J. Integr. Sci. Technol. 2024, 12(2), 741

Journal of Integrated
SCIENCE & TECHNOLOGY

Text summarization based on human behavioural learning model

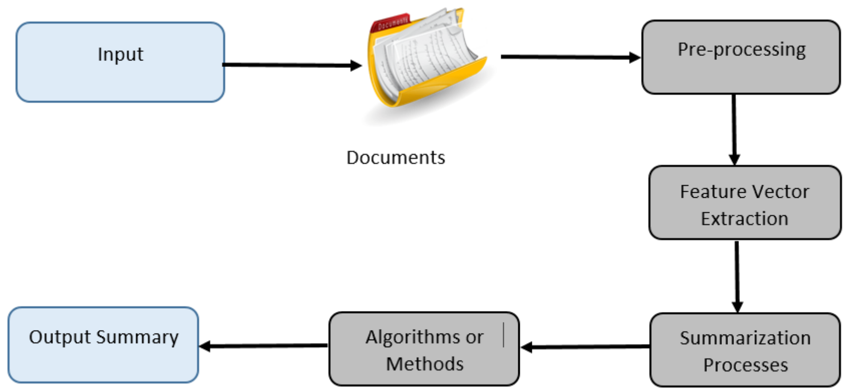
Namita Kale,* Ranjana P. Dahake, Kalpana V. Metre

MET'S Institute of Engineering, Nashik, Maharashtra, India

Received on: 02-Aug-2023, Accepted and Published on: 16-Oct-2023

Review

ABSTRACT



Summarization of text documents has begun to play an important role in information collection. Summarization has traditionally been done physically by humans, which has resulted in a time-consuming operation as the volume of data has become more and larger. To resolve this issue, automated text summarization has become a critical need for efficiently managing congested data. Previous research on text summarization has focused on summarizing pre-specified materials with no extra requirements and is sometimes referred to as a generic summary. Automatic document summarization, on the other hand, is the function of reducing the size of papers while still providing considerable semantic value. The automatic document summarization method consists of three phases: preprocessing, feature vector extraction, and summarization. Efficient preprocessing is crucial for achieving an excellent summarization system. These preprocessed documents are used for feature vector extraction, which is used to construct a sentence matrix. The extracted feature vectors are then used for summarization, generating a summary as output. The development of recent advances in the communication field has brought up deep learning methods and human knowledge intervention with cognitive models. As a result, this study investigates how modern artificial intelligence with optimized deep learning methods, as well as human information processing behavior, structures, and underlying processes, might be utilized in document summarization utilizing computational cognitive models. Based on precision, recall, and F-measure, this study also examines the usefulness of these models and their application in diverse document summarizing settings and activities.

Keywords: automated text summarization, artificial intelligence, optimized deep learning methods, computational cognitive models, precision

INTRODUCTION

The advent of high-speed networks, a huge increase in the number of interconnected mobile devices, and the enormous growth of social media platforms have resulted in a massive growth of textual, audio, and video data available for human use.[1] Various types of content from news services, movie and product reviews and recommendations, online course content, class notes, and sports to medical and legal references are in abundance. In addition, large organizations generate a large amount of textual and other types of data based on internal organizational information and client information.[2] While this increase in information about concepts and processes can help people, it can also bring about significant challenges. It is very difficult to find the required, relevant information when needed. This problem is compounded when the end user of the available information is a human. Humans are severely limited in processing a large amount of information.[3] Hence, there is a need for aggregating and condensing the information that is available on a topic from disparate sources, and in different forms. Even if only the textual information is considered, the amount of information that is available for human consumption is daunting.[4] As humans have difficulty processing large texts, summaries of such textual information can help humans understand the available information and make effective decisions in less time. Therefore, the text document summarization task and the currently available methods to create effective summaries of input text need to be investigated in detail.[5]

Many definitions for Document Summarization (DS) or text summarization exist. Summarization aims to perform a reductive transformation of the input text into a summary text by performing

*Corresponding Author: Namita Kale
Email: mrsnrkale@gmail.com

Cite as: *J. Integr. Sci. Technol., 2024, 12(2), 741.*
URN:NBN:sciencein.jist.2024.v12.741

 ISSN: 2321-4635
http://pubs.thesciencein.org/jist

content condensation and selecting the crucial information in the original text.[6] A tripartite processing model is assumed in summary generation: source text is interpreted to create a source text representation, the source text representation is then transformed into a summary text representation, and finally, the summary text representation is used to generate the summary text.[7] It is evident that the applications of document summarization are numerous and are quite diverse. Hence, identifying better summarization methods will help build better solutions in all these application domains.[8] Most existing datasets for document summarization use human-generated, manual text summaries as the gold standard. Statistical and learning model-based methods proposed by various research efforts are evaluated against such human-generated summaries.[9] Hence, naturally, the question of whether the human text comprehension and summarization processes can be used for creating summaries of the text arises. To answer this question, an understanding of human cognitive processes is necessary.[10]

The technique of automatic document summarization gathers a partly configured source text from manifold texts written on the identical subject, mines data contents from it, and offers the most valuable information to the client in a way, which gives maximum solace to the client.[11] Deep learning is the emerging field of machine learning, which is used to solve problems in several computer science domains like image processing, robotics, motion, etc. Recently it has also been used in the domain of natural language processing with very encouraging results.[12] An algorithm is considered a deep learning algorithm if the input is passed to the algorithm through several nonlinear layers so that the output of most modern learning algorithms including SVM and Naive Bayes classifier is shallow. Also, the restricted Boltzmann machine (RBM) is used to extract the top most relevant sentences from the given documents.[13] Computational Cognitive Models (CCM) are computational implementations of cognitive psychology models. Though different definitions of computational cognitive models exist in the literature, the significant, widely used definitions are given below. A cognitive psychology model is a hypothetical understanding of cognition and may be implemented as a computational, algorithmic model. Such an implemented computational model may make its assumptions and interpretations.[14] As long as the fundamental cognitive psychology model tenets are not violated, the computational cognitive model is assumed to be a reasonable algorithmic representation of the psychology model. A computational cognitive model implements the cognitive psychology model's processes, components, and relations between the processes and components.[15] It is a partial homomorphism and need not carry over all processes, components, and relations present in the original cognitive psychology model. However, it does need to preserve whatever relations are carried over to the computational model.[16] The rapid growth of textual, audio, and video data has led to information overload, making it difficult to find relevant information. Existing summarization methods may struggle to handle the diversity of data, which is available in various forms and sources. The summarization process can be complex, and existing methods may not always produce high-quality summaries. Evaluation standards often rely on human-generated manual summaries, which can be time-consuming and subjective. Understanding human text comprehension and summarization processes is essential for creating effective summaries. The proposed paper aims to address these limitations by focusing on efficiency, handling diverse data types, addressing complexity, and understanding human comprehension. By addressing these issues, the paper contributes to ongoing research efforts in improving text summarization methods.

Thus, this research tries to find out the efficiency of the recent trends like machine learning and cognitive model behavior analyze the functioning of the model and finally determine their superiority over each other. Cognitive models of behavior are essential for text summarization due to their ability to produce summaries that align with human comprehension, improve content selection, enhance coherence, handle ambiguity, adapt to different text types, consider reader expectations, provide a complex evaluation approach, and reduce the likelihood of biased or unfair summaries. These models bridge the gap between automated summarizing and human comprehension by incorporating ethical principles and human-like thinking, ensuring summaries are more in line with human reading and deciphering. The contribution of the proposed method is:

- Preprocessing must be done efficiently to achieve an excellent summary system.
- Initial preprocessing steps for the input documents used for automatic document summarization include sentence segmentation, tokenization, stop word removal, and stemming.
- The feature vector extraction procedure then uses the preprocessed input documents as input.
- The input for the summarization is the extracted feature vector.
- Numerous document summarizing methodologies and techniques have been documented in the scientific literature that are now available on automatic text summarization.

Background of the Study

The general architecture of the automatic document summarization method consists of three phases namely preprocessing, feature vector extraction, and summarization process as shown in Figure 2.

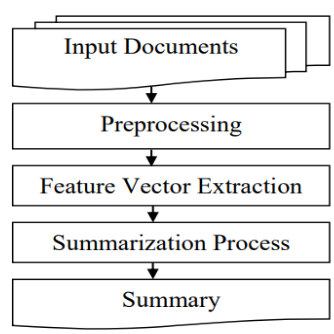


Figure 2. Automatic document summarization process

To achieve an excellent summarization system, preprocessing should be done efficiently. Initially, the input documents utilized for automatic document summarization are exposed to a set of preprocessing steps like, sentence segmentation, tokenization, stop word removal, and stemming.[17] The preprocessed input documents are then used as input for feature vector extraction.[18] It is performed for every sentence in each input document. The features will be defined as the characteristics of each sentence, which will help identify the sentences according to their relevance. The selected feature vectors will be used to construct a sentence matrix. The matrix contains the feature vectors of each sentence. The row of the matrix represents the sentences present in the document and the column represents the features extracted from the input document. The extracted feature vector is the input for the summarization. With the help of the summarization techniques, the resultant summary is generated as the output.[19] In the available scientific literature regarding automatic text summarization, numerous document summarization approaches and techniques have been listed. The most widely used and effective approaches for document summarization are as follows

Baseline Approaches

Lead baseline summary[20] is typically constructed by selecting the lead sentences in documents or at the important locations of the document texts such as paragraphs, subsections, etc. The number of sentences selected could vary. For example, a lead-3 summary would extract 3 sentences from the preferred location (the beginning of a document, a paragraph, and a section of the document). This approach requires the ability to recognize the start of a section of a document. Variations include selecting sentences from various parts of the same section such as the beginning, middle, and end of the section. The number of sentences selected can also vary between implementations.

A random baseline summary[21] is constructed by selecting sentences randomly from the input text. In addition to a completely random selection of sentences, random selections within sections of a document can also be used for constructing the summary. These techniques are easier to implement when compared to linguistic, graph-based, machine-learning, and cognitive approaches. In addition, these techniques are typically very efficient. However, due to their simplistic nature and disregard for information content and context while creating summaries, the effectiveness achieved by the lead and random baseline methods[22] are typically inadequate for practical applications.

Statistical Approaches

A combination of statistical and semantic similarity has been used to create text summaries.[23] The given input text sentences were converted into a graph structure based on statistical similarity and semantic similarity. Text Rank related sentences to find their relative importance. Finally, the sentences were grouped based on a similarity measure. This work also considered co-references and discourse relations for creating the summary. Information redundancy was reduced and diversity was enhanced by selecting high-scoring sentences from each group and selecting sentences from all groups.

A generic summarization technique using the Latent Semantic Analysis (LSA) for identifying the semantic importance of sentences has been proposed.[24] In Singular Valued Decomposition (SVD) based summarization, sentences are represented as high dimensional vectors and are reduced to a sentence matrix of a much smaller dimension. However, the target dimension might not be known. The summarization dataset was the collection of documents from Reuter's collection. The proposed technique was compared with Random baselines (aggregate of 10 random baselines) and a TF-IDF frequency-based summarizer. Cosine similarities between the original text and the produced summaries were used as metrics. Though this work proposed that the reduction in dimension using SVD[25] can be viewed as a semantic reduction of information present in the original text, SVD does not make use of any explicit or implicit semantic information present in the original text. Hence the technique proposed in this work did not exploit all information available for summarization.

Statistical techniques for text summarization typically do not make use of linguistic information in the text. However, statistical models create probabilistic models that indirectly learn the linguistic features present in the text.[26] All term frequency-based approaches that remove stop words from the original text follow Luhn's approach and are the most basic and common summarization techniques. The simplicity of such techniques makes them very attractive in real-world applications for their efficiency. Methods like LSA make use of well-established reduction techniques such as SVD that have been proven across domains and applications as the base and hence yield impressive results. Hence statistical-based text summarization techniques can be used for evaluating other summarization techniques. However, not using rich linguistic information in the text is still a gap that diminishes the chances of producing human-like summaries.

Clustering

A cluster-based conditional Markov random walk (Cluster CMRW) model[27] has been used for MDS. The random walk model was used for evaluating the importance of a vertex based on the global information extracted repeatedly from the graph. The transition model thus constructed, along with the PageRank algorithm[28] is used to construct a sentence scoring function. Then a penalty awarding procedure was followed to limit redundancy amongst the sentences selected. The Cluster CMRW algorithm used the cluster information in the links of the graphs based on link analysis. Finally, the salience scores of the sentences were computed and used for creating the summary. The DUC 2001 and DUC 2002 document summarization datasets were used to evaluate the proposed Cluster CMRW method.

However, as the clusters are randomly formed from the sentences, the quality of themes in such constructed clusters is not guaranteed.[29] DS techniques based on clustering approaches report good summarization results. Hence these approaches can serve as good benchmarks for the DS task. However, such methods are language agnostic and do not exploit the rich linguistic information available in the text. These techniques, by design, do not store and use any knowledge for producing text summaries.

Linguistic Techniques

Word frequency, Code quantity principle, and Text entailment had been used to create extractive summaries of text.[30] Text entailment determines whether a part of a text can be inferred from another text. Text entailment could be used to check whether a concept and content present in a sentence is present in another sentence and can be used to reduce redundancy. The code quantity principle supported the fact that coding elements in a text such as syllables, and phrases and their frequency of occurrences could be used to measure the importance of a code element and the sentences in which the element is present. The sentences that contained code elements with the highest importance were used to construct the summary. DUC 2002 document summarization dataset and five well-known fairy tales and their summaries were used to evaluate this approach. MDS using submodular functions and their budgeted maximization has been implemented.[31] Term frequency – Inverted Sentence Frequency (TF-ISF) with cosine similarity measure was used to compute the similarity between sentences and against the mean vector. This approach used a graph-based representation of input text with sentences and vertices and similarity between sentences as edges. Cosine similarity based on TF-IDF[32] was used as the similarity measure between sentences. To create the summary, a greedy algorithm that used maximization of the sub-modular set function with a budget constraint was employed. A budget was used to limit the length of the summary. Graph cut and redundancy was the submodular functions used to generate the multi-document summary. The DUC 2003 document summarization dataset was used for development and was used to determine the parameters of the created model.

Linguistic approaches attempt to solve the DS task using human understanding and grammar of languages.[33] Hence these approaches have been quite promising. However, other human text processing behaviors, including usage of knowledge and principles of knowledge storage and retrieval are typically not used in these techniques. The review of the linguistic approaches shows that the usage of grammatical and other linguistic elements in the text can yield good results for the DS task.

EVOLUTIONARY METHODS

MDS had been considered as an optimization problem to improve information coverage and reduce redundancy in a summary for a specific summary length. This work identified and counted unique terms in the input text. These word counts were combined to create a mean vector. Then sentences in the input text are compared with this mean vector.[34] Sentences more similar to this mean vector were assumed to be rich in information content and were included in the summary. Similarity evaluation between the input text sentences was used to remove redundancy in the constructed summary.

The summary content coverage and summary diversity have been optimized using the differential algorithm for MDS.[35] The summarization problem was defined as a discrete optimization problem. Each sentence was depicted using a real value and then converted to a binary value for differential evolution computation. An adaptive crossover rate with a fixed number of iterations of the algorithm was used in this approach.

Each agent in this approach represented a collection of sentences. An agent is a binary vector with several dimensions that equal the total number of sentences in the input text. The binary value represents whether a sentence is included or not in the agent. Sentences were represented using the vector space model.[36] This work used the position of sentences, the relationship of sentence to title, length of sentences, cohesion between sentences in summary, and the information coverage of sentences in summary as features in the fitness function evaluation. Cohesion was defined as the similarity between sentences in the summary (an individual) and information coverage was defined as the similarity between the summary sentences and all sentences in the input document. The cosine similarity was used to find the similarity between sentences. In the selection step, parents were selected based on Rank selection and the Roulette wheel-based approach. The first new individual was created using a point cross-over and allowed to exist only if its sentences met the summary limit criterion. The second new individual was created using the same process, but with the roles of the parents reversed. The multi-bit mutation was used as the mutation operator. A guided local search was used with sentences as characteristics. The cost of an individual was defined as a combination of its position in the original text, its similarity to the title, and the maximum similarity of the title with all sentences in the original document. A penalty for the characteristic was defined based on its cost and its previous penalty. The fitness of a characteristic was diminished based on its penalty. A restricted competition approach was used for replacement of individuals and finally, convergence was achieved by selecting characteristics that are closer to the overall mean fitness of the population. The number of iterations was used as the stopping criterion.

The literature survey on evolutionary approaches for DS suggests that sentence similarity measures play an important role in DS.[37] In addition, the DUC conference text summarization datasets and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics are frequently used for the evaluation of DS techniques. Most research literature on document summarization refers to and compares the evolutionary approaches of text summarization. Though evolutionary approaches for DS used sentence location and similarity predominantly, these approaches typically do not make use of linguistic features in sentences. In addition, the evolutionary approaches are very dissimilar to the human text processing behavior and hence may never be able to achieve results very close to the human summaries.

GRAPH-BASED TECHNIQUES

Another document summarization technique constructed event graphs based on the event information in the original input text.[38] This method used event-constructed event graphs that were event-based document models. Sentences in the given input text were analyzed and the event anchors (event verbs) in the verb phrases and noun phrases of the text were identified by a pre-trained classifier. These event graphs contained the connected, sentence-level event cues that were identified from the original input text. Logical regression models that were built from the Time Bank corpus were used for event detection and temporal detection. An incoming query was matched against each document's event graph structure and higher-ranked results based on similarity measures were returned.[39]

Lex Page Rank[40] is a widely used, prevalent MDS technique that assesses the chance of a sentence being included in the summary based on the centrality of the sentence. The centrality of a sentence is, in turn, based on the centrality of the words in that sentence.

Centrality in this context was defined similarly to the prestige concept in web document links and social networks. Sentences that are similar to a large number of other sentences in the input text were designated as prestigious sentences. Sentences were represented using the Bag of Words model and each sentence was a vector in this model. TF-IDF-based cosine similarity was used to measure the similarity between sentences. A cosine similarity matrix was constructed based on the cosine similarity of every pair of sentences in the original document. This matrix represented a sentence cluster. From this sentence cluster, the degree centrality of a sentence was computed by removing smaller values in the cosine similarity matrix, by assuming sentences as nodes, and by undirected links between the sentence pairs (nodes) that have higher cosine similarity values.

In addition, TextRank[41] also associates links with weights. A text unit in TextRank can be a word, a sentence, or a paragraph. Similarly, the links can represent any of the various semantic relationships between sentences. Given an input text for DS, TextRank identifies the text units and the links (relations) between them. Then TextRank constructs the graphical representation of the text using the identified text units and the links. The links in TextRank represent values computed using semantic overlap between words in the connected sentences (nodes) as a similarity measure and the results are normalized. TextRank then applies the PageRank algorithm on the graph to identify the important nodes (sentences) in the graphical representation. The nodes are sorted based on their importance and the top-ranked sentences are used to construct the summary.

The summarization techniques that use graphs for creating the summary typically use the concept of the importance of a node (usually a sentence) in a graph that represents the set of all sentences in the input text.[42] Even algorithms that use some linguistic features in the text do not intelligently exploit all possible linguistic features available in the sentences. The literature review on the graph-based techniques for DS suggests that sentence connectivity between sentences in the input text must be used for creating a good summary. In addition, the review also suggests that there are possible avenues of improvement based on better usage of linguistic features that exist in the text. Hence, an intelligent approach rather than a mechanical approach may provide better results. This discovery naturally leads us to explore intelligent (human-like) cognitive model-based approaches for DS.

DS Using Neural Networks

Summa Runner[43] is a Recurrent Neural Network (RNN) based sequence classifier for creating extractive summaries. This RNN used GRU units and had two bidirectional layers. It was trained using abstractive summaries. Each sentence in the document was evaluated and a binary decision about whether to include the sentence in the extractive summary was made. The first layer in this RNN-GRU network[44] learned the sequence of words in the normal and reverse directions. The second layer obtained word-level learning as input and encoded the sentence-level representations using bidirectional sequence. In the second pass, a logistic layer decided to include or exclude a sentence from the extractive summary. As obtaining extractive summaries for large datasets is cumbersome and laborious, this work used the ROUGE score computed based on comparing the available abstractive summaries with the sentences from the original text that were added one by one to the extractive summary.

Deep Q-Networks (DQN)[45] have been used for extractive text summarization. In this approach, each sentence was considered as a potential action. The DQN was used to compute the future reward potential of each such action. The highest valued action was considered as a candidate sentence for the extractive summary. The Q-value for an action (sentence) was determined based on the information content, its salience, and its redundancy with the sentences selected for summary so far. The CNN-RNN and RNN-RNN networks were used to optimize the computed ROUGE metric so that the best summary could be obtained.

Extractive text summaries have been created using the Convolutional Neural Network.[46] The word embedding representations of the input text were given to the input layer of the Convolutional Neural Network (CNN). A convolutional layer was used to analyze phrases with a specific number of words and create phrase-level representations. Sentence representations were formed by the max pooling layer based on the dimensional maximum values from the phrase representations. Representations of contextual words and average sentence representations were used to construct an n-gram model for predicting the next word. This approach was an unsupervised learning model as sentence labels were not used. Plausible next words were differentiated from the noise words using a noise contrastive estimation. Based on sentence similarities, a sentence adjacent graph was built. The prestige of a sentence was computed using the PageRank algorithm. Finally, an objective function based on prestige and diversity along with the sentence representations, their similarities, and a summary length limit was optimized.

Shallow Artificial Neural Networks (ANN) were employed to create summaries based on the identified features.[47] A fuzzy rule set and a fuzzy vector of these features were proposed to perform extractive text summarization. The final finding was that the fuzzy method provided better extractive summaries compared to the shallow ANN-based summarizers.

Artificial Neural Network (ANN) and deep neural networks[48] for text summarization gain knowledge from available data and use such knowledge to create summaries. However, the knowledge in neural networks cannot be interpreted and understood by other systems or humans. Even transfer learning using the learned, available knowledge in neural networks typically does not work unless the learned model and target model features are similar. In addition, changes in domain, type of information, and documents will necessitate re-learning in neural networks. In neural networks, any learned contextual information is also un-interpretable.

Cognitive Models and DS

The system proposed in this work accepted XML or HTML documents as input. A parser was used to analyze the sentences and extract them from the input text. This system used the event index cognitive model indices to summarize the given text content. This system performed sentence extraction and then used a parser to identify the constituent parse trees of the given text. It then used a custom algorithm to resolve anaphora. Then this technique extracted the protagonist, spatial locations, and temporal elements from the text using the grammar of the language and a custom list of names. Then this system clustered sentences based primarily on the protagonist, spatial and temporal indices considering only them as essential for content-based relations. Clusters of sentences were formed based on the indices and sentences were selected based on cluster membership. This process was repeated until a length limit of 100 words till the summary size was met.

The developed system considered the intention index identification process as a very vague process. It did not attempt to identify noun phrases that champion an intention. This system also used parameter values that were not well justified. For example, any cluster that was smaller than one-third of the largest cluster was discarded. There were no reasonable and sufficient justifications provided for such arbitrary actions and arbitrary selection of parameter values. This system could not be used for large and dynamic text sources due to the cumbersome clustering process as it used both vertical indexing and horizontal indexing.

The Construction-Integration (CI) cognitive model has been used for creating text summaries. This specific work recreated the text comprehension model and used proposition as the fundamental text unit for constructing the summary. The proposed systems attempted to create coherent text summaries. It used distributed LSA and online LDA to construct the long-term memory content. This system tried to make use of left and right neighbors of words to construct a context. The proposed system used three types of memory, namely, working memory, episodic memory, and long-term memory. It also used the concept of spreading activation in the memories.

This model did not make use of human emotion, which is a fundamental necessity for storing and retrieving knowledge from memory. In addition, this model just uses the adjacent words to construct the context of a word which is not defensible based on semantics or cognitive psychology models. In addition, this system is quite complex as it extracts propositions from input sentences, and then reverts to identify sentences based on propositions. This processing procedure is not defensible based on the CI model or any other cognitive process in the human text-processing behavior. In addition, the construction integration model does not explicitly list the processes and behavior involved in constructing a mental representation of a summary. Hence, the proposed system developed may not truly reflect human behavior in constructing a mental representation of the read text to create a summary.

Symbolically and numerically represented knowledge using variants of semantic nets has been used to construct text summaries. This system was constructed using the Learning Intelligent Distribution Agent (LIDA) architecture. A perceptive memory that stores words and their synonyms, LIDA codes for text processing and analysis, and sentence and discourse level text analysis were used by the proposed system to identify summary sentences. 30 text documents that are not openly and freely available were used for evaluating the proposed system. A 70% accuracy based on user summaries was reported. The generated and reference summaries were not provided. Since the evaluations of the proposed system were not standard, the results are neither repeatable nor verifiable. Hence, the proposed system may not work with large, dynamic text sources.

Though a few cognitive psychology-based implementations and approaches for DS exist, they are severely limited by their stringent assumptions and/or limited applications. Even these existing techniques for text summarization do not make use of existing human memory models and emotions effectively for knowledge storage and retrieval. To achieve a performance similar to humans in the DS task, the cognitive processes underlying human text comprehension and human text summarization behavior should be leveraged. In addition, such fundamental cognitive processes, for knowledge storage and retrieval, should be effectively integrated with prevalent and robust models of human memory. Such an integrated solution may deliver text understanding and processing behavior and results similar to human text processing behavior.

Development of deep learning algorithm for retrieving the important concepts layer by layer effectively for automatic multi-document summarization. Apply human text understanding and processing behavior and underlying cognitive processes for extractive, Multi-Document Summarization (MDS).

Research Method

Multi-document summarization is an extension of single-document summarization in that it constructs a summary from a cluster of documents as shown in Figure .3. Multi-document summarization has a multitude of applications in real-world scenarios such as search engines and news aggregation. Multi-document summarization has significant, additional challenges compared to single-document summarization (SDS). In both SDS and MDS, an underlying assumption is that each document is reasonably written well. In other words, closely related semantically relevant sentences (thematic consistency), very little redundancy, and correct sentence structures are assumed in individual documents. Hence, a summary obtained by performing SDS over a single document will retain part of the structural consistency factors. However, in MDS, though individual documents may have these attributes, combined text from multiple documents need not have a high level of thematic consistency. In addition, since the multiple documents considered are about the same or similar topics, the combined text is expected to have redundancy. Hence a summary created using the text from many documents under similar topics may not retain the structural consistency of the original single documents. These factors make MDS more challenging than SDS. Hence an MDS system must have anti-redundancy measures while preserving information coverage, recognize and use the temporal dimension cues in the text, and perform rigorous co-reference resolution. This research developed two models for multi-document summarization and examined their performance.

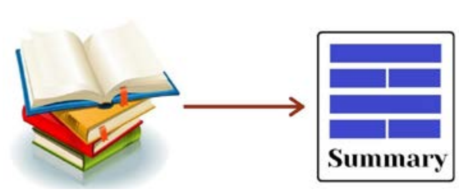


Figure 3. Multi-document summarization

MDS Based On Deep Learning Algorithm

In the proposed method, initially, the input documents perform preprocessing and feature vector extraction. The extracted feature vector is given as input to the fuzzy model and obtains the optimized feature vector. Based on the optimized feature vector, the summary is generated by the proposed system. The proposed approach consists of the following process preprocessing, feature vector extraction, fuzzy model optimized by hybrid genetic Particle Swarm Optimization (PSO), deep learning for summarization, and summary generation as shown in Figure 4.

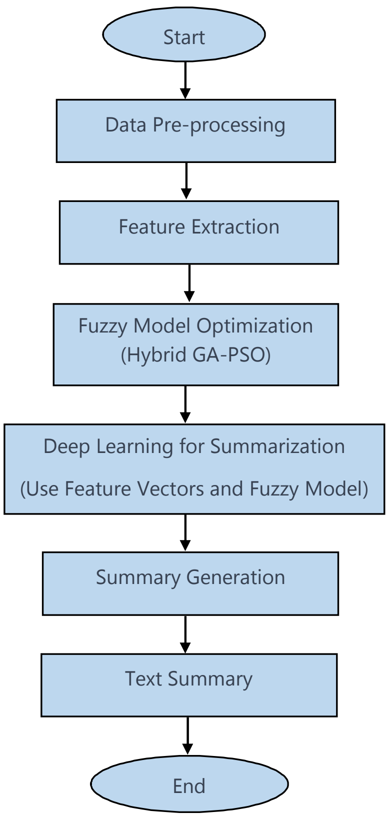


Figure 4. Flow Diagram for Proposed Methodology

Initially, the input documents are subjected to the preprocessing phase. After preprocessing the essential features of the input documents are extracted. This is called as feature vector extraction phase. The fuzzy model based on hybrid genetic PSO used in the proposed approach performs the following processes namely, fuzzifier, fuzzy rule base, inference engine integrated by hybrid genetic PSO, and defuzzifier. The fuzzifier is used to translate the feature vector into linguistic values based on the given membership function which in turn is used as linguistic variables for the input. It is the most important part of the fuzzy model where the IF-THEN rules are defined.

The Inference engine is used to obtain the input values from the fuzzifier which then checks them with the knowledge base to decide the significance of the sentence and to optimize for the getting expected results. So, to reduce the redundancy in the set of rules, an optimization algorithm is used. The optimization algorithm used by the proposed approach is a hybrid of GA and PSO. The generated set of rules is considered as the initial population P to the genetic algorithm as shown in equation (1). The initial population is generated based on the knowledge base. Then the rules are listed as follows

$$P = R_i \tag{1}$$

IF (s_f1 is very high) AND (s_f2 is high) AND (s_f3 is very high) AND (s_f4 is high) AND (s_f5 is very high) AND (s_f6 is very high) AND (s_f7 is very high) THEN (sentence is important)

In this rule, "very high" and "high" are distinct levels of the features that are represented by each condition (s_f1, s_f2, s_f3, s_f4, s_f5, s_f6, s_f7). The rule states that a statement is important if all of these requirements are satisfied concurrently (all features must be "very high" or "high").

According to the generated rule, the numerical values such as 1, 2, 3, 4, and 5 are assigned to Very Low, Low, Medium, High, and Very High respectively. The numeric values corresponding to each rule are extracted from each rule that is generated. The extracted values are then stored in a set Ri. In corresponding to the rule represented in the above example, the value in Ri becomes written in the following equation (2)

$$R_i = [F_1 : 2][F_2 : 4][F_3 : 3][F_4 : 0][F_5 : 0][F_6 : 1][F_7 : 1] \tag{2}$$

F1, F2 …F7 represents the feature vector values s_f1, s_f2 … s_f7 respectively. Since the rule Ri holds the value for F1, F2, F3, F6, and F7 the numerical values for these features are assigned and the values of F4, and F5 are assigned to zero since the rule Ri does not hold feature vector values.

The optimization process starts by initializing the initial population. Thus, according to the basic genetic algorithm procedure, a fitness function is applied to the rules. The fitness function is defined in the following equation (3)

$$Fitness = \sum_{i=1}^{7} wf_i \times F_i \tag{3}$$

Where wf_i is a weighted factor for balancing among the parameters and $\sum wf_i = 1$. Fi represents the feature vector values. The rule that holds the maximum fitness value is highly ranked and so on. Now, the genetic algorithm triggers the selection process on the rules according to their fitness values. Thus, the proposed system finds fitness for all sets of rules present in the initial population. The aim of the genetic algorithm here is to minimize the number of populations by selecting the rules with maximum fitness values. Though the initial population contains rules with minimum fitness value, are cannot be neglected because the initial population is a random selection. Therefore, the initial population needs to be processed through cycles of genetic algorithm.

The further processing related to the rule selector process is the crossover. Here each pair of rules is selected from the set P' and a feature crossing option is applied. According to the fuzzier phase, each rule contains seven features as attributes. Therefore, the crossover operation is applied by selecting a crossover point and exchanging the features between the rules. Let us discuss the crossover operation defined in the proposed approach through an example defined in the following equations (4) and (5).

$$R\ [F_i : 2][F_i : 1][F_i : 3][F_i : 2][F_i : 1][F_i : 1][F_i : 5] \quad i = 1\ 2\ 3\ 4\ 5\ 6\ 7 \tag{4}$$

$$R\ [F_j : 1][F_j : 2][F_j : 1][F_j : 2][F_j : 4][F_j : 4][F_j : 5] \quad j = 1\ 2\ 3\ 4\ 5\ 6\ 7 \tag{5}$$

Therefore, to maintain the feature count in each rule as seven, we adopt a single-point crossover operation. Thus a random feature is selected from both the rules and they are been exchanged. Thus, for the above example, the crossover point is considered as feature three. Similarly, every rule is selected from the set Plocal and is processed for velocity and position update for a single iteration. Now rules are again subjected to the selection process and the steps are repeated until several best rules are left for processing. The rules after the optimization process are stored in a set, which contains the top relevant rules for defuzzification.

The final step of the fuzzy model is the defuzzification of the fuzzy sets generated by the fuzzification. In the defuzzification process, all the linguistic values obtained from the inference engine will be reverted to the crisp values with the help of the fuzzy centroid method using the generalized triangular membership function which is divided into membership functions. After performing the preprocessing and feature vector extraction, the extracted features are optimized by the Fuzzy Model. The optimized feature vector is given as input to the restricted Boltzmann machine. In the summary generation phase, for summary generation first task is obtaining the sentence score for each sentence of the document. After this step ranking of the sentence is performed and the final set of sentences for text summary generation defining the summary is obtained.

Algorithm 1: Text Summarization Based on Human Behavioral Learning Model

Input: Raw Text Document
Step 1: Start by performing data preprocessing on the input images
Step 2: Extract feature vectors from the preprocessed images
Step 3: Fuzzy Model Optimization using Hybrid Genetic PSO
Step 3.1: Initialize the parameters of the fuzzy model.
Step 3.2: Define the fitness function that evaluates the quality of the model's output summaries.
Step 3.3: Use a hybrid Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) approach to optimize the fuzzy model's parameters.
Step 3.4: Iterate through multiple generations, applying genetic operations like selection, crossover, and mutation, as well as PSO swarm updates to find the best model parameters.
Step 3.5: The optimized fuzzy model will be used to assign importance scores to sentences.
Step 3.6: The optimized feature vector is given as input to the restricted Boltzmann machine.
Step 4: Deep Learning for Summarization
Step 4.1: Train a deep learning model (e.g., a neural network) for summarization.
Step 4.2: Use the feature vectors and the optimized fuzzy model to guide the deep learning model.
Step 4.3: The deep learning model learns to generate summaries by considering the importance scores assigned to sentences.
Step 5: Summary Generation
Step 5.1: Apply the trained deep learning model to the preprocessed text.
Step 5.2: Generate a summary by selecting the most important sentences or content based on the learned model.
Step 5.3: Combine the selected sentences to create the final text summary.
Output: Text Summary
End of Algorithm

DS Using Cognitive Learning Model

The Knowledge-Based Event-Index (KB-EI) computational cognitive model combines the event-index cognitive psychology model, hierarchical human memory model, and usage of emotion to store and retrieve declarative knowledge elements from memory. KB-EI model uses this combination to create a multi-document summary from a given text document. When new information is stored in a memory element, the KB-EI model attaches emotion to the element. This emotional component attached to the knowledge element is increased or decreased in strength based on the element's usage under specific contexts. As knowledge elements' chances of retrieval from memory are based on the emotional component attached to them, the KB-EI computational cognitive model is adaptive.

The learning phase in the KB-EI computational cognitive model is further divided into multiple learning episodes. In each learning episode, the KB-EI model analyses a large number of natural language documents to teach the intra-sentential, explicit causality, and intentionality relations that exist in the documents. The overall processing of documents and creation of causality and intentionality relations in episodic and semantic memories by the KB-EI computational cognitive model.

KB-EI model performs topic extraction on the document to identify the existing topics in the document. The set of topics in a document constitutes the context of the document.

$$Context_{document} = \{t | t \in topic\ in\ the\ document\} \quad (6)$$

The topic extraction function of the KB-EI model analyzes a document to determine the main topics or subjects that are covered in the text. The primary concepts or subject areas covered in the document are represented by these extracted topics. The topics that were taken out of the document serve to fundamentally establish its context or main subject matter.

The sentences in the document and the words in the sentences are identified using sentence identification and word extraction techniques. Part of Speech (POS) identification and Named entity recognition (NER) are used to recognize protagonists, and spatial information in the text. KB-EI model uses Time ML and Clear TK-Time ML classifiers and models for identifying and ordering temporal information in the text. Temporal event cues in the text are extracted using the temporal text connectives such as before, after, dates, and temporal propositions such as on, at, and temporal functions.

Synonyms of the verb cause and low ambiguity causal verbs are used to identify the intra-sentential, explicit causal relations. Synonyms of the word intention are used to identify the explicit intentionality relations. The KB-EI model associates every identified intention in the document text with a protagonist. The event-index model indices in the KB-EI model's learning phase are defined similarly to the indices definitions in the CR-GP model. Causality relations and intentionality relations are stored in the episodic memory along with emotion. For causality relations, valence is defined as the combination of the cause in the relation and the context of the document. For a causal relation, CR, valence is computed as shown in Equation 7.

$$Valence_{CR} = Cause_{CR} + Context_{Document} \quad (7)$$

Where $Cause_{CR}$ may refer to a variable or element included in a model's causality relations. It might stand for the cause or causing component in the particular relationship or occurrence. The actual values it accepts will vary depending on the input or data being processed.

$Context_{Document}$ probably refers to information or elements found in the document that gives context. It might contain specifics on the topic, setting, or historical backdrop of the document. The values would vary based on the document's content that was being examined.

A cause-and-effect link within a text's cause-and-effect sequence has an emotional or evaluative component, which can be measured using valence in the context of causality relations ($Valence_{CR}$). In this sense, "valence" refers to the emotional tone or attitude that the cause in the causality relation conveys as well as how it relates to the document's overall context.

Similarly, for intentionality relations, valence is defined as the combination of the protagonist along the context of the document. For an intentional relation, IR valence is computed using

$$Valence_{IR} = Protagonist_{IR} + Context_{Document} \quad (8)$$

Where Protagonist_{IR} values could indicate the identities or characteristics of the primary character or focal point (protagonists) related to intentionality relationships.

Valence in intentionality interactions (Valence_{IR}) is an important component for comprehending the emotional and evaluative elements of intents. It supports the interpretation of human behavior, weighs the moral weight of choices, and shapes how intentions are interpreted in a range of situations, such as storytelling, communication, and dispute resolution.

Arousal for both causality and intentionality relations is defined as the number of times a relation was used during the learning episode. The emotional attribute core affect is constructed using valence and arousal and is attached to the learned relation. Each time a relation is stored or retrieved the arousal attached to the relation is increased. The KB-EI learning phase algorithm processes all documents in the document corpus. Hence the time and space complexity of the algorithm depends on the number of documents in the document corpus. In addition, the number of topics identified in each document, and the number of causal relations and intentional relations present in each document influence the time and space complexity of the algorithm.

Relations that already exist in the semantic memory are further strengthened in their arousal value in their existing, attached core affect components. New, unseen relations are added to the semantic memory along with their attached emotional attributes, core affect components. As a result, the semantic memory contains the knowledge of causal and intentional relations and can be used as the knowledge base for such relations during the summarization phase. This knowledge base is a significant, fundamental enhancement over CR-GP. Hence this variation of the EI computational cognitive model for text summarization is known as the Knowledge-based EI computational cognitive model (KB-EI). The KB-EI algorithm for copying learned causal and intentional relations in a learning episode to semantic memory. The KB-EI semantic memory update algorithm processes all causal and intentional relations stored in the episodic memory during a learning episode. Hence the time and space complexity of this algorithm are dependent on the number of such relations in the episodic memory and the complexity involved in accessing the episodic and semantic memories.

During the summarization phase, the KB-EI computational cognitive model for text summarization creates a summary of a given text document. The overall processing of a multi-document to create its summary by the KB-EI computational cognitive model during the summarization phase. NLP techniques used in the learning phase including sentence identification, word extraction, POS tagging of words, NER, and coreference resolution are performed to extract the linguistic features present in the text. In addition, the explicit causality relations and intentionality relations in the text are extracted in the learning phase. However, in the summarization phase, additional, implicit causality relations and intentionality relations present in the text are also identified. If a cause or effect of a causality relation present in the semantic memory is present in a sentence of the input text, the corresponding effect or cause is searched for. The text document's context, based on the topics present in the document, is used to identify the corresponding effect or cause. If the matching effect or cause is found in another sentence, an implicit causal relation between the sentences is assumed. Hence, causal relations that are not explicitly defined in the input text are identified by the KB-EI computational cognitive model. In addition, the KB-EI model does not assume or require the cause and effect to be present in a sequential manner in the text. Hence KB-EI model can identify any already learnt causal relation present in the input text. If there is more than one possible causal relation that matches the semantic memory for a cause or effect, the causal relation with the highest arousal value is chosen. During a relation search, a partial match of all topics in the document is used to find a similar context in the knowledge base in the semantic memory. Each time a relation is used or retrieved from semantic memory, the arousal value in the associated core effect is incremented.

The KB-EI computational cognitive model for text summarization uses the cognitive processes defined for constructing situation models to understand the text by humans, as illustrated by the EI cognitive psychology model. In addition, it uses the hierarchical human memory architecture and uses emotion, in the form of core affect and its dimensions, to store and retrieve knowledge elements from the knowledge base constructed through learning experiences earlier. Hence, the KB-EI computational cognitive model closely follows the human cognitive processes and structures for reading and understanding a text.

RESULT AND DISCUSSION

DATASET DESCRIPTION

DUC 2002 document summarization dataset contains 59 document sets and an average of 10 documents (NIST 2002b). DUC 2002 document summarization provides both multi-document summaries and single-document summaries. The nature and types of source documents and the structures for single document summaries in the DUC 2002 dataset are similar to the ones in DUC 2001. Each document has two manually created, abstractive, references for each document summaries. Both manual summaries are used for the evaluation of all document summaries created by DS methods evaluated in this research work.

Performance Evaluation

Finally, the salient sentences are extracted based on the sentence score derived by the deep learning algorithm concerning the compression ratio. For experimentation, the summary is generated for different compression ratios and the generated summary is evaluated with the help of evaluation measures such as precision, recall, and f-measure. The computed evaluation measures for compression ratio = 40, 50, and 60 are given in Table 1.

Table 1: Performance of deep learning model for various compressive ratio

Compression ratio	Precision	Recall	F-Measure
40	0.95	0.7595	0.8441
50	1	0.7600	0.8636
60	1	0.7890	0.8821

The deep learning approach has offered 0.95, 0.7595, and 0.8441 for precision, recall and F-measure respectively which is for a lower compression ratio is 40. While comparing with the existing query-oriented deep extraction system, the deep learning algorithm integrated with fuzzy model the value is increased to 1, 0.78, and 0.88 for compression ratio = 60 with a standard deviation range value of 1 as shown in Figure 5.

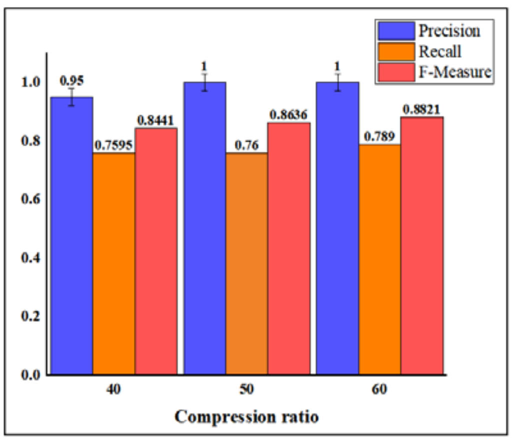


Figure 5. Performance of deep learning model for various compressive ratio

The mean precision and recall values are averages of precision and recall achieved by the various methods across the documents in the DUC 2002 document summarization dataset. KB-EI,[49] LexRank,[50] and TextRank[51] which have been utilized in text summarization selected and worked with the DUC 2002 dataset, and the results obtained are given in Table: 2. KB-EI, LexRank, and TextRank produced the best results in prevalent MDS techniques as shown in Figure 6. However, KB-EI performed significantly better than all other compared techniques and achieved an overall improvement of 18% in mean precision, 20% in mean recall and 19% in F-measure with standard deviation of range is 1.

Table 2: Performance of cognitive model for various learning methods

Cognitive method	Precision	Recall	F-Measure
KB-EI	0.1774	0.2025	0.1869
LexRank	0.1437	0.1652	0.152
TextRank	0.1419	0.1682	0.1524

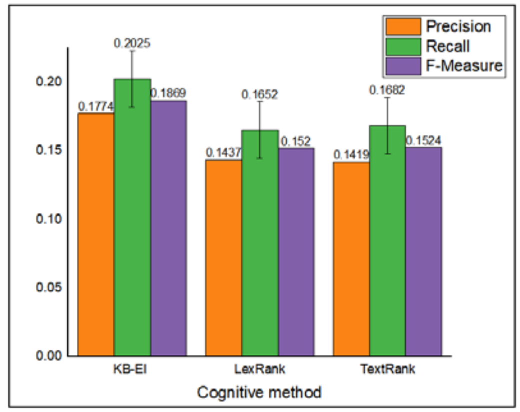


Figure 6. Performance of cognitive model for various learning methods

Comparative Analysis

The comparative analysis given in Table 3 was performed to evaluate the performance superiority of both summarization models. The results clearly showed that both the model performances are similar. Yet, deep learning performs higher than the cognitive model as shown in Figure7 due to its optimization and advances. However cognitive models have a better possibility to attain better results in the future with required updations.

Table 3: Comparative performance of deep learning and cognitive model

Method used for MSD	Precision	Recall	F-Measure
Deep learning model	1	0.7890	0.8821
Cognitive model	0.7	0.8025	0.1869

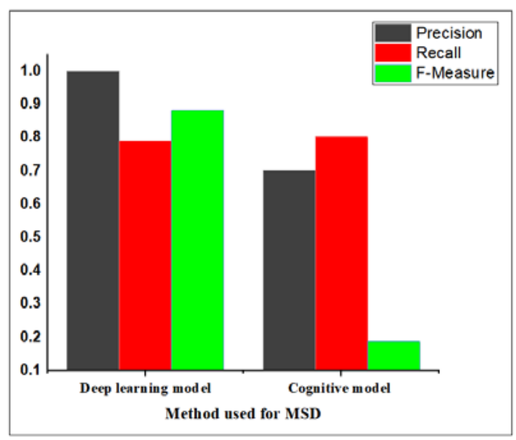


Figure 7. Comparative performance of deep learning and cognitive model

The effectiveness of the designed computational cognitive models has been analyzed based on the quality of summaries created by these models using standard document summarization datasets and evaluation metrics. The validity and consistency of the obtained results have been verified using statistical measures. These experimental results have proved that the document summarization models designed in this research work achieve significantly better results than prevalent document summarization systems. The effectiveness of the computational cognitive models for document summarization designed in this research work makes them applicable in varied scenarios and highly relevant to current, real-world applications.

Table 4. Comparative results of deep learning with traditional machine-learning models.

Model	Recall	Precision	F-Measure
Logistic Regression	0.42	0.34	0.36
AdaBoost	0.49	0.48	0.46
Gradient Boost	0.50	0.59	0.45
ANN	0.29	0.08	0.13
kNN	0.35	0.36	0.37
Deep learning model	0.7890	1	0.8821

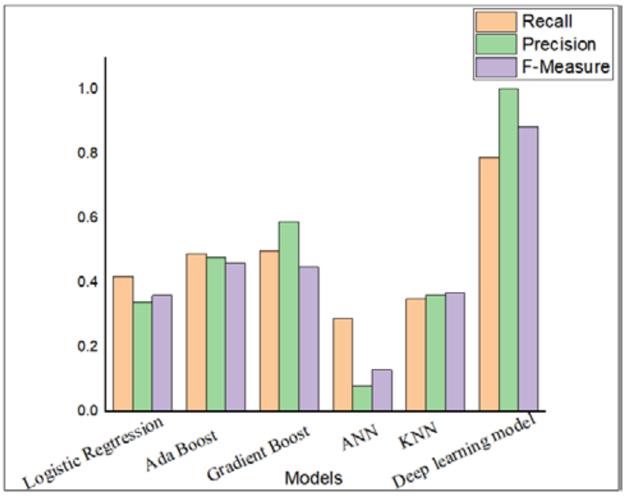


Figure 8. Comparative results of deep learning with traditional machine-learning models

The results of this study set out to measure how well the proposed deep-learning model performed in comparison to other machine-learning models. Table 4 shows that the proposed deep learning model is compared with traditional machine learning algorithms like Logistic Regression, Ada-Boost (Adaptive Boosting), Gradient Boost, ANN (Artificial Neural Network), and KNN (K-Nearest Neighbor) in terms of precision, recall, and F-measure values. The comparison result shows that the proposed model attained the highest performance than the traditional machine learning algorithms by achieving the highest precision value of 1, the highest recall value of 0.7890, and the highest F-measure value of 0.8821 due to its capacity to automatically extract complex features, handle longer documents effectively, and leverage pre-trained language models for improved performance.

Discussion

In today's environment, which is so data-rich, the design of an autonomous text summarizer is crucial. Providing readers with a succinct overview of each document would ease the suffering people experience while reading massive volumes of data. Based on human-created summaries, we have designed an automated text summarizer. Based on the rules established from an optimized deep-learning algorithm and a computational cognitive model, it provides a summary. The outcomes of the experiment show that our strategy is workable. Even the optimized deep learning method yielded acceptable performance. The assignment was effectively performed by two algorithms we took into consideration, and their performance was superior to that of the traditional machine learning methods. Particularly, the proposed optimized deep learning algorithm in our system outperformed the alternative methods.

Furthermore, deep learning models can be enhanced on particular summarising tasks, enabling transfer learning from previously learned language models. The requirement for huge labeled datasets, which may be expensive and time-consuming to produce, is considerably decreased as a result. Compared to traditional machine learning models, fine-tuning a pre-trained model on a particular summarization task frequently results in higher performance because deep learning models have previously acquired complex language representations from a large amount of text. In conclusion, deep learning performs at text summarization because of its ability to automatically extract intricate features, efficiently handle larger texts, and take advantage of already-trained language models for enhanced performance. These benefits make deep learning an effective and flexible method for summarising text, helping to provide more precise and contextually relevant summaries for a variety of applications.

Conclusion

Document summarization has become a significant information retrieval problem due to the explosion of textual information. Document summary is useful in a wide range of real-world applications. Although many document summarization techniques exist, they do not accurately reflect human summarization behavior. Human-created summaries like deep learning and computational cognitive models are used as recent summaries in most standard document summarization datasets showing that human text summarization behavior is superior to existing document summarization techniques. This research work has explored the human text summarization behavior and the underlying processes to design a comparative computational study of both summarization techniques. Based on the precision, recall, and F-measure it is clear that deep learning models are currently working efficiently with required optimization. Also, the computational cognitive model has better efficiency but requires more focus in the future to improve their applicability in text summarization.

Acknowledgments

Authors acknowledges supoort from MET's Institute of Engineering for providing necessary facilities for this study.

Conflict of Interest Statement

Authors declare that there is not CoI for publication of this work.

References and Notes

1. S. Peng, L. Cao, Y. Zhou, Z. Ouyang, A. Yang, X. Li, & S. Yu. A survey on deep learning for textual emotion analysis in social networks. *Digital Communications, and Networks* **2022**, 8(5), 745-762.
2. J.S. Rha, & H.H. Lee, Research trends in digital transformation in the service sector: a review based on network text analysis, *Service Business* **2022**, 16(1), 77-98.
3. H.Y. Koh, J. Ju, M. Liu, & S. Pan. An Empirical Survey on Long Document Summarization: Datasets, Models, and Metrics. *ACM computing surveys* **2022**, 55(8), 1-35.
4. B. Pang, E. Nijkamp, W. Kryściński, S. Savarese, Y. Zhou, & C. Xiong. Long document summarization with top-down and bottom-up inference. *arXiv preprint arXiv:2203.07586* **2022**.

5. G. Kaur, A. Sharma. A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis. *J. Big Data* **2023**, 10(1), 1-23.
6. J.M. Sanchez-Gomez, M.A. Vega-Rodríguez, & C.J. Pérez, A new multi-objective evolutionary algorithm for citation-based summarization: Comprehensive analysis of the generated summaries. *Engineering Applications of Artificial Intelligence* **2023**, 119, 105757.
7. Q. Chen, R. Leaman, A. Allot, L. Luo, C.H. Wei, S. Yan, & Z. Lu, Artificial intelligence in action: addressing the COVID-19 pandemic with natural language processing. *Ann. Rev. biomedical data science* **2021**, 4, 313-339.
8. F. You, S. Zhao, & J. Chen. A topic information fusion and semantic relevance for text summarization. *IEEE Access* **2020**, 8, 178946-178953.
9. E. Durmus, H. He, & M. Diab, FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. *arXiv preprint arXiv:2005.03754* **2020**.
10. T. Shi, Y. Keneshloo, N. Ramakrishnan, & C.K. Reddy. Neural abstractive text summarization with sequence-to-sequence models. *ACM Transactions on Data Science* **2021**, 2(1), 1-37.
11. S. Ghodratnama, A. Beheshti, M. Zakershahrak, & F. Sobhanmanesh. Extractive document summarization based on dynamic feature space mapping. *IEEE Access* **2020**, 8, 139084-139095.
12. D.G. Ghalandari, C. Hokamp, N.T. Pham, J. Glover, & G. Ifrim. A large-scale multi-document summarization dataset from the Wikipedia current events portal. *arXiv preprint arXiv:2005.10070* **2020**.
13. R.R. Kumar, M.B. Reddy, & P. Praveen. Text classification performance analysis on machine learning. *Int. J. Adv. Sci. Technol.* **2019**, 28(20), 691-97.
14. N. Hollenstein, M. Barrett, M. Troendle, F. Bigiolli, N. Langer, & C. Zhang. Advancing NLP with cognitive language processing signals. *arXiv preprint arXiv:1904.02682* **2019**.
15. J. Annis, I. Gauthier, & T.J. Palmeri. Combining convolutional neural networks and cognitive models to predict novel object recognition in humans. *J. Exp. Psychology: Learning, Memory, Cognition* **2021**, 47(5), 785.
16. L. Van Maanen, & S. Miletić. The interpretation of behavior-model correlations in unidentified cognitive models. *Psychonomic Bull. Rev.* **2021**, 28, 374-383.
17. S. Song, H. Huang, & T. Ruan. Abstractive text summarization using LSTM-CNN based deep learning. *Multimedia Tools Appl.* **2019**, 78, 857-875.
18. T. Shi, Y. Keneshloo, N. Ramakrishnan, & C.K. Reddy. Neural abstractive text summarization with sequence-to-sequence models. *ACM Transactions on Data Science* **2021**, 2(1), 1-37.
19. X. Zhang, F. Wei, & M. Zhou. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. *arXiv preprint arXiv:1905.06566* **2019**.
20. Q. Ruan, M. Ostendorff, & G. Rehm. Histruct: Improving extractive text summarization with hierarchical structure information. *arXiv preprint arXiv:2203.09629* **2022**.
21. S. Saha, S. Zhang, P. Hase, & M. Bansal. Summarization programs: Interpretable abstractive summarization with neural modular trees. *arXiv preprint arXiv:2209.10492* **2022**.
22. A. Ghadimi, & H. Beigy. Hybrid multi-document summarization using pre-trained language models. *Expert Syst. Appl.* **2022**, 192, 116292.
23. S. Xu, X. Zhang, Y. Wu, & F. Wei. Sequence level contrastive learning for text summarization. *In Proceedings of the AAAI Conference on Artificial Intelligence* **2022**, 36(10), 11556-11565.
24. A. Jain, A. Arora, J. Morato, D. Yadav, & K.V. Kumar. Automatic text summarization for Hindi using real coded genetic algorithm. *Applied Sciences* **2022**, 12(13), 6584.
25. M. Kornilova, V. Kovalnogov, R. Fedorov, M. Zamaleev, V.N. Katsikis, S.D. Mourtas, & T.E. Simos. Zeroing neural network for pseudoinversion of an arbitrary time-varying matrix based on singular value decomposition. *Mathematics* **2022**, 10(8), 1208.
26. A.P. Widyassari, S. Rustad, G.F. Shidik, E. Noersasongko, A. Syukur, & A. Affandy. Review of automatic text summarization techniques & methods. *J. King Saud Uni.-Comp. Inform. Sci.* **2022**, 34(4), 1029-1046.
27. N. Alharbe, M.A. Rakrouki, A. Aljohani, & M. Khayyat. A New Text Summarization Approach based on Relative Entropy and Document Decomposition. *Int. J. Adv. Comp. Sci. Appl.* **2022**, 13(3).
28. S.S.S. Reddy, C.D. Kumar, S.R. Pisati, & R.D. Kolli. Document Summarization Model Using Modified Pagerank Algorithm. *In Artificial Intelligence and Data Science: First International Conference, ICAIDS 2021, Hyderabad, India, December 17–18, 2021,* **2022**, 430-439.
29. P. Verma, A. Verma, & S. Pal. An approach for extractive text summarization using fuzzy evolutionary and clustering algorithms. *Appl. Soft Comp.* **2022**, 120, 108670.
30. W.S. El-Kassas, C.R. Salama, A.A. Rafea, H.K. Mohamed, Automatic text summarization: A comprehensive survey, *Expert syst. appl.* **2021**, 165, 113679.
31. A. Kumar, S. Bhattamishra, M. Bhandari, & P. Talukdar, Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation, *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* 1, 3609-3619.
32. L.C. Pei, & S.H. Huspi. Extractive Generic-Based Summarization of Multiple Biomedical Documents Using Hybrid TF-IDF Algorithm and Cosine Similarity Method. *Academia of Information Computing Research* **2020**, 1(2).
33. A. Aries, & W.K. Hidouci. Automatic text summarization: What has been done and what has to be done. arXiv preprint arXiv:1904.00688 **2019**.
34. S. Mandal, G.K. Singh, & A. Pal. Pso-based text summarization approach using sentiment analysis. *In Computing, Communication, and Signal Processing: Proceedings of ICCASP* **2019**, 845-854.
35. N. Saini, S. Saha, D. Chakraborty, & P. Bhattacharyya. Extractive single document summarization using binary differential evolution: Optimization of different sentence quality measures. *PloS one* **2019**, 14(11), e0223477.
36. W. Chen, K. Ramos, K.N. Mullaguri, & A.S. Wu. Genetic algorithms for extractive summarization. *arXiv preprint arXiv:2105.02365* **2021**.
37. J.M. Sanchez-Gomez, M.A. Vega-Rodríguez, & C.J. Perez. A decomposition-based multi-objective optimization approach for extractive multi-document text summarization. *Appl. Soft Comp.* **2020**, 91, 106231.
38. L. Chen, Z. Gan, Y. Cheng, L. Li, L. Carin, & J. Liu. Graph optimal transport for cross-domain alignment. *In Int. Conf. Machine Learn.* **2020**, 1542-1553.
39. L. Wu, Y. Chen, K. Shen, X. Guo, H. Gao, S. Li, & B. Long. Graph neural networks for natural language processing: A survey. *Foundations, Trends Machine Learning* **2023**, 16(2), 119-328.
40. S. Ullah, & A.A.A. Islam. A framework for extractive text summarization using the semantic graph-based approach. *In Proceedings of the 6th international conference on networking, systems, and security* **2019**, 48-56.
41. J.N. Madhuri, & R.G. Kumar. Extractive text summarization using sentence ranking. *In 2019 Int. conf. data sci. communication (IconDSC)* **2019**, 1-3.
42. C. Mallick, A.K. Das, M. Dutta, A.K. Das, & A. Sarkar. Graph-based text summarization using modified Text Rank. *In Soft Computing in Data Analytics: Proceedings of International Conference on SCDA* **2019**, 137-146.
43. G.L. Sriranga, P. Likitha, B. Meghana, & N. Jayanthi. Efficient Text Summarizer Using Point To Generator Technique.
44. R.A. Dar, & A.D. Dileep. 2020. Multi-Headed Self-Attention-based Hierarchical Model for Extractive Summarization. *In Soft Computing for Problem Solving 2019: Proceedings of SocProS* **2020**, 1, 87-96.
45. B. Omidvar-Tehrani, A. Personnaz, & S. Amer –Yahia. Guided Text-based Item Exploration. *In Proceedings of the 31st ACM International Conference on Information & Knowledge Management* **2022**, 3410-3420.
46. D. Anand, & R. Wagh. Effective deep learning approaches for summarization of legal texts. *Journal of King Saud University-Computer and Information Sciences* **2022**, 34(5), 2141-2150.
47. A.K. Singh, & M. Shashi. Vectorization of text documents for identifying unifiable news articles. *International Journal of Advanced Computer Science and Application*s, **2019**, 10(7).
48. D. Anand, & R. Wagh. 2022. Effective deep learning approaches for summarization of legal texts. *Journal of King Saud University-Computer and Information Sciences* **2019**, 34(5), 2141-2150.
49. M. Rajangam, & C. Annamalai. Extractive document summarization using an adaptive, knowledge-based cognitive model. *Cognitive Systems Research* **2019**, 56, 56-71.
50. B. Elayeb, A. Chouigui, M. Bounhas, & O.B. Khiroun. 2020. Automatic Arabic text summarization using analogical proportions. *Cognitive Computation* 12, 1043-1069.
51. M. Yang, C. Li, Y. Shen, Q. Wu, Z. Zhao, & X. Chen. Hierarchical human-like deep neural networks for abstractive text summarization. *IEEE Transactions on Neural Networks and Learning Systems* **2020**, 32(6), 2744-2757.