2.1 Feature Extraction

To identify the key features of data we need to do some processing on the data (Abualigah et al. 2020; El-Kassas, et al. 2020). At the stage of extracting features from web text, a Hierarchical Attention pointer Stacked Denoising Variational Autoencoder Model has been proposed that undergoes two phases. In the first phase, SDVAE encodes word and sentences, and generate contextualized sentence representation using deep stacked layers that capture all possible word meanings based on context from preprocessed text and produce a smooth latent representation of sentences via a Denoised latent distribution layer. In the second phase, the Bidirectional Attentive Pointer Mechanism points its attention on the words from contextualized sentences in a bidirectional routine that extracts features including polysemy and thereby contextualized feature extraction is achieved.

3.2.2 Scoring and Ranking the Sentences

For scoring and ranking in summary generation, the Multilayered Competitive Probable Modular Perception Model has been proposed in which sentences are scored based on proximity and singularity of words using competitive probability. Then, scored sentences were ranked with the optimal percentage of string kernels based on conditional probabilistic rules which avoid irrelevant cataphoric expressions in ranked sentences.

3.3 Post-Processing

Additionally, Graph-based Quadruplicate Lexicon Summarization has been proposed that form quadruplicates of nouns, adjectives, determiners, and predicate lexicon chains in a graphical structure to consider verb referents thereby eliminating dangling anaphoric expressions. Hence, the proposed model increases the accuracy and precision of text summarization with consideration of polysemy, antecedent morphological expression, and verb referents.

4 The Proposed Work

The proposed model for web text summarization has been shown in Fig. 1 in which the preprocessing is enhanced by using CBoW-based vectorization and then features were extracted using a hierarchical attention pointer-based SDVAE model with bidirectional connection.

The SDVAE incorporates the variational inference principle which makes it suitable for dealing with uncertainty and task of data summarization. SDVAE has structured latent space which makes it beneficial for feature extraction. Denoising is the key component of SDVAE which helps it to capture noise. Then, the summary is generated based on the ranking and scoring process in the Multilayered Competitive Probable Modular Perception Model also, to generate the summary with verb referents and without sentence repetition, Graph based Quadruplicate Lexicon Summarization has been used.

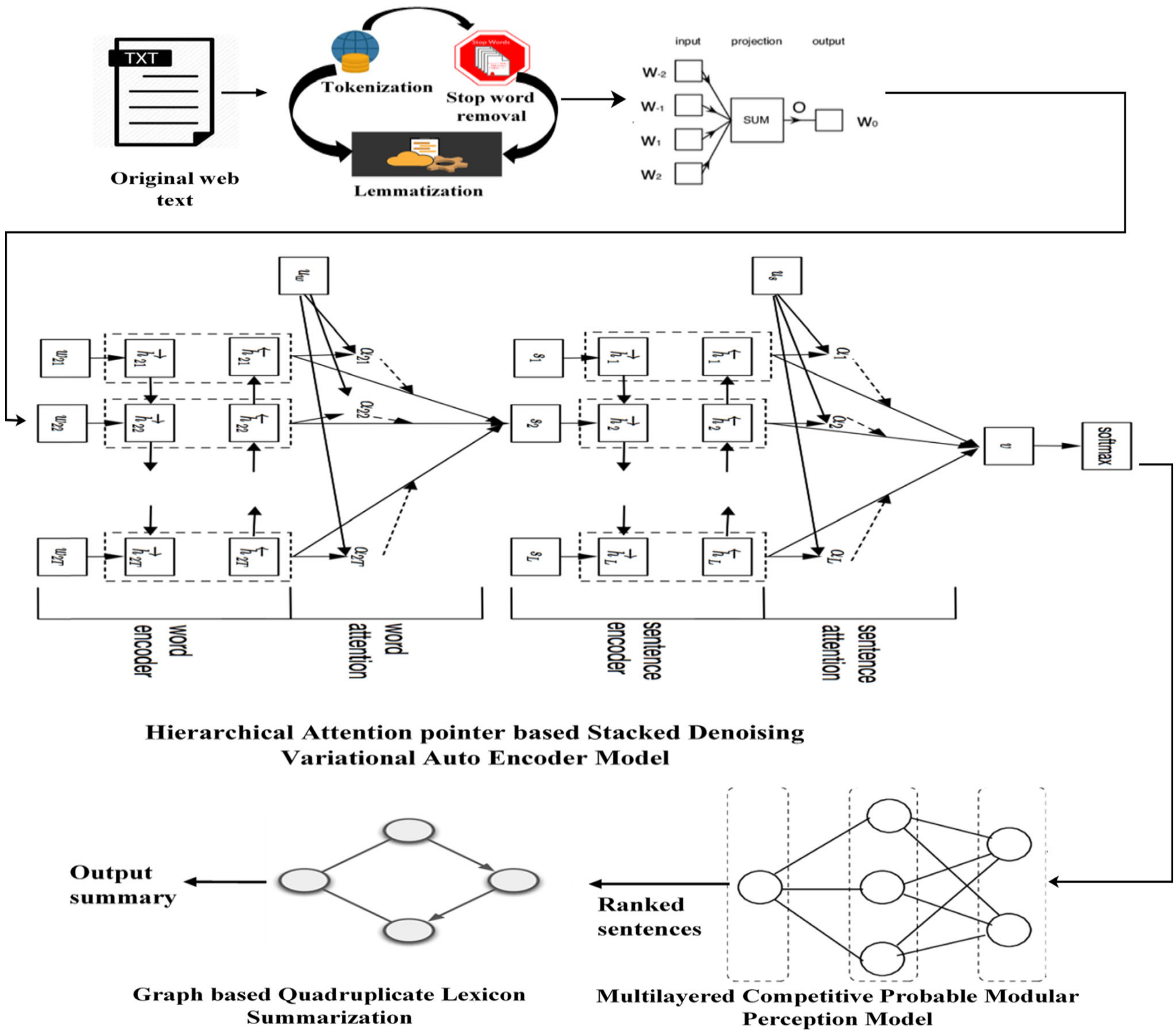


Fig. 1 Architecture of proposed web text summarization

4.1 CBoW Text Vectorization Model

Web text from Twitter has been taken as an input and this web text data is preprocessed by removing stop words from the sentences and grouping different forms of the same word using lemmatization. Then, tokenization was performed to divide a chunk of text into distinct words using a delimiter, which forms word tokens. These word tokens were given as the input to the CBoW Text Vectorization model. The process takes place on the CBoW text vectorization model which is shown in Fig. 2.

The CBOW model tries to understand the context of the words and uses that information as input. It then tries to predict the center word on the basis of the context of surrounding terms. If four context words are used to forecast one target word, the input layer will have four $1XW$ input vectors where W is the number of words in the vocabulary. The hidden layer receives these input vectors and multiplies them by a WXN matrix, where N size of the hidden layer. Finally, the $1XN$ output matrix from the hidden layer enters the sum layer, which performs an element-wise summing on the vectors before performing a final activation and obtaining the output. These preprocessed and vectorized words were given to the hierarchical model for feature extraction which is explained in the next subsection.

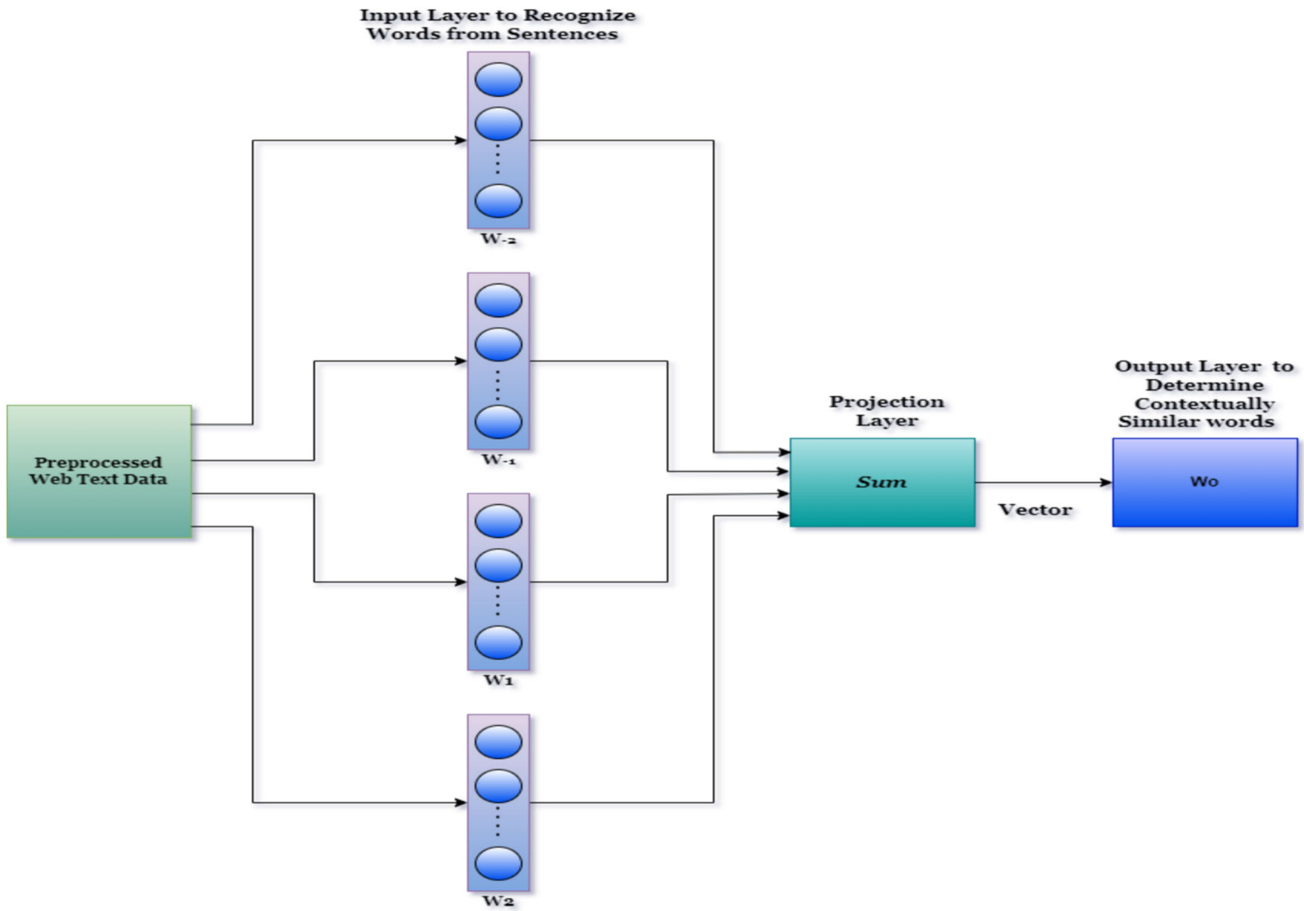


Fig. 2 CBoW text vectorization model

4.2 Hierarchical Attention Pointer Based Stacked Denoising Variational Autoencoder Model

Contextualized feature extraction with the consideration of polysemy words has been performed using the Hierarchical Attention pointer-based Stacked Denoising Variational Autoencoder (SDVAE) Model. This model undergoes a two-phase process to extract the features efficiently. In the first phase, words were encoded using SDVAE and in the second phase, an attention mechanism was utilized to extract polysemy, linguistic, and semantic features with the consideration of contextualized words. These two phases, encoding and attention pointing have been performed for sentences also. The SDVAE architecture for performing word and sentence encoding has been shown in Fig. 3.

The preprocessed vectorized web text data has been given as input to the SDVAE model. The noise in the input layer is reduced by denoising the variational auto encoder which improves the robustness and generalization ability in the extraction of more useful feature representation. Only a small amount of label noise is introduced into the original input, allowing the model to rebuild "pure" data from "polluted" input. The logarithm of the variance of the latent variables in an SDVAE parameters that is learned during the training process, and its value will vary depending on the specific input data. The purpose is to capture the uncertainty or variance in the latent space, which allows the SDVAE to generate data with varying degrees of randomness during the decoding process. The parameter settings are eliminated into enter the local optimum space. The smooth latent representation of words was obtained after performing denoising and latent state distribution in hidden layers of the encoder. Latent vectors are low-level data representations that are created by the encoder from high-level data distribution representations. After that, the decoder takes in low-level data representation and produces high-level data representation. Latent contextualized distribution relates the observable vector y to the low-dimensional latent variable x. SDVAE calculates the probability of observable vector as follows:

$$P_{\varphi}(y) = \int P_{\varphi}(y|x)P_{\mu}(x)dx \quad (1)$$

In Eq. (1), $P_{\mu}(x)$ is the prior probability of latent variables modeled by SDVAE model with parameter μ and

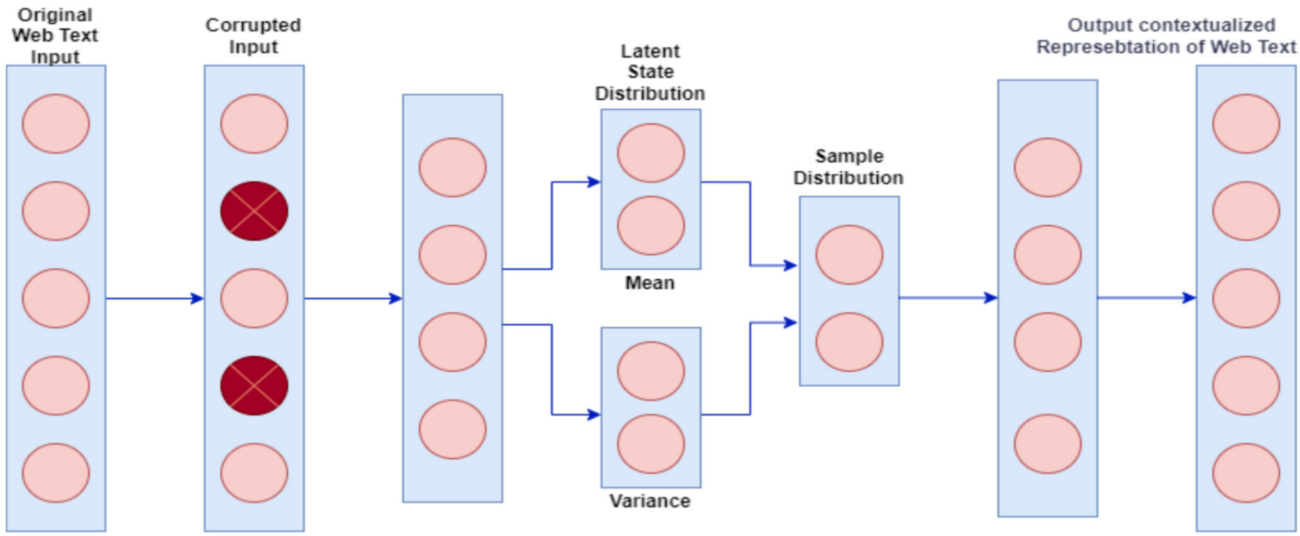


Fig. 3 Word and sentence encoding in SDVAE model

$P_{\varphi}(y|x)$ is the posterior probability of latent variables modeled by SDVAE model with parameter φ. The mean and variance of vectorized words were generated in the latent state distribution to form the contextualized distribution of words. By merging data from both directions for each word, our approach creates word annotations that incorporate contextual information. The forward and backward hidden states are concatenated to create an annotation for a specific word which encapsulates the knowledge of the entire sentence. Then, a bidirectional attentive pointer mechanism is introduced to extract keywords for the sentence's meaning and combine the representations of those informative words to generate a sentence vector. The architecture of the bidirectional attentive pointer mechanism is shown in Fig. 4.

In the attention pointer mechanism, the word annotation is first fed through a one-layer MLP network to get the word prominence vector (v_{jt}) from the hidden representation (H_{jt}) and word context vector (v_w). The word context vector v_w is thought of as an upper demonstration of a query over words, similar to what memory networks utilize. During the training procedure, v_w is initialized randomly and trained jointly. Then, the importance of the word is measured as a similarity of v_{jt} with a normalized prominence weight β_{jt} through the softmax function. The sentence vector (k_j) is then computed as a weighted sum of the word annotations depending on the prominence weights β_{jt} and hidden representation H_{jt}. Bidirectional attentive pointer mechanism for words is explained in Eqs. (2–4),

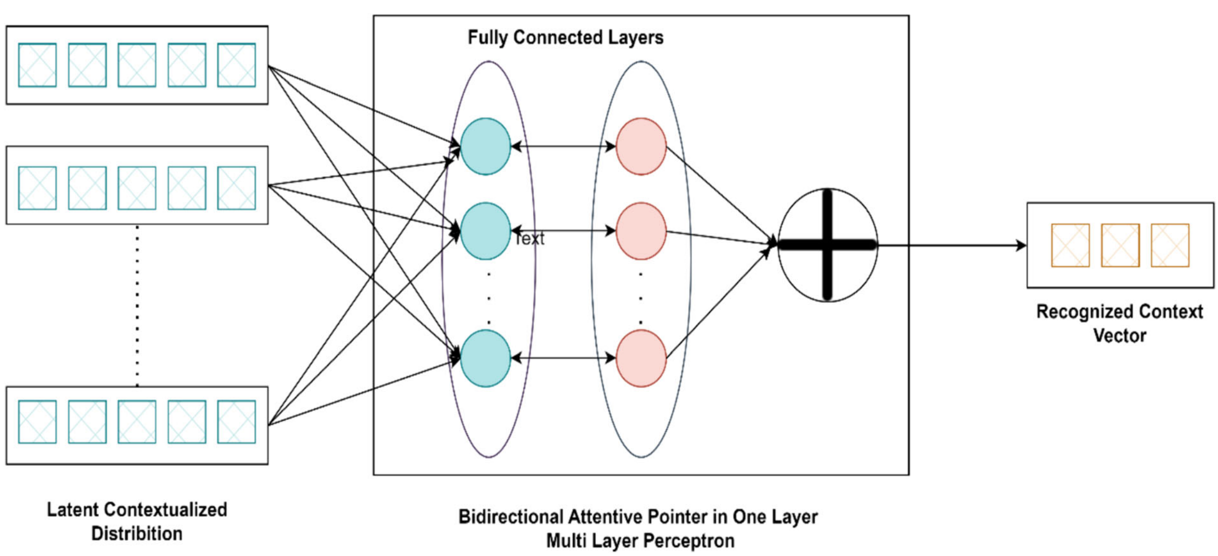


Fig. 4 Architecture of bidirectional attentive pointer mechanism

$$v_{jt} = tanh\left(W_{word}H_{jt} + v_w\right) \tag{2}$$

$$\beta_{jt} = \frac{\exp(v_{jt}^T v_w)}{\sum_t \exp(v_{jt}^T v_w)} \tag{3}$$

$$k_j = vector\ open \tag{4}$$

The sentence annotation is fed through a one-layer MLP network to get the sentence prominence vector (v_j) from the hidden representation (H_j) and sentence context vector(v_s). The sentence context vector v_s is thought of as an upper demonstration of a query over sentences. During the training procedure, v_s is initialized randomly and trained jointly. Then, the importance of the sentence is measured as a similarity of v_j with a normalized prominence weight β_j through the softmax function. The document vector (d) is then computed as a weighted sum of the sentence annotations depending on the prominence weights β_j and hidden representation H_j. Bidirectional attentive pointer mechanism for sentences is explained in Eqs. (5–7),

$$v_j = \tanh(W_{sent}H_j + v_s) \tag{5}$$

$$\beta_j = \frac{\exp(v_j^T v_s)}{\sum_j \exp(v_j^T v_s)} \tag{6}$$

$$d = \sum_j \beta_j H_j \tag{7}$$

The overall process flow of the proposed Hierarchical Attention pointer based Stacked Denoising Variational Autoencoder Model is shown in Fig. 5.

%3CmxGraphModel%3E%3Croot%3E%3CmxCell%
20id%3D%220%22%2F%3E%3CmxCell%20id%3D%2
21%22%20parent%3D%220%22%2F%3E%3CmxCell%
20id%3D%222%22%20value%3D%22%26lt%3Bb
%26gt%3B%26lt%3Bfont%20style%3D%26quot%3B-
font-size%3A%2018px%3B%26quot%3B%20face%
3D%26quot%3BTimes%20New%20Roman%26quot%3B
%26gt%3BPre-Processed%26lt%3Bbr%26gt%3Bweb%
20Text%26amp%3Bnbsp%3B%26lt%3Bbr%26gt%3Bfr
om%20CBoW%26lt%3Bbr%26gt%3BModel%26lt%3B
%2Ffont%26gt%3B%26lt%3B%2Fb%26gt%3B%22%20
style%3D%22rounded%3D0%3BwhiteSpace%3Dwrap%
3Bhtml%3D1%3B%22%20vertex%3D%221%22%20p
arent%3D%221%22%3E%3CmxGeometry%20x%3D
%22–210%22%20y%3D%22,110%22%20width%3D%
22,120%22%20height%3D%22,130%22%20as%3D%
22geometry%22%2F%3E%3C%2FmxCell%3E%3C%2
Froot%3E%3C%2FmxGraphModel%3E.

The two-phase operation of the proposed model is useful in extracting features from contextualized distribution including the words with multiple meanings and semantic characteristics in which the SVDAE model generates smooth latent distribution of contextualized words and sentences with extraction of useful features as well as attention mechanism is adopted to produce context level vector in order to get the information about the contextualized web text with extracting keywords to determine the meaning of sentences and words in web text. Furthermore, after feature extraction, ranking and

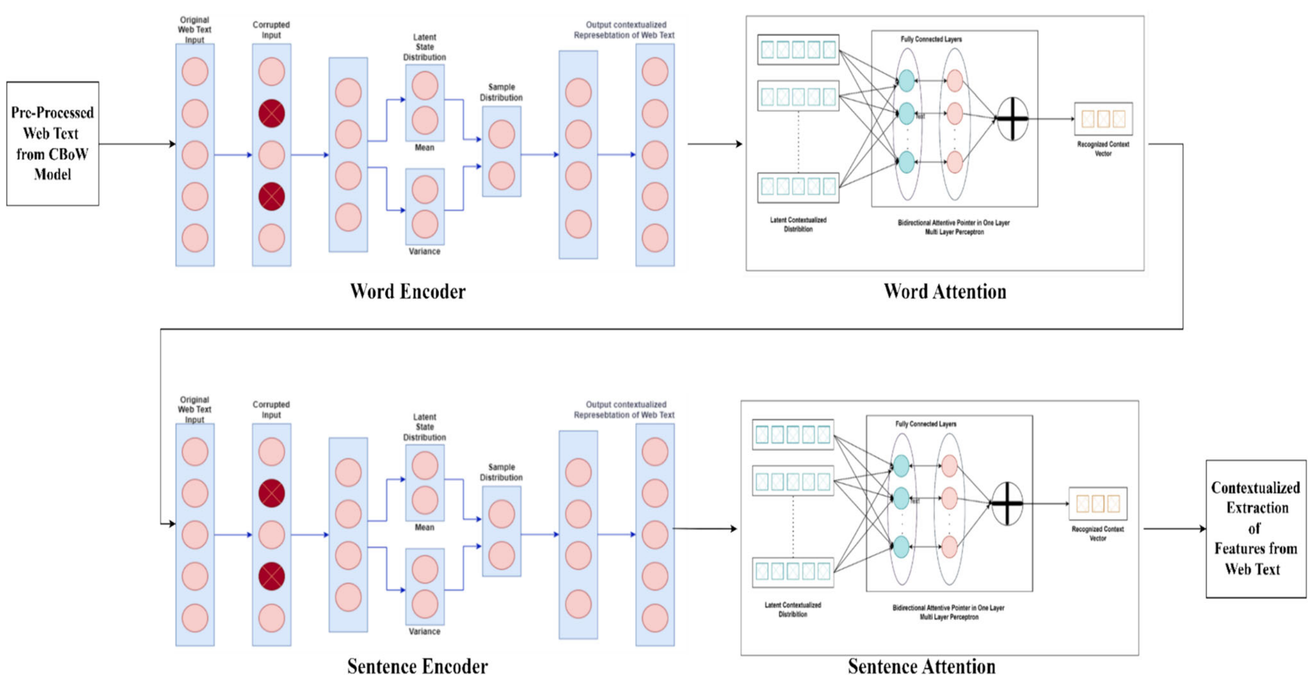


Fig. 5 Architecture of hierarchical attention pointer based stacked denoising variational autoencoder model

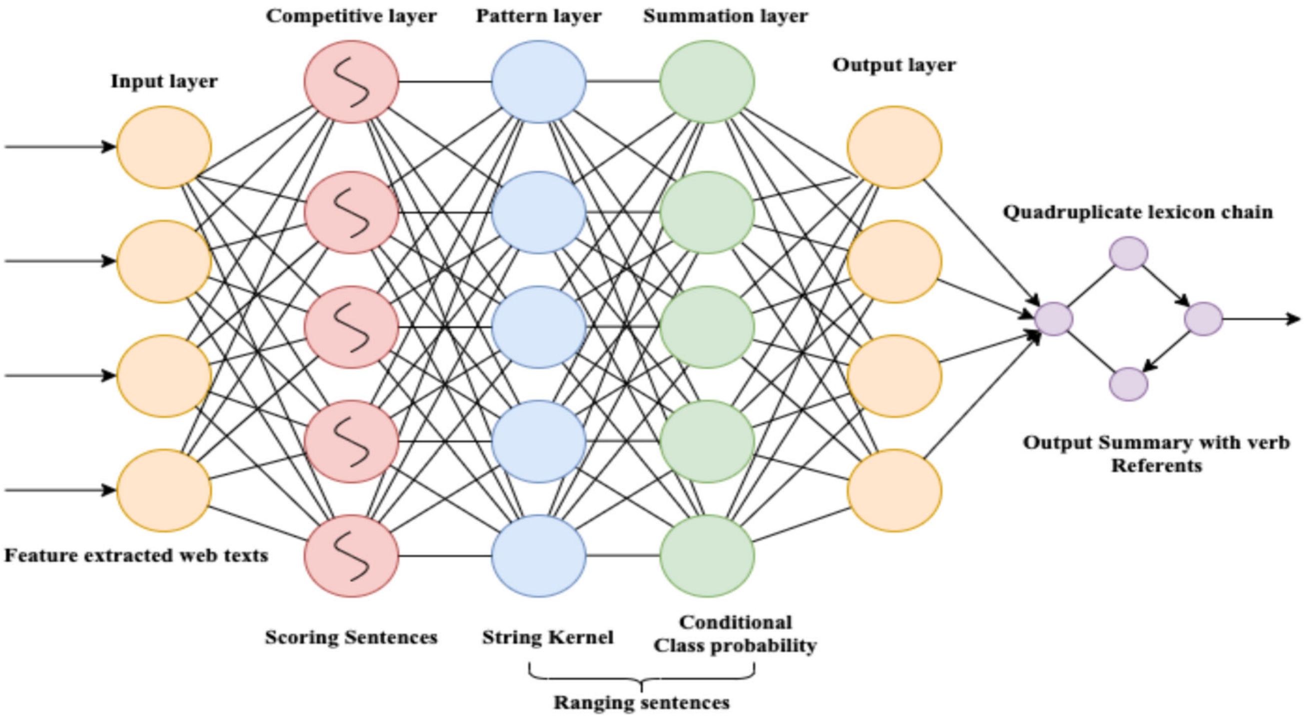
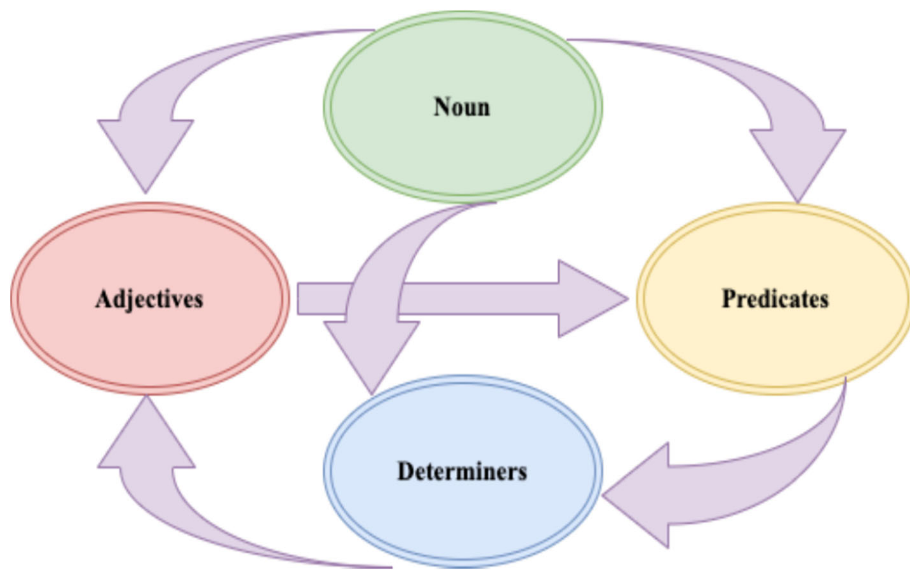


Fig. 6 Ranking and scoring in multilayered competitive probable modular perception model with summary generation using graph-based quadruplicate lexicon chain

Fig. 7 Graph-based quadruplicate lexicon chains



scoring have to be performed to produce the summary which is explained in the next subsection.

4.3 Multilayered Competitive Probable Modular Perception Model Graph-Based Quadruplicate Lexicon Summarization

The sentence-level context vector formed from the hierarchical model is taken as the input to the Multilayered Competitive Probable Modular Perception Model. This proposed model effectively scores and ranks the sentences based on proximity, singularity, cue words, word frequency, sentence position, and length value. Then, verb referents were checked in these ranked sentences without dangling anaphora using Graph-based Quadruplicate Lexicon Summarization. The process takes place in a multilayered competitive probable modular model and Graph

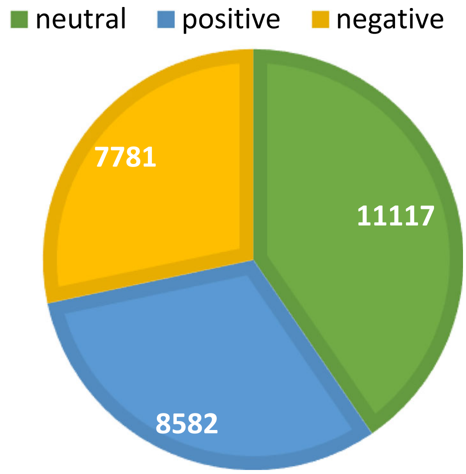


Fig. 8 Polarity of sentiments in the dataset

based Quadruplicate lexicon summarization is shown in Fig. 6.

The input layer processes the feature and keyword-extracted web text and sends the sentence-level context vector to the competitive layer. The competitive probability layer is responsible for scoring the sentences. In this competitive layer, the similarity and proximity measure between the individual input vector and the neuron weight for each competitive neuron has been determined. The neurons in this layer compete with each scoring for each input vector to see which one is the most similar to it thereby effectively score the sentences. Then, the scored sentences were ranked using pattern layer with string kernel and summation layer with conditional probability. The magnitudes of classmates' string kernels are sorted and rated in ascending order. Only a part of the string kernels competes to be included in the class-conditional probability calculation. The number of selected string kernels in class-a is designated as the number of leaders. The ratio μ_a is defined as the ratio of number of leaders (L_a) to the total number of string kernels in each class-$a(C_a)$ which is expressed in Eq. (8):

$$\mu_a = \frac{L_a}{|C_a|} \rightarrow L_a = [\mu_a |C_a|] \quad (8)$$

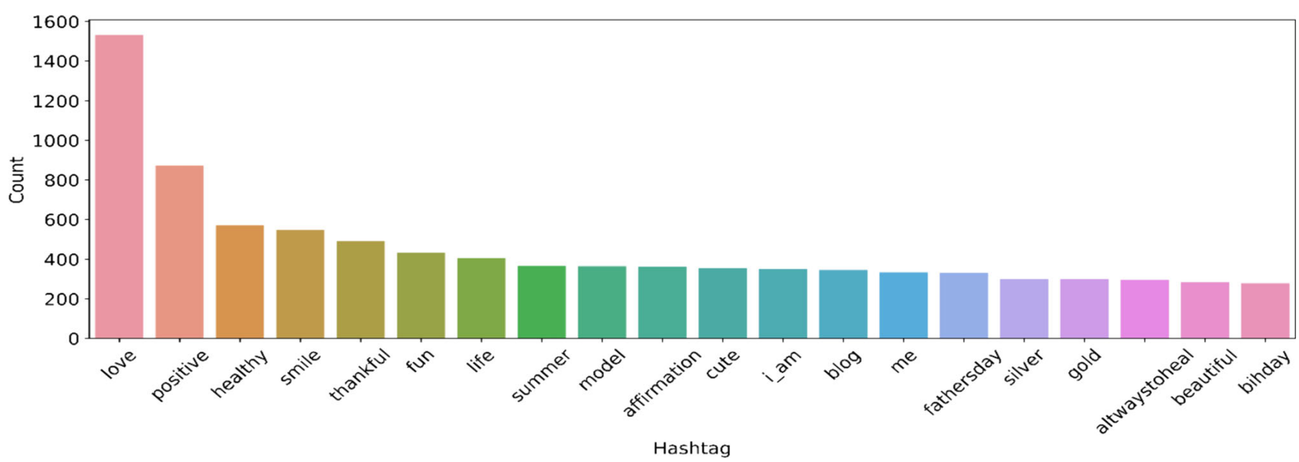


Fig. 9 Keyword count from hashtag

Fig. 10 Smooth latent representation of neutral word features

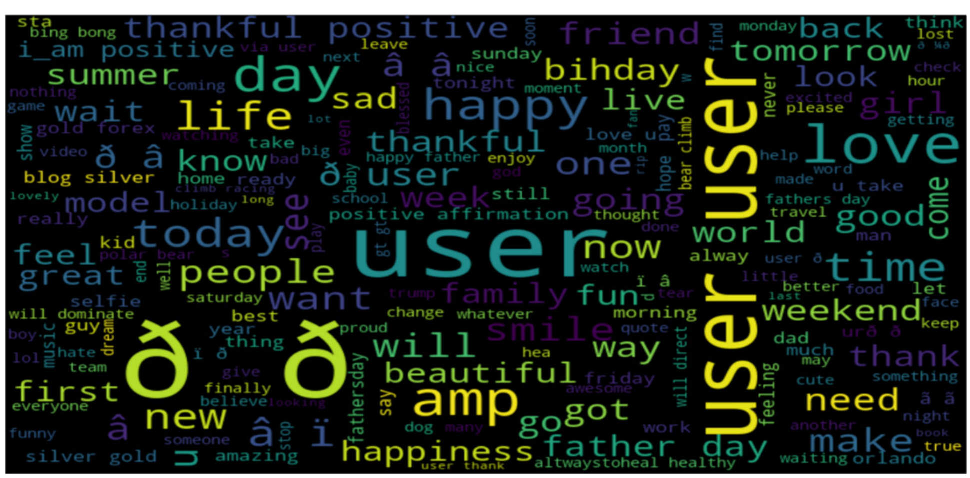


Fig. 11 Extracted keyword features from vocabulary in reviews

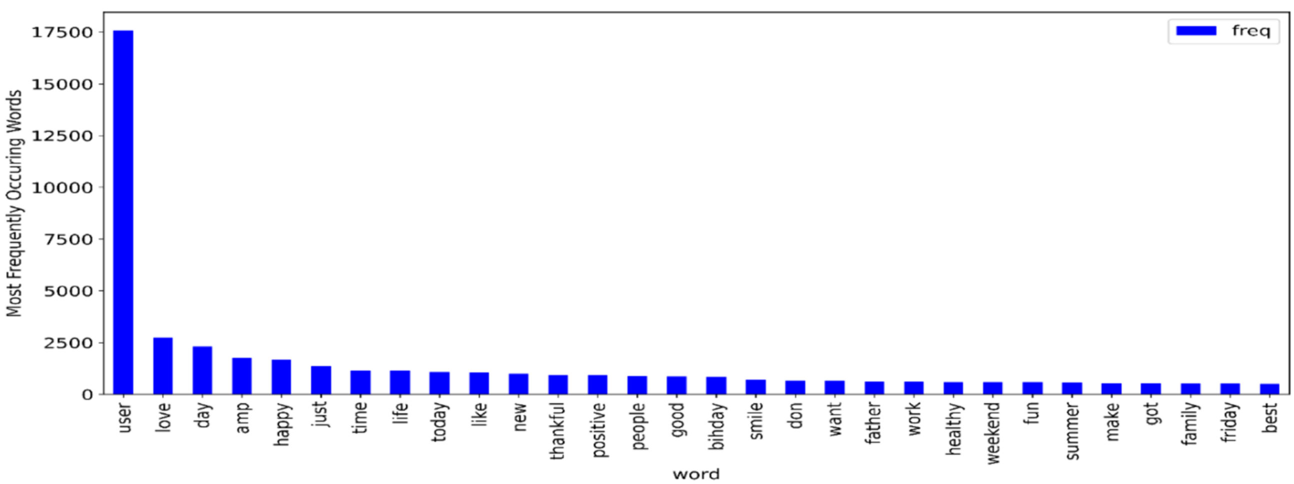


Fig. 12 Scoring and ranking based on similarity and proximity

This ratio μ_a is normalized and minimized which is expressed in Eq. (9):

$$\mu_{amin} = \frac{1}{|C_a|} \leq \mu_a \leq 1 \tag{9}$$

The subscript is a discrete variable with a finite number of possible values within the limit 1. The class-conditional probability P_a of each class is calculated using this Eq. (10):

$$P_a = \frac{1}{L_a}\sum_{k=1}^{L_a} W_{ak} = \frac{1}{[\mu_a|C_a|]}\sum_{k=1}^{[\mu_a|C_a|]} W_{ak} \tag{10}$$

where,W_{ak} is the weight distribution parameter in the classes a and k.

Based on these calculations, the class-conditional probability ranks the scored sentences to form the summary. These scored sentences were given to the graph-based Quadruplicate Lexicon Summarization mechanism in which the lexicon chain is formed with the quadruplicates of nouns, adjectives, determiners, and predictors in a graphical structure which is shown in Fig. 7.

The lexicon chain is formed in a graph with the consideration of noun, predicates, determiners, and adjectives in a sentence to form the summary without dangling anaphora expressions since these quadruplicates form the major grammatical parts of the sentence. Noun candidate words, determiner candidate words, predicate candidate words, and adjective candidate words are the quadruplicate parts of lexical chains' candidate words. All of the words were first steamed. For noun candidate terms, choose all named entities that describe the text's topics. Predicate candidate words are made up of all predicates in each phrase. Each phrase has a single predicate that serves as both the root of the dependency tree and a representation of the attribute that a subject possesses. As adjective (adverb) candidate terms, all noun-qualifying adjectives and adverbs were picked. Determiner words that were used before the nouns are selected as determiner candidate words. The quadruplicate lexicon chain is expressed in Eq. (11) as:

$$QLC = <NChain, DChain, PChain, AChain> \tag{11}$$

Equation (11) includes noun, determiner, predicate, and adverb terms in a quadruplicate lexicon chain format. The graphic structure created in Fig. 7 is an undirected graph, where each word's occurrence is represented by a vertex, and the semantic relationship between two occurrences is represented by an edge. If a word occurs in more than one sense, find and save the most pertinent one. Remove all additional senses and their boundaries. In the traverse semantic graph, words connected by edges create a lexical chain. All building lexical chains contain the final lexical chains. These chains provide sentences with correct meaning and grammatical form without cataphoric and anaphoric expressions. The overall algorithm for web text summarization model has been explained as follows:

Algorithm 1 Algorithm of proposed web text summarization model

Input: *Source web text document*

Output: *Summary of web text*

Initialize *web text summarization model*

do

{

Get *sentences S from web text document D*

/ perform stop word removal, lemmatization, and tokenization as preprocessing steps*/*

for *sentences in D*

    stop words = set (stop words. words(S))

    tokens= do linguistic analysis (S)

    word= word tokenize(S)

    for *words in S*

        lemmatization= porter stemmer. Stem (W)   */*W is words*/*

    end

end

Generate *vectorized words (v_w) using the CBoW model*

Construct *contextualized representation of W and S in encoding process*

Determine *the probability of contextualized representation using eqn (1)*

Extract *features and keywords using eqn (2) - (7)*

for *scoring sentences in D*

    $S[i]$ *= word frequencies[word]+singularity [words]*

end

Rank *scored sentences using eqn (8) – (10)*

Generate *extractive summary with correct grammatical form using eqn (11)*

end

Overall, Extractive Web Text Summarization using CBoW Text Vector Model has been presented to generate a summary for web text by preprocessing and vectorizing the web text. Then, features were extracted to understand the meaning of all words and sentences in the web text. Finally, the summary has been generated by efficient ranking with the construction of graph graph-based quadruplicate lexicon chain. The next section explains the result obtained from the proposed model and discusses it in detail.

5 Experiment Result and Discussion

This section includes a thorough analysis of the performance of the proposed system, the implementation results simulated in the Python platform, and a comparison section to make sure the suggested system is appropriate for web text summarizing.

5.1 Dataset Description

Twitter sentiments-text summarization dataset[1] consists of 940 tweets annotated by 22 human annotators with sentiments in positive polarity, negative polarity, and neutral. This dataset is organized with 27,482 rows with 4 columns for describing the text ID, text, selected text, and sentiments. Where the first column the text ID, contains the Twitter ID of the text. The second column Text, presents the original tweet of the text while the third column, selected text, contains only the selected text from the tweets. The fourth column, sentiments, contains positive, neutral, and negative polarity sentiments. Figure 8 depicts the 27,482 rows categorized into three polarity of the sentence 11,117 neutral tweets, 8583 positive tweets, and 7782 negative tweets.

5.2 Experimental Results

Web text data has been preprocessed, vectorized, feature extracted, and summarized using novel techniques, and the results obtained were described in this section. The Dataset was split into 25% for testing a 75% training set.

Figure 9 depicts the important keyword count from the Hashtags in the Twitter sentiments-text summarization dataset. The Twitter text was preprocessed by removing the unwanted stop words, extracting stem words, and tokenization. Then, the CBoW text vectorization model was applied which transforms the text into vectors and forms a distributed combination of words with the most useful keyword counts as shown in Fig. 9.

The smooth latent representation of neutral words in the Twitter reviews were shown in Fig. 10. This contextualized distribution of words was obtained using a Stacked Denoising Variational Autoencoder which extracts all possible meanings of words and performs a denoising mechanism to represent the text without noise.

Figure 11 depicts the keyword features extracted from the vocabulary in reviews. The bidirectional attentive pointer mechanism in the hierarchical model extracts the important keywords and forms sentence vectors in a bidirectional pattern. This mechanism focuses on the keywords from vocabulary in reviews by understanding the meaning of words based on contextualized sentence which is shown in Fig. 11.

The scoring and ranking of words in sentences are depicted in Fig. 12. The words in the sentences were scored based on the similarity and proximity of their occurrence as shown. These scored words in the sentence were ranked by using a pattern layer with string kernel and class-conditional probability mechanism. Then, a quadruplicate lexicon chain is constructed to represent the summary finally.

5.3 Performance Metrics

Precision: The number of positively predicted cases that were truly accurate is known as precision. This formula is used to compute it:

$$Pricision = \frac{True\,Positive}{True\,Positive + False\,Positive} \tag{12}$$

Recall: It assesses a model's capacity to locate every pertinent instance within the dataset. This formula is used to compute it:

$$Recall = \frac{True\,Positive}{True\,Positive + False\,Negative} \tag{13}$$

F1-Score: The harmonic mean of recall and precision is the F1 score. It is an effective statistic for striking a balance between precision and recall because it incorporates both into one. This formula is used to compute it:

$$F1_{Score} = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \tag{14}$$

A wider measure called accuracy evaluates how accurate a model's predictions are overall. This formula is used to compute it:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{15}$$

[1] https://www.kaggle.com/competitions/tweet-sentiment-extraction/data.

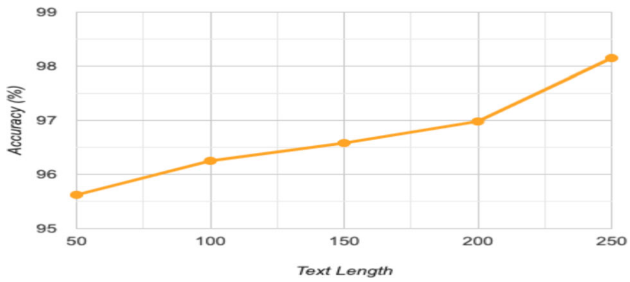


Fig. 13 Accuracy of proposed model based on text lengths

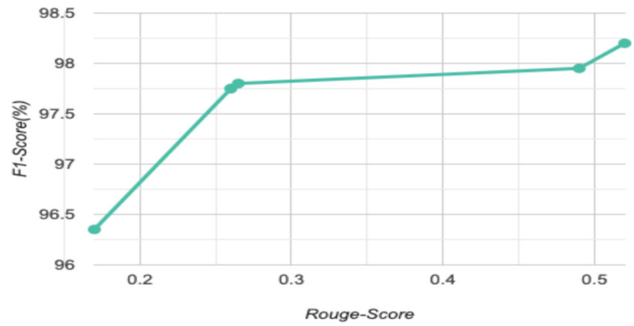


Fig. 15 Rouge score of proposed web text model

ROUGE-N: Calculates how much the generated text and the reference text overlap in terms of n-grams, or word sequences. Unigrams (single words) are referred to as ROUGE-1, bigrams (two-word sequences) as ROUGE-2, and so on. ROUGE-L determines the generated text's longest common subsequence (LCS) from the reference text. It considers how long the longest common subsequence is.

The performance of the proposed approach and the achieved outcome were explained in detail in this section.

The accuracy of the proposed model by varying the text length of the tweet data is demonstrated in Fig. 13. The accuracy attains a maximum value of 98.30% while increasing the text length. The accuracy of the proposed system attains a minimum value of 95.63% while decreasing the length of text. The accuracy of the proposed system is increased by using Multilayered Competitive Probable Modular Perception Model and Graph-based Quadruplicate Lexicon Summarization that effectively rank and generates summary with class conditional probability mechanism and graphical lexicon chains.

Figure 14 depicts the F1-Score of the proposed model and F1-Score was calculated for varying the text length from 50 to 250. The proposed model has a maximum F1-Score of 98.4% and a minimum F1-Score value of 96.2%. The F1-Score of the proposed model increases with the increase in the length of text. The F1-Score of the proposed web text model is improved by a Hierarchical Attention pointer based Stacked Denoising Variational Autoencoder Model which extracts keyword features accurately with a bidirectional attention mechanism. The recall of the proposed web text model by varying the text length of the tweet data is also depicted. The recall attains a maximum value of 98.8% while increasing the text length. The recall of the proposed system attains a minimum value of 97.5% while decreasing the length of text. The recall of the proposed system is increased by using the Multilayered Competitive Probable Modular Perception Model and Graph-based Quadruplicate Lexicon Summarization that accurately generates summary with class conditional probability mechanism and quadruplicate lexicon chains. The overall precision of the proposed model and the precision was calculated by varying the text length from 50 to 250. The proposed model has a maximum precision of 98% and a minimum precision value of 95.3%. The precision of

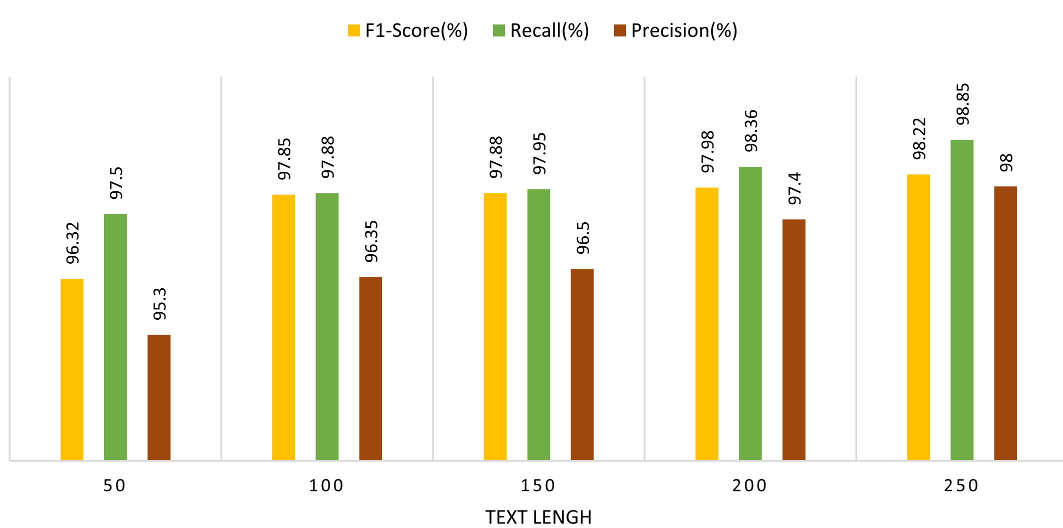


Fig. 14 Recall, F1-Score, precision of over text length of the proposed work

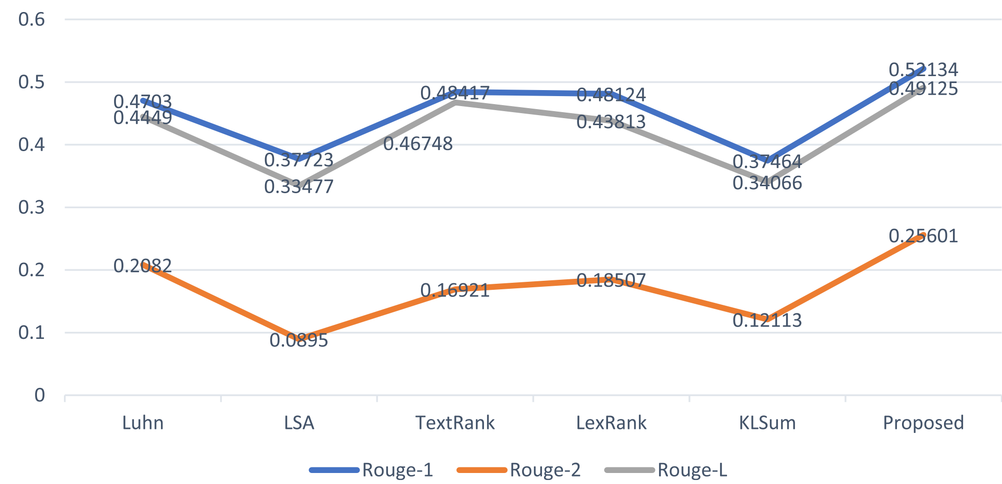


Fig. 16 Comparison of proposed model Rouge-1, Rouge-L and Rouge-2 with the existing approaches in terms of F-measure

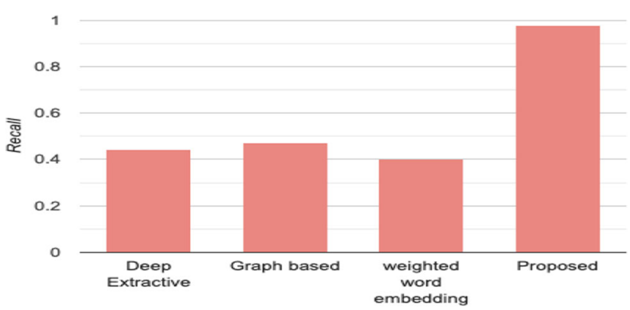


Fig. 17 Comparison of recall

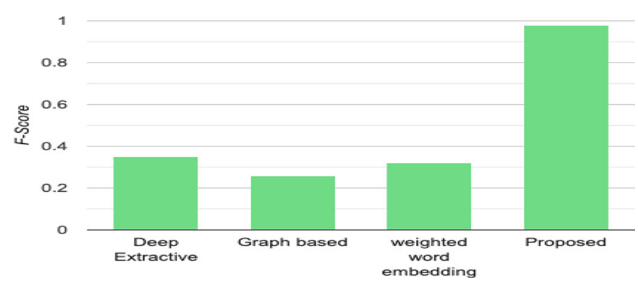


Fig. 19 Comparison of F-score

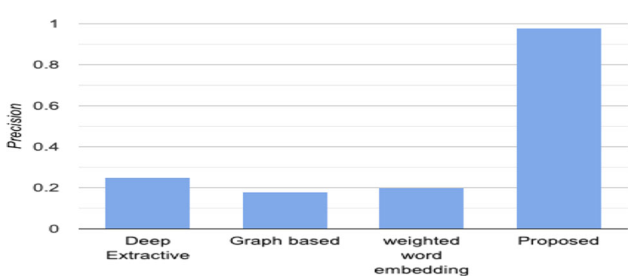


Fig. 18 Comparison of precision

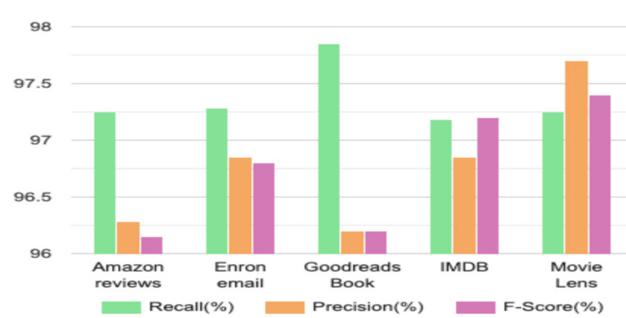


Fig. 20 F1-Score comparison with various datasets

the proposed model increases with the increase in the length of the text. The precision of the proposed model is improved by the Multilayered Competitive Probable Modular Perception Model that precisely scores the sentences based on proximity and singularity using the Competitive probable layer.

The Rouge score of the proposed web text model in terms of F1-Score has been shown in Fig. 15. The Rouge score has the highest value of 0.54 when the F1-score is 98.4% and has the lowest value of 96.2%. The rouge score increases with the increase in the F-score. The rouge-score and F-Score were improved by using the Hierarchical Attention pointer-based Stacked Denoising Variational Autoencoder Model.

5.4 Comparison Results of the Proposed Method

This section highlights the proposed model performance by comparing it to the outcomes of existing approaches and showing their results based on various metrics.

Figure 16 depicts the comparison of rouge-1 that is rouge score of unigram in terms of F-measure. The rouge-1 of the proposed model is compared with existing techniques such as Luhn (Luhn Apr. 1958), LSA (Landauer et al. 1998), TextRank (Mallick, et al. 2019), LexRank (Erkan and Radev 2004), and KLSum (Haghighi and Vanderwende 2009). The rouge-1 of the proposed system has the maximum value of 0.524 whereas the rouge-1 of Luhn, LSA, TextRank, LexRank, and KLSum is 0.472, 0.375, 0.472, 0.426, and 0.375 respectively. The rouge-1 score of the proposed model is high whereas the rouge-1 score of LSA and KLSum is low. Figure 16 depicts the comparison of rouge-2 which is the rouge score of bigram in terms of F-measure. The comparison of rouge-2 of the proposed system with existing techniques such as Luhn, LSA, TextRank, LexRank, and KLSum. The rouge-2 of the proposed model attains the maximum value of 0.3 whereas the rouge-2 of existing techniques such as Luhn, LSA, TextRank, LexRank, and KLSum are 0.22, 0.7, 0.17, 0.18, and 0.13 respectively. Hence, the rouge-2 of the proposed model is high whereas the rouge-2 of LSA is low. The comparison of Rouge-L which is the rouge score of word length in terms of F-measure is also discussed in Fig. 16. The rouge- L of the proposed model is compared with existing techniques such as Luhn, LSA, TextRank, LexRank, and KLSum The rouge- L of the proposed system has the maximum value of 0.484 whereas the rouge L of Luhn, LSA, TextRank, LexRank and KLSum are 0.446, 0.342, 0.470, 0.433 and 0.345 respectively. The rouge- L score of the proposed model is high whereas the rouge- L score of LSA is low.

Figure 17 shows a comparison of the recall of the proposed model with existing techniques such as deep extractive (Bhargava and Sharma 2020), Graph based (Belwal et al. 2021) and weighted word embedding (Rani and Lobiyal 2021). The recall of the proposed system attains a maximum value of 98% whereas the recall of deep extractive, Graph based, and weighted word embedding are 44%, 47%,and 40% respectively. Hence, the proposed web text model has the highest recall, whereas weighted word embedding has the lowest recall.

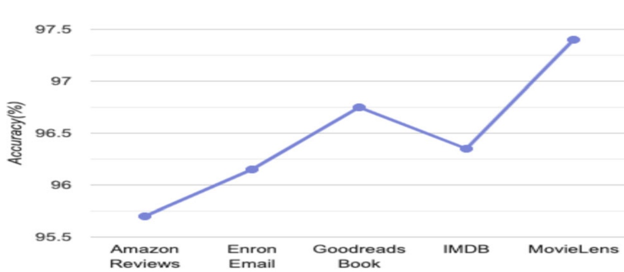


Fig. 21 Accuracy comparison with various datasets

Figure 18 shows a comparison of the precision of the proposed model with existing techniques such as deep extractive, Graph based,and weighted word embedding. The precision of the proposed system attains a maximum value of 98% whereas the precision of deep extractive, Graph based and weighted word embedding are 25%, 18% and 20% respectively. Hence, the proposed web text model has the highest precision, whereas graph based has the lowest precision.

The comparison of the F-measure of the proposed system with existing techniques such as deep extractive, Graph based, and weighted word embedding is shown in Fig. 19. The F-measure of the proposed model attains the maximum value of 98% whereas the F-measure of existing techniques such as deep extractive, Graph based, and weighted word embedding are 35%, 26%, and 32% respectively. Hence, the F-measure of the proposed model is high whereas the F-measure of Graph based is low.

Figure 20 shows the F1-score comparison of various datasets such as Amazon reviews, Enron email, Goodreads Book, IMDB, and MovieLens. It is noticed that the proposed web text model performs well with all datasets with a minimum of 96.15% and a maximum of 97.4% F1-Score.

Figure 21 shows the accuracy comparison of various datasets such as Amazon reviews, Enron email, Goodreads Book, IMDB, and Movie Lens. It is noticed that the proposed web text model performs well with all datasets with a minimum of 96.2% and a maximum of 97.5% accuracy. Figure 20 shows the recall comparison of various datasets such as Amazon reviews, Enron email, Goodreads Book, IMDB, and Movie Lens. It is noticed that the proposed web text model performs well with all dataset with a minimum of 97.18% and maximum of 97.85% recall. Figure 20 shows the precision comparison of various datasets such as Amazon reviews, Enron email, Goodreads Book, IMDB, and Movie Lens. It is noticed that the proposed web text model performs well with all dataset with a minimum of 96.2% and a maximum of 97.7% precision.

Overall, Extractive Web Text Summarization using CBoW Text Vector Model outperforms existing techniques such as Luhn, LSA, TextRank, LexRank, KLSum deep extractive, Graph based, and weighted word embedding by processing text quickly with a continuous bag of words and extracting features accurately with understanding the context using a hierarchical model. Then, the summary is generated using competitive and class-conditional probability mechanisms. Thus, the result achieved has a high rouge score of 0.524, precision of 98%, recall of 98%, and F-score of 98%.

6 Conclusion

In this article, Extractive Web Text Summarization using CBoW Text Vector Model has been introduced to solve the issues such as huge processing time, difficulty in extracting polysemous features, and dangling anaphora in the summary generation of web text data. The vectorized word distribution is obtained using the CBoW text vectorization model and contextualized sentence distribution with consideration of all possible word meanings is extracted using Hierarchical Attention pointer based Stacked Denoising Variational Autoencoder Model and finally, the summary is generated using Multilayered Competitive Probable Modular Perception Model and Graph-based Quadruplicate Lexicon Summarization. The result obtained from Extractive Web Text Summarization using the CBoW Text Vector Model outperforms existing techniques with a high rouge score of 0.524, precision of 98%, recall of 98%, and F-score of 98%. As part of our future work, researchers can take the opportunity to explore these problems for abstractive summarization.

References

Abualigah L, Bashabsheh MQ, Alabool H, Shehab M (2020) Text summarization: a brief review. In: Elaziz MA, Al-Qaness MAA, Ewees AA, Dahou A (eds) Recent advances in NLP: the case of Arabic language. Springer, Cham, pp 1–15

Alami N, Meknassi M, En-nahnahi N (2019) Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning. Expert Syst Appl 123:195–211

Al-Maleh M, Desouki S (2020) Arabic text summarization using deep learning approach. J Big Data 7(1):1–17

Belwal RC, Rai S, Gupta A (2021) A new graph-based extractive text summarization using keywords or topic modeling. J Ambient Intell Human Comput 12(10):8975–8990

Bhargava R, Sharma Y (2020) Deep extractive text summarization. Procedia Comput Sci 167:138–146

Chatterjee N, Yadav N (2019) Hybrid latent semantic analysis and random indexing model for text summarization. In: Fong S, Akashe S, Mahalle P (eds) Information and communication technology for competitive strategies. Springer, Singapore, pp 149–156

Doğan E, and Kaya B (2019) Deep learning based sentiment analysis and text summarization in social networks. In: 2019 International artificial intelligence and data processing symposium (IDAP). IEEE

Elbarougy R, Behery G, El Khatib A (2020) Extractive Arabic text summarization using modified PageRank algorithm. Egypt Inform J 21(2):73–81

El-Kassas WS et al (2020) EdgeSumm: graph-based framework for automatic text summarization. Inform Process Manag 57(6):102264

El-Kassas WS et al (2021) Automatic text summarization: a comprehensive survey. Expert Syst Appl 165:113679

Erkan G, Radev DR (2004) Lexrank: graph-based lexical centrality as salience in text summarization. J Artif Intell Res 22:457–479

Gangathimmappa M, Subramani N, Sambath V, Ramanujam RAM, Sammeta N, Marimuthu M (2023) Deep learning enabled cross-lingual search with metaheuristic web based query optimization model for multi-document summarization. Concurr Computat: Pract Exp 35(2):e7476

Ghadimi A, Beigy H (2023) SGCSumm: an extractive multi-document summarization method based on pre-trained language model, submodularity, and graph convolutional neural networks. Expert Syst Appl 215:119308. https://doi.org/10.1016/j.eswa.2022.119308

Ghodratnama S et al (2020) Extractive document summarization based on dynamic feature space mapping. IEEE Access 8:139084–139095

Haghighi A, and Vanderwende L (2009) Exploring content models for multi-document summarization. In: Proceedings of human language technologies. The 2009 annual conference of the North American chapter of the association for computational linguistics

Hernández-Castañeda Á et al (2020) Extractive automatic text summarization based on lexical-semantic keywords. IEEE Access 8:49896–49907

Hossain Md, Hoque MM (2020) Semantic meaning based Bengali web text categorization using deep convolutional and recurrent neural networks (DCRNNs). In: Misra R, Kesswani N, Rajarajan M, Bharadwaj V, Patel A (eds) International conference on internet of things and connected technologies. Springer, Cham

Joshi A, Fidalgo E, Alegre E, Fernández-Robles L (2023) DeepSumm: exploiting topic models and sequence to sequence networks for extractive text summarization. Expert Syst Appl 211:118442

Kanapala A, Pal S, Pamula R (2019) Text summarization from legal documents: a survey. Artif Intell Rev 51(3):371–402

Kumar Y, Kaur K, Kaur S (2021) Study of automatic text summarization approaches in different languages. Artif Intell Rev 54(8):5897–5929

Landauer TK, Foltz PW, Laham D (1998) An introduction to latent semantic analysis. Discourse Process 25(2–3):259–284. https://doi.org/10.1080/01638539809545028

Li S, Pan R, Luo H, Liu X, Zhao G (2021) Adaptive cross-contextual word embedding for word polysemy with unsupervised topic modeling. Knowl-Based Syst 218:106827

Luhn HP (1958) The automatic creation of literature abstracts. IBM J Res Dev 2(2):159–165. https://doi.org/10.1147/rd.22.0159

Luo Z, Xie Q, and Ananiadou S (2023) Chatgpt as a factual inconsistency evaluator for abstractive text summarization. arXiv preprint arXiv:2303.15621

Ma C, Wu Z, Wang J, Xu S, Wei Y, Liu Z, Guo L et al (2023) ImpressionGPT: an iterative optimizing framework for radiology report summarization with chatGPT. arXiv preprint arXiv:2304.08448

Madhuri JN, and Kumar RG (2019) Extractive text summarization using sentence ranking.In: 2019 International conference on data science and communication (IconDSC). IEEE

Magdum PG, Rathi S (2021) A survey on deep learning-based automatic text summarization models. In: Chiplunkar NN, Fukao T (eds) Advances in artificial intelligence and data engineering. Springer, Singapore, pp 377–392

Mallick C et al (2019) Graph-based text summarization using modified TextRank. In: Nayak J, Abraham A, Krishna BM, Sekhar GTC, Das AK (eds) Soft computing in data analytics. Springer, Singapore, pp 137–146

Manjari KU et al. (2020) Extractive Text Summarization from Web pages using Selenium and TF-IDF algorithm.In: 2020 4th International conference on trends in electronics and informatics (ICOEI)(48184). IEEE

Mao X et al (2019) Extractive summarization using supervised and unsupervised learning. Expert Syst Appl 133:173–181

Mao, Yuning, et al. (2020) Multi-document summarization with maximal marginal relevance-guided reinforcement learning. arXiv preprint arXiv:2010.00117

Mishra AR, Panchal VK, and Kumar P (2019) Extractive text summarization-an effective approach to extract information from Text. In: 2019 International conference on contemporary computing and informatics (IC3I). IEEE

Patel D, Shah S, Chhinkaniwala H (2019) Fuzzy logic based multi document summarization with improved sentence scoring and redundancy removal technique. Expert Syst Appl 134:167–177

Rani R, Lobiyal DK (2021) A weighted word embedding based approach for extractive text summarization. Expert Syst Appl 186:115867

Shini RS, and Kumar VDA (2021) Recurrent neural network based text summarization techniques by word sequence generation. In: 2021 6th International conference on inventive computation technologies (ICICT). IEEE, 2021.

Shirwandkar NS, and Kulkarni S (2018) Extractive text summarization using deep learning. In: 2018 Fourth international conference on computing communication control and automation (ICCUBEA). IEEE

Siautama R, Suhartono D (2021) Extractive hotel review summarization based on TF/IDF and adjective-noun pairing by considering annual sentiment trends. Procedia Comput Sci 179:558–565

Song S, Huang H, Ruan T (2019) Abstractive text summarization using LSTM-CNN based deep learning. Multimed Tools Appl 78(1):857–875

Steinberger J, Poesio M, Kabadjov MA, Ježek K (2007) Two uses of anaphora resolution in summarization. Inf Process Manag 43(6):1663–1680

Suleiman D, and Awajan AA. (2019) Deep learning based extractive text summarization: approaches, datasets and evaluation measures. In: 2019 Sixth international conference on social networks analysis, management and security (SNAMS). IEEE

Uçkan T, Karcı A (2020) Extractive multi-document text summarization based on graph independent sets. Egypt Inform J 21(3):145–157

Wang D et al (2013) Comparative document summarization via discriminative sentence selection. Trans Knowl Discov Data 7(1):2

Zhou Q et al (2020) A joint sentence scoring and selection framework for neural extractive document summarization. IEEE/ACM Trans Audio Speech Lang Process 28:671–681