International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 6, December 2023, pp. 6663~6672
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i6.pp6663-6672

A hybrid approach for text summarization using semantic latent Dirichlet allocation and sentence concept mapping with transformer

Bharathi Mohan Gurusamy[1], Prasanna Kumar Rengarajan[1], Parthasarathy Srinivasan[2]
[1]Department of Computer Science Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Chennai, India
[2]Oracle SGSISTS, Lehi, United States

Article Info

Article history:

Received Oct 11, 2022
Revised Mar 13, 2023
Accepted Apr 3, 2023

Keywords:

Hybrid model
Semantic latent Dirichlet allocation
Sentence concept mapping
Text summarization
Transformer

ABSTRACT

Automatic text summarization generates a summary that contains sentences reflecting the essential and relevant information of the original documents. Extractive summarization requires semantic understanding, while abstractive summarization requires a better intermediate text representation. This paper proposes a hybrid approach for generating text summaries that combine extractive and abstractive methods. To improve the semantic understanding of the model, we propose two novel extractive methods: semantic latent Dirichlet allocation (semantic LDA) and sentence concept mapping. We then generate an intermediate summary by applying our proposed sentence ranking algorithm over the sentence concept mapping. This intermediate summary is input to a transformer-based abstractive model fine-tuned with a multi-head attention mechanism. Our experimental results demonstrate that the proposed hybrid model generates coherent summaries using the intermediate extractive summary covering semantics. As we increase the concepts and number of words in the summary the rouge scores are improved for precision and F1 scores in our proposed model.

Corresponding Author:

Bharathi Mohan Gurusamy
Department of Computer Science Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham
Chennai, India
Email: g_bharathimohan@ch.amrita.edu

1. INTRODUCTION

Today's world is full of information, mostly from web articles [1]. Users read the articles on the web based on their requirements and need to process the data further. Users need to read one or more articles many times to understand and comprehend the required information. The main goal of a text summarizer is to apply some methods and natural language processing (NLP) to reduce the original data in text documents. When generating a summary, we reduce the content of the original documents without compromising their main concepts [2]. The summary we generate from a large document helps the user to skim the documents, saving them time. Text summarization is a challenging task that has been studied extensively, and the approaches used for this task can be broadly classified into three categories: extractive, abstractive, and hybrid summarizers [1]. Extractive summarization techniques extract information from the original document's content and arrange the sentences to provide a summary. Ranking sentences in a document involves statistical and semantic approaches, which assign a weight to each sentence based on its position in the ordered list.

In contrast, abstractive summarization approaches aim to create a semantic and meaningful summary by generating new sentences that convey essential information from the original document(s) using

natural language generation techniques [2]. These techniques are more widely used than extractive summarization approaches to create concise, informative, and readable summaries. As the name suggests, hybrid summarizers combine extractive and abstractive approaches to leverage their respective strengths [3]. For instance, a hybrid approach may generate an extractive summary using semantic and statistical methods and refine it using an abstractive summarization model. The hybrid method can produce a more coherent and informative summary than either method alone. BART, T5, Marian, and mBART are examples of transformer models commonly used for tasks such as summarization, question answering, and translation [4]. Transformers can accomplish NLP tasks such as sentence and word classification, text generation, text answer extraction, and sentence development.

An abstractive summarizer creates a summary by learning the essential concepts in the original text. It is also called natural language generation-the mostly encoder-decoder neural networks used along with some attention models in abstractive summarization. Abstractive summarization generates a document summary based on its content, using natural language techniques rather than representing the original document in an intermediate format. It is worth noting that hugging face transformer has also been used in abstractive text summarization in recent times [5]. A hybrid summarizer [6] takes advantage of extractive and abstractive approaches, the extractive model initially fed the original text to obtain a summary based on a statistical measure. The summary generated by this approach relies solely on the word count or percentage of the original text. The abstractive model then further refines the initial summary generated by the extractive model. To generate the final summary, the summary obtained from the extractive model serves as input to the abstractive model.

The proposed method takes advantage of two powerful approaches: extractive and abstractive. The input text is first fed into the extractive model to obtain extracted content. The paper used the semantic latent Dirichlet allocation (semantic LDA) approach to find the hidden topics and several concepts in Wikipedia articles. This paper applied a sentence concept mapping strategy to map articles' sentences to different Wikipedia concepts. Since one or more concepts may map onto many sentences, a sentence ranking algorithm retrieves highly ranked sentences from other concepts. The intermediate summary generated from the extractive approach is more semantically related and covers different article concepts. The content is further generalized using the abstractive model. Our experimental results on real-world data show that the proposed hybrid semantic model achieves better competitive results over extractive and abstractive models.

2. LITERATURE REVIEW

Extractive text summary generation using neural networks proposed in recent times; one such approach is BERTSUM [7]. The author proposed a variant of BERT for extractive text summarization to learn complex features. A novel training algorithm, which optimizes the ROUGE matrices through reinforcement learning objectives, improves the performance of summarizers [8]. The algorithm trains the neural network model on convolutional neural network (CNN) and Daily Mail datasets. SummaRuNNer [9] is a simple recurrent network-based sequence classifier that treats extractive summarization as a classification problem. The algorithm processes every sentence in the original document in document order and decides whether it can be part of the final summary. An extractive model using Restricted Boltz Machine [10] was used to enhance the selected features of sentences. Enhanced features help to provide better sentence scores to choose sentences that are part of the summary. The sentences are represented as continuous vectors [11] to make them a semantically aware representation for finding similarities between sentences. The constant vector representation is handy for multi-document summarization. It employs the feed forward neural network by using a set window of words as input and predicting the next term.

Suleiman and Awajan [12] has comprehensively reviewed deep learning-based text summarization approaches, including datasets and evaluation metrics. It helps understand the importance of deep learning in extractive summarization. The methods include a restricted Boltzmann machine (RBM), recurrent neural network (RNN), convolutional neural network (CNN), and variation auto-encoder (VAE). The datasets used for evaluating and training were DUC 2006, DUC 2007, DUC 2002, Daily Mail, SKE, BC3 datasets, and Essex Arabic Summaries Corpus (EASC). ROUGE metrics are more frequently used as the evaluation measure that evaluates most approaches. Neural network model sentence relation-based summarization SRSum [13] learns sentence relation features from data. The model uses five sub models: PriorSum uses a convolutional neural network to understand the sentence's meaning. SFSum models surface information using sentence length and position, CSRSum considers the relation of a sentence with its local context, TSRSum models the semantic relationship between sentences and titles, and QSRSum assigns weights to relevant queries to capture the relation between query and sentence.

Contextual relation-based summarization [13] is another neural network model that learns sentence and context representation. Using a two-level attention mechanism, it retains the similarity scores between

sentences and context. Multiview CNN [14] enhanced version of CNN is used to find the crucial features of sentences. Word embedding is used to train the model and features of sentences learned to rank the sentences. Sentence position embedding is also used to increase the learning capacity of the neural model. Test summarization can be considered classification [15] by a multi-modal RNN model using the sentence-image classification method. It creates a summary of documents with images. The proposed method encodes sentences and words in the RNN model and the image set is encoded with the CNN model. It uses a logistic classifier for selecting sentences based on their probability and sentence-image alignment probability.

A new deep neural network (DNN) model for extractive summarization sentences and words from alternating pointer networks (SWAP-NET) [16] used encoder-decoder architecture for selecting essential sentences. The architecture uses keywords in the selection of sentences. The attention-based mechanism is used to learn important words and sentences. CNN/DM [17] used the approach for dividing the training set based on its domain. CNN/DM achieves significant improvement in training the neural network BERT. The author explored constituent and style factors to analyze their effect on the generalization of neural summarization models. They examined how different model architectures; pre-training strategies react to datasets. Some combined supervised learning with unsupervised learning to measure the importance of a sentence in the document [18]. The author used three methods: the first used a graph and a supervised model separately and then combined them to assign a score to the sentence. The second method evaluated the importance of sentences by using the graph model as an independent feature of the supervised model. The third model used a priori value to the graph model to score the sentences using a supervised approach.

Multi-document summarization using deep learning architecture as a hybrid model [19] generates comprehensive summaries from news articles on specific topics. The architecture performed better than the traditional extractive model when evaluated using DUC 2004 data. Extracting the gist of documents is possible by using information such as titles, image captions, and side headings [20]. The author has proposed a single-document summarizer framework with a hierarchical document encoder with attention to side information. The extractive summarization framework with side information generates a better summary with fluency. Another framework matches extracted summary with the original document in semantic space [21] and models sentence relationships. It also provides a deep analysis of the gap between summary-level and sentence-level extractors based on the features of a dataset.

One of the main driving forces in recent development in abstractive text summarization is the availability of new neural architectures and new strategies in training. However, there is a need to address issues such as a proper model and data analysis tool and understanding the failure model of summarization. SummVis [22], an open-source tool, allows us to visualize, generate a summary, and analyze the summarization models and the evaluation metrics used. Topic modeling has been recently used in text summarization to identify hidden topics in the document [23]. Latent Dirichlet allocation (LDA) performs better than latent semantic analysis (LSA) if the number of features increases in the sentences. A hybrid approach for text summarization [24] proposed a novel sentence scoring method for extractive summarization. The sentence scoring parameter significantly improves the performance of the model. The researchers presented a single-document text summarization technique based on sentence similarity and document context [25]. Their approach utilized undirected graph-based scoring to evaluate sentences and determine which ones should be included in the summary. Extractive text summarization [26] considers sentence length, position, cue phrases, and cohesion when selecting sentences for summarization. In recent years, the use of neural networks for text summarization has become widespread, as these models can learn sentence patterns.

Mostly used deep learning method is the recursive neural network (RNN) [27]. Long short-term memory (LSTM), gated recurrent units (GRU), and transformers were other approaches for solving gradient disappearance. The extractive method has given results since it can easily combine many techniques and improve performance. Using content attention, two-phase multi-document summarization [28] extracts subtopics from documents. The summary was formulated using different sub-topics as an optimization problem of minimizing the sentence distance. Huang *et al.* [29] employed Hepos, a novel encoder-decoder attention, to extract features from original documents. They also introduced a new dataset called GovReport, which includes lengthy documents and their corresponding summaries. The evaluation model for text summarization has its shortcoming in using neural networks [30]. The author has tried to overcome the shortcomings of evaluation metrics of text summarization in five dimensions. He re-evaluated the metrics using neural networks and benchmarked metrics using recent text summarization models. Sentence regression [31] identifies essential features that represent the sentences. The sentence relation such as contextual sentence relations, query sentence relations, and title sentence relations are used to extract basic sentences from the documents.

Training a neural network for text summarization has some difficulty processing large documents. We can replace phrases with general terms in semantic content generalization. Some used the pointer generator network [32], copying the original content and combining the semantic content generalization.

There are many attention-based mechanisms to generate an article summary; one such method is attentive encoder-based summarization (AES) combined with unidirectional recurrent neural network (Uni-AES) [33]. Mohan *et al.* [34], [35] also compared the performance of bidirectional recurrent neural network (Bi-AES) with Uni-AES and experiment results showed that Bi-AES shows better results than Uni-AES.

3. METHOD

3.1. Dataset

The WikiHow dataset is a publicly available dataset [36] comprising articles and their corresponding summaries collected from the WikiHow website. The dataset includes over 200,000 articles with their respective summaries, making it one of the most extensive datasets available for text summarization research. Each article in the WikiHow dataset consists of a title, a summary, a list of steps or instructions, and an optional image. The summaries in the dataset are written by the authors of the articles and are intended to provide a brief overview of the article's content.

3.2. Hybrid model

The proposed system has built a hybrid text summarization model combining the best features of an extractive and an abstractive summarizer. Figure 1 shows the overall architecture diagram for out proposed methodology. First, the system has extracted data from Wikipedia prepossessed and given it as input to the extractive summarizer. Then it needs to identify related concepts of the Wikipedia page, and then it has to apply LDA to know the topics present in the articles by suing semantic LDA. The model will perform sentence concept to find the map each sentence with concepts identified in the semantic LDA process. The *K* concepts are chosen and sentence are ranking using our sentence ranking algorithm and top *N* sentences are chosen as part of the intermediate extractive summary. The intermediate extractive summary given as input to our T5 transformer which is fined tuned to produce the abstractive summary.

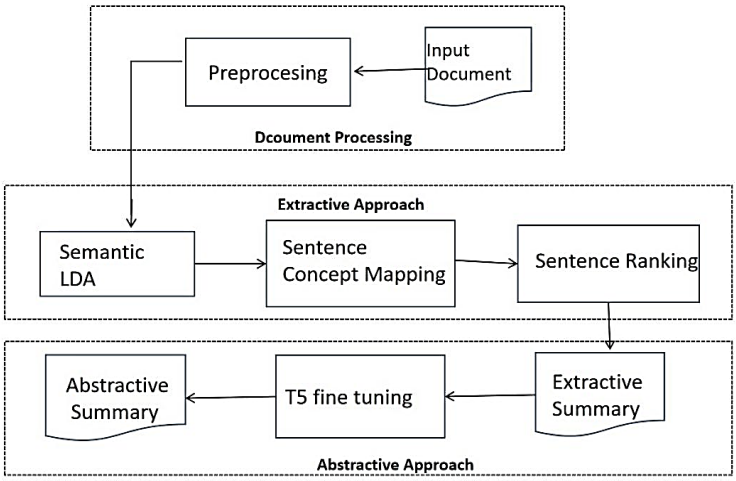

Figure 1. Hybrid approach for text summarization

The semantic LDA is the probabilistic model tries to learn the distributions of the topic using two distribution parameters. Those two parameters are the word distribution parameter *w* and the document distribution parameter *d* utilizing the expression. The algorithm 1 for embedding semantics LDA is as:

Algorithm 1. Semantic LDA

1. Let the number of topics K
2. for every document $D(d1, d2, \ldots dn\}$.
3. for each word w in Document D
Assign randomly one of the topics.
Represents topic of all n documents and distribution of word of all the K topics
Calculate the probability of words reflecting a topic $p(K|D)$
Calculate the probability of words w assigned to K topics in D as $p(w|K)$
Adjust the association of the topic to w with $p(K|D) * p(w|K)$.
4. Until convergence

3.2.1. Extractive text summarization

In the extractive approach, the system used the genism python library, which extracts semantic topics from input documents. Gensim is an open-source vector space and topic modeling toolkit that implements the TextRank algorithm. Text summarization using genism uses a summarizer based on the text rank algorithm. TextRank algorithm ranks sentences by constructing a graph model. It builds a graph representation of text using keyword extraction and sentence extraction. Since the TexkRank algorithm is better suited for sentence extraction, it will rank sentences considering each sentence as vertex and edge as the relationship between sentences. The summarizer can produce a summary based on word count and the ratio of summary based on the original document.

The input document is represented as *D*; the text document is parsed into sentences and mapped to Wikipedia concepts. The sentences are prepossessed and described as queries; based on the query, Wikipedia articles are extracted and represented as concepts. After obtaining *D*'s sentence-concept mapping, the system has to find some sentence overlap across one or more wiki concepts. The model represents sentence concept mapping as two sets of nodes, one as a set of document sentences and another as wiki concepts. It can be viewed as a bipartite graph, where the edge between the document sentence node and wiki concept node represents a sentence-concept mapping. The sentences are ranked based on sentence-concept mapping, and sentences mapped to more concepts are selected first. The sentence is ranked based on the decreasing order of their mapping degree. The sentence with a ranking of more than some *k* is chosen to be part of the summary. The value of *K* is changed from 2 to 5 and captures the summary generated from our model. Algorithm 2 shows the algorithm of sentence ranking.

Algorithm 2. Sentence ranking
Input: Sentence-Concept (S-C) mapping
Output: Sentence/concept Score and Sentence Ranks
1. Initialize rank of Sentence (s_i)=a; concept (c_i)=b
2. loop until convergence (k=1...10)
3. Compute b as the sum of sentences belonging to the set
4. Compute an as the sum of all concepts belonging to the set
5. Normalize a
6. end loop
7. Rank sentences in descending order of scores r=desc (a)

3.2.2. Abstractive summarization

The output of an extractive summarizer using genism extracts sentences and forms the top *k* ranked sentence as a summary Figure 2. The final summary depends upon the word count and the ratio parameter is passed in the summarizer. The output of the extractive summarizer is given as input to the more abstract summary model. Here the paper used an abstractive summarizer based on hugging face transformers. That produces a summary using entirely different text. The model creates new sentences in a new form using transformers. Hugging face transformers provide many pre-trained models for major NLP tasks, including text summarization, classification, machine translation, text generation, and chatbot. The most straightforward implementation involves using transformers as a pipeline application programming interface (API) in the summary model. Summarization is given a task to the pipeline to download model architecture, weights, and token level configurations.

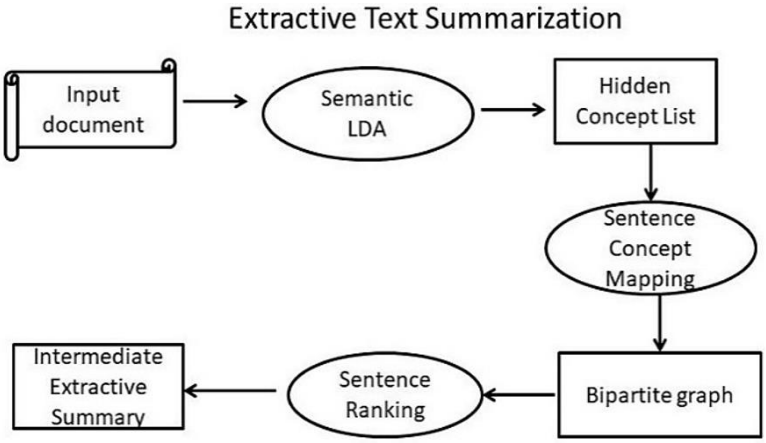

Figure 2. Extractive text summarization

The proposed system configured the model to generate a summary using the T5 transformer model and its tokenizer, shown in Figure 3. First, the model needs to encode the text to tensors of integers using tokenizers. Transformers are also used for machine translation apart from text summarization. So, it can be used to convert text from English to German by specifying the parameters in the model. Many parameters are passed to generate the model: *max_length* for determining the number of tokens, *min_length* specifies a minimum number of tokens to generate, and *length_penalty* is used to increase the size of the output. Another parameter system that can change is *num_beams*, used to set the model for beam search. The last parameter is *early_stoppping*, set to True to make generation finish when the beam search reaches the end of the string token.

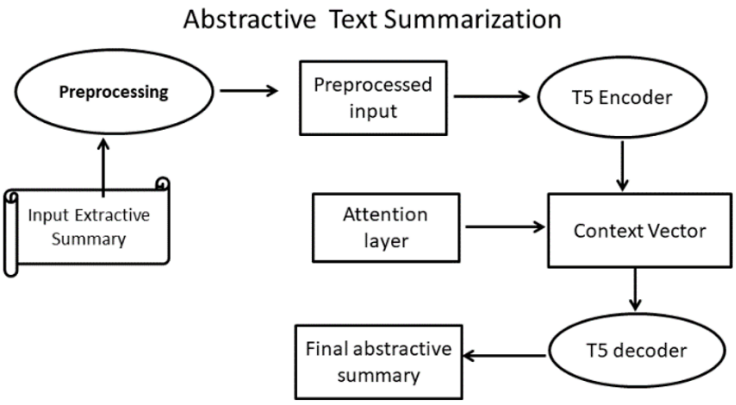

Figure 3. Abstractive summarizer using T5 transformer

3.2.3. Transformer model

Any transformer model can be loaded and trained through a single API, providing ease of work, flexibility, and simplicity. The pipeline used in transformers is grouped into three categories: preprocessing, input to model, and post-processing. Since transformers cannot process raw text, the model must first convert text into numbers. The model needs a tokenizer for identifying tokens; each token is mapped to an integer. Transformers have some pre-trained models, such as base modules, where the output will be a hidden state called features for a given input. The production of the base module will be significant with three dimensions; where the first one is batch size represents the number of sentences processed. The second dimension represents the length of a sequence, sequence length. The last dimension is hidden size: the length of hidden features or vectors. Model process the high-dimension features to different and lower dimensions for text summarization. The output from the transformer model is processed by the model head using architecture for text summarization. The model head can be configured for summary length regarding the number of words or percentage concerning the original summary.

4. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed system built a hybrid text summarizer model where the output of the extractive summarizer will be input for the abstractive summarizer. For the input, the model considered the Wikipedia content for text summarization. The model initially used the Genism model for summary generation. Gensim is available as a python package and uses a text rank algorithm. NLP preprocessing is applied over the extracted Wikipedia content using a package as a pipeline trained on web text such as blogs, news, and comments. The proposed algorithm semantic LDA is applied as topic modeling to identify the hidden topics in the articles. The sentence is ranked based on the concept mapping of sentences. The summarizer method uses sentence concept mapping in the summary model. The generated extractive summary using our model has been compared with state-of-the-art text summarization methods, such as Seq-to-Seq with an attention mechanism and the TextRank-based summary model. The performance of our extractive summarization model shows better results when using the longest common subsequence, as shown in Table 1.

The summary can also be generated by specifying the desired number of concepts, *K*, to be present in the generated summary. The resulting summary will be an extractive summary ranked based on statistical approaches, such as the number of concepts covered and semantic approaches, such as the number of topics covered in the document. The output summary will consist of the top-ranked sentences. Table 2 shows the average Rouge-1 and Rouge-2 scores for different sentence concept mapping *K* numbers. The sentence which

captures most of the concepts gives better results across all the summarizers. Our proposed approach for extracted summarization using topic modeling and sentence -concepts mapping shows better results in overall rouge scores.

Table 1. Comparison of performance of summarization models on WikiHow dataset

	Models		
Metrics	Seq-to-seq with attention	TextRank	Semantic LDA-based extractive summarizes
Rouge-1	22.04	27.53	27.10
Rouge-2	6.27	7.4	6.98
Rouge-L	20.87	20.00	25.34

Table 2. Average Rouge 1 and Rouge-2 scores for concepts *K*

Summarizer	K=2		K=3		K=5	
	Rouge 1	Rouge-2	Rouge 1	Rouge-2	Rouge 1	Rouge-2
First few sentences	0.45	0.22	0.46	0.22	0.48	0.24
Random sentences	0.41	0.15	0.44	0.17	0.45	0.19
Best sentences	0.51	0.29	0.49	0.28	0.5	0.26
Proposed (Wiki concept)	0.46	0.23	0.49	0.23	0.51	0.28
Frequency-based	0.47	0.23	0.46	0.21	0.45	0.2

Table 3 shows the performance of our model compared with the existing model as a BERT extractive summary, graph-based extractive model. The performance of our model outperforms the related model as it effectively identifies the semantically related topics from the document. The output of the extractive summarizer is given as input to the next-level abstractive summarizer. The proposed hybrid model builds an abstractive summarizer using hugging face transformers. Transformers are pipelined to process of extraction of features from the input text. The summarizer model is made using a T5 transformer and there, we can set the length of the summary. The summary generated using the T5 hugging face model is compared with the summary generated from the extractive summary approach. The Rouge metrics were used to evaluate the model in terms of F-measure, Precision, and Recall. The experiment results show that the performance of the hybrid approach is better than the extractive approach.

The performance of the hybrid model is evaluated using ROUGE metrics. Figure 4 shows that as we increase the number of words in the end summary, the ROUGE metrics precision improves better, although F-measure and recall remain more or less the same. The sample articles from WikiHow dataset with human annotated summary was shown in Table 4. We tested our hybrid model with a few random samples and evaluated the performance of our model in each approach using rouge scores. The summary generated by our model in extractive and abstractive is given in the Table 5.

Table 3. Comparison of performance of summarization models on the DUC2002 dataset

	Models		
Metrics	BERT based extractor and LSTM pointer network	Topic modeling based weighted graph representation	Proposed model topic modeled using semantic LDA
Rouge-1	43.39%	48.10%	48.35%
Rouge-2	19.38%	23.30%	29.53%
Rouge-L	40.14%	NA	41.72%

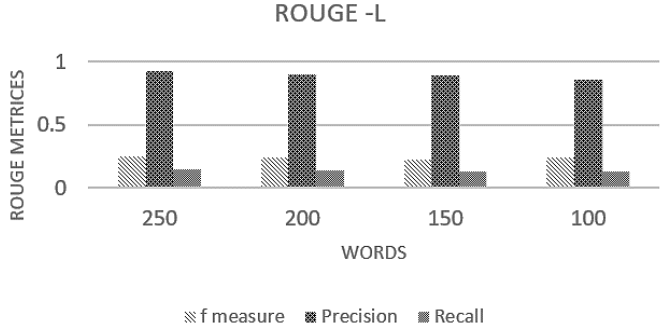

Figure 4. ROUGE-1 vs. words in summary

Table 4. Sample of a WikiHow article along with its corresponding human-annotated summary

Sample Article
Title: How to Change Your Name Introduction: Changing your name is a big decision, but sometimes it is right. You may choose to change your name for personal reasons, because you have recently married, divorced, or changed your gender, or simply because you do not like the name you were given. Whatever your reason, you can change your name legally and fairly easily, depending on where you live. Human-annotated summary**:** Changing your name can be a big decision, but it is legally and fairly easy to do. You need to understand the reasons why you want to change your name and check your state's specific requirements. Once you have filled out the necessary paperwork, you must submit it to the court and wait for the court to approve your request. Once your name has been legally changed, you will need to update your personal information on legal documents such as your driver's license, Social Security card, passport, and any other legal documents.

Table 5. Output summary for sample article shown on Table 4

Extractive summary	Abstractive summary
Extractive summary: changing your name can be done legally and fairly easily, but it is important to understand your state's specific requirements. You will need to fill out the necessary paperwork, submit it to the court, and wait for approval. Once your name has been legally changed, you must update your personal information on all legal documents.	Changing your name can be a daunting decision, but it can also be a necessary one for personal or legal reasons. The process varies by state, but typically involves filling out paperwork and waiting for court approval. Once your name is legally changed, you will need to update all relevant documents. It is important to understand the process and requirements before beginning.

5. CONCLUSION AND FUTURE WORK

The need for text summarization is indeed an automatic choice for many web-based readers. Text summarization is classified as extractive and abstractive summarization. The extractive text summarizers used statistical features in the documents to generate summaries. Semantic summaries make interesting to readers. The semantics summaries are generated by applying a deep learning approach. This paper introduces a hybrid model that combines the best features of both extractive and abstractive text summarization. Using semantic LDA and sentence concept mapping algorithms, our hybrid model is first trained to generate an extractive summary over the WikiHow dataset. semantic LDA is used to identify the hidden topics in the document, and sentence concept mapping is used to map different concepts in the articles. Then, our proposed system configures the pipeline of hugging face transformers to generate an abstractive summary from the extracted summary. Our experimental results show that our model's performance is better compared to extractive summarization alone and the precision improves as we increase the number of concepts and words in the article. In the future, one has to focus on improving the performance of the summary model by enhancing semantic features.

REFERENCES

[1] "Total number of websites," *Internet Live Stats*. https://www.internetlivestats.com/total-number-of-websites/ (accessed Sep. 05, 2022).

[2] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: a comprehensive survey," *Expert Systems with Applications*, vol. 165, Mar. 2021, doi: 10.1016/j.eswa.2020.113679.

[3] N. Moratanch and S. Chitrakala, "A survey on abstractive text summarization," in *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, Mar. 2016, pp. 1–7, doi: 10.1109/ICCPCT.2016.7530193.

[4] T. Ma, Q. Pan, H. Rong, Y. Qian, Y. Tian, and N. Al-Nabhan, "T-BERTSum: topic-aware text summarization based on BERT," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 3, pp. 879–890, Jun. 2022, doi: 10.1109/TCSS.2021.3088506.

[5] T. Wolf *et al.*, "Transformers: state-of-the-art natural language processing," in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 2020, pp. 38–45.

[6] C. Ma, W. E. Zhang, M. Guo, H. Wang, and Q. Z. Sheng, "Multi-document summarization via deep learning techniques: a survey," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–37, May 2023, doi: 10.1145/3529754.

[7] Y. Liu, "Fine-tune BERT for extractive summarization," *arXiv preprint arXiv:1903.10318*, Mar. 2019.

[8] S. Narayan, S. B. Cohen, and M. Lapata, "Ranking sentences for extractive summarization with reinforcement learning," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1747–1759, doi: 10.18653/v1/N18-1158.

[9] R. Nallapati, F. Zhai, and B. Zhou, "SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Feb. 2017, vol. 31, no. 1, doi: 10.1609/aaai.v31i1.10958.

[10] S. Verma and V. Nidhi, "Extractive summarization using deep learning," *Research in Computing Science*, vol. 147, no. 10, pp. 107–117, Dec. 2018, doi: 10.13053/rcs-147-10-9.

[11] W. Xiao and G. Carenini, "Extractive summarization of long documents by combining global and local context," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3009–3019, doi: 10.18653/v1/D19-1298.

[12] D. Suleiman and A. A. Awajan, "Deep learning based extractive text summarization: approaches, datasets and evaluation measures," in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, Oct. 2019, pp. 204–210, doi: 10.1109/SNAMS.2019.8931813.

[13] P. Ren, Z. Chen, Z. Ren, F. Wei, J. Ma, and M. de Rijke, "Leveraging contextual sentence relations for extractive summarization using a neural attention model," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Aug. 2017, pp. 95–104, doi: 10.1145/3077136.3080792.
[14] Y. Zhang, M. J. Er, R. Zhao, and M. Pratama, "Multiview convolutional neural networks for multidocument extractive summarization," *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3230–3242, 2017, doi: 10.1109/TCYB.2016.2628402.
[15] J. Chen and H. Zhuge, "Extractive summarization of documents with images based on multi-modal RNN," *Future Generation Computer Systems*, vol. 99, pp. 186–196, Oct. 2019, doi: 10.1016/j.future.2019.04.045.
[16] A. Jadhav and V. Rajan, "Extractive summarization with SWAP-NET: sentences and words from alternating pointer networks," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 142–151, doi: 10.18653/v1/P18-1014.
[17] M. Zhong, D. Wang, P. Liu, X. Qiu, and X. Huang, "A closer look at data bias in neural extractive summarization models," in *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, 2019, pp. 80–89, doi: 10.18653/v1/D19-5410.
[18] X. Mao, H. Yang, S. Huang, Y. Liu, and R. Li, "Extractive summarization using supervised and unsupervised learning," *Expert Systems with Applications*, vol. 133, pp. 173–181, Nov. 2019, doi: 10.1016/j.eswa.2019.05.011.
[19] A. K. Singh and M. Shashi, "Deep learning architecture for multi-document summarization as a cascade of abstractive and extractive summarization approaches," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 3, pp. 950–954, Mar. 2019, doi: 10.26438/ijcse/v7i3.950954.
[20] S. Narayan, N. Papasarantopoulos, S. B. Cohen, and M. Lapata, "Neural extractive summarization with side information," *arXiv preprint arXiv:1704.04530*, Apr. 2017.
[21] M. Zhong, P. Liu, Y. Chen, D. Wang, X. Qiu, and X. Huang, "Extractive summarization as text matching," in *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6197–6208, doi: 10.18653/v1/2020.acl-main.552.
[22] J. Vig, W. Kryscinski, K. Goel, and N. Rajani, "SummVis: interactive visual analysis of models, data, and evaluation for text summarization," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, 2021, pp. 150–158, doi: 10.18653/v1/2021.acl-demo.18.
[23] G. B. Mohan and R. P. Kumar, "A comprehensive survey on topic modeling in text summarization," in *Micro-Electronics and Telecommunication Engineering*, Springer Nature Singapore, 2022, pp. 231–240.
[24] C. Jyothi and M. Supriya, "Abstractive text summarization on templatized data," in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer International Publishing, 2021, pp. 225–239.
[25] S. S. Rani, K. Sreejith, and A. Sanker, "A hybrid approach for automatic document summarization," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sep. 2017, pp. 663–669, doi: 10.1109/ICACCI.2017.8125917.
[26] M. Kirmani, N. M. Hakak, M. Mohd, and M. Mohd, "Hybrid text summarization: a survey," in *Advances in Intelligent Systems and Computing*, Springer Singapore, 2019, pp. 63–73.
[27] S. Turky, A. A. AL-Jumaili, and R. Hasoun, "Deep learning based on different methods for text summary: a survey," *Journal of Al-Qadisiyah for Computer Science and Mathematics*, vol. 13, no. 1, pp. 26–35, 2021.
[28] L. Dong, M. N. Satpute, W. Wu, and D.-Z. Du, "Two-phase multidocument summarization through content-attention-based subtopic detection," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 6, pp. 1379–1392, Dec. 2021, doi: 10.1109/TCSS.2021.3079206.
[29] L. Huang, S. Cao, N. Parulian, H. Ji, and L. Wang, "Efficient attentions for long document summarization," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 1419–1436, doi: 10.18653/v1/2021.naacl-main.112.
[30] A. R. Fabbri, W. Kryściński, B. McCann, C. Xiong, R. Socher, and D. Radev, "SummEval: Re-evaluating summarization evaluation," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 391–409, Apr. 2021, doi: 10.1162/tacl_a_00373.
[31] P. Ren *et al.*, "Sentence relations for extractive summarization with deep neural networks," *ACM Transactions on Information Systems*, vol. 36, no. 4, pp. 1–32, Oct. 2018, doi: 10.1145/3200864.
[32] Y. Wu and K. Wakabayashi, "Effect of semantic content generalization on pointer generator network in text summarization," in *Proceedings of the 22nd International Conference on Information Integration and Web-based Applications and Services*, Nov. 2020, pp. 72–76, doi: 10.1145/3428757.3429118.
[33] C. Feng, F. Cai, H. Chen, and M. de Rijke, "Attentive encoder-based extractive text summarization," in *Proc. of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 1499–1502, doi: 10.1145/3269206.3269251.
[34] G. B. Mohan and R. P. Kumar, "Lattice abstraction-based content summarization using baseline abstractive lexical chaining progress," *International Journal of Information Technology*, vol. 15, no. 1, pp. 369–378, 2023, doi: 10.1007/s41870-022-01080-y.
[35] G. B. Mohan and R. P. Kumar, "Survey of text document summarization based on ensemble topic vector clustering model," in *IoT Based Control Networks and Intelligent Systems*, Springer Nature Singapore, 2023, pp. 831–847.
[36] M. Koupaee and W. Y. Wang, "Wikihow: a large scale text summarization dataset," *arXiv preprint arXiv:1810.09305*, 2018.

BIOGRAPHIES OF AUTHORS

Bharathi Mohan Gurusamy pursuing Ph.D. degree in Amrita School of Engineering Amrita Vishwa Vidyapeetham Chennai, Tamil Nadu, India. He has 15 years of teaching experience and 3 experience of research experience. He is currently working as an Assistant Professor with the Amrita School of Computing, Amrita Vishwa Vidyapeetham University, and Chennai, India. Since 2021 he has been publishing scientific papers in text summarization. His research interests include natural language processing, computational intelligence, machine learning, and deep learning. He can be contacted at email: g_bharathimohan@ch.amrita.edu.

Prasanna Kumar Rengarajan received his Ph.D. degree from Anna University, Chennai, Tamil Nādu, and India. He is currently working as Chairperson and Associate Professor in Amrita School of Computing, Amrita University, India. He has 20 years of experience in teaching. Since 2010 he has published scientific papers in data mining, data analysis, time series analysis and computer vision. His areas of interest include data analytics, machine learning, theory of computation, compiler design, and python programming. He can be contacted at email: r_prasannakumar@ch.amrita.edu.

Parthasarathy Srinivasan currently working as the Senior Technical Analyst at oracle, Lehi USA. He has more than 20 years' experience in database management system and contributed to niche areas of software Artifacts by researching and leveraging the knowledge to procedure optimal solutions. He can be contacted at email: parthasarathy.s.srinivasan@oracle.com.