Deep Learning based Abstractive text Summarization: A Survey

G Uday Kiran
Department of CSE(AI & ML)
B V Raju Institute of Technology
Narsapur, India
udaykiran.goru@bvrit.ac.in

Ramakrishna Gandi
Department of Computer Science & Engineering
Madanapalle Institute of Technology & Science, India
gandiramakrishna2@gmail.com

M. Lavanya
Department of CSE(Data Science)
Vignana Bharathi Institute of Technology
Hyderabad, India
lavanyareddy.1982@gmail.com

Girish Bhagwant Desale
Department of Computer Science & IT
JET'S Z. B. Patil College,
DHULE. (M.S.)
girishdesale@gmail.com

Ch. Upendar Rao
Department of Information Technology
MLR Institute of Technology
Hyderabad, India
upendarcse@gmail.com

B. Veerasekhar Reddy
Department of Information Technology
MLR Institute of Technology
Hyderabad, India
Bhargavisekhar68@gmail.com

2024 Parul International Conference on Engineering and Technology (PICET) | 979-8-3503-6974-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/PICET60765.2024.10716176

***Abstract*—The explosion in the amount of textual data in the last several years has created an abundant source of resources for data extraction and analysis. It is necessary to summarize this data in order to access relevant information in a fair amount of time. Here we take a look at some of the more current methods for summarizing abstractive texts with deep learning models. A hotspot for research in the field of Natural Language Processing (NLP) is summarization, the process of condensing a document without losing any of its significance. You can classify summarization methods as either "extractive" or "abstractive" depending on whether they use natural language processing to construct new phrases or keep the exact ones from the source text. Extensive research has been conducted on extractive summarization, and the issue has now attained maturity. At this point, the focus of the study is on providing an abstract summary. Abstractive Text Summarization (ATS) is a tough and complicated operation because of the inherent complexity of the natural language content. Researchers can get a thorough overview of Deep Learning based ATS in this work. This paper provides a comprehensive survey of deep learning models employed for abstractive summarization, delving into their architectures and methodologies. Furthermore, it scrutinizes prevalent datasets utilized for training and evaluation purposes, offering insights into their characteristics and suitability for benchmarking. The study evaluates the performance of various abstractive summarization methods on these datasets, shedding light on their effectiveness and limitations. Lastly, emerging themes in the field are discussed, along with unresolved challenges, aiming to foster deeper understanding and advancement in DL-based abstractive text summarization among researchers.**

Keywords*— *Text Summarization, Natural Language Processing, Deep learning, abstractive summarization.

I. INTRODUCTION

The amount of textual data stored in cloud resources including WebPages, blogs, news, user communications, and social network platforms is growing at an exponential rate in this digital age. There is a wealth of textual material in a variety of archives, including novels, books, scientific papers, legal records, biomedical documents and articles. Consequently, the problem of information overload is escalating. Users lose a lot of productivity every day because they have to spend so much time reading cumbersome texts and removing unnecessary material [1-2]. There is a great need for text summarization services because of the abundance of accessible textual data, such as online documents, articles, news, and reviews. Text summarization is crucial for a number of reasons, such as speedily retrieving important information from lengthy texts, solving issues with evaluation criteria, and making the most important parts of long texts easy to load. There is a need to analyze and summarize artificial text summarizing methods because of their growth and evolution, which have produced notable results in many languages [3].

Web applications that rely on text summary include news websites and search engines. To aid in the recovery of knowledge, search engines create previews in the form of snippets, while news websites create headlines that characterize the news. Function, genre, summary context, summarizer type, and document count are some of the factors that classify text summarization into different groups. There is a method for classifying text summarization that distinguishes between extractive and abstractive steps [4].

Structured and semantic-based approaches are examples of methods for abstract text summarization. In contrast to structured approaches, which use schemas like ontology, tree, body phrases and also template or rule-based schemas to encode documents' essential features, semantic-based approaches focus on the text's meaning and use its information representation to summarize it. Among the many methods that rely on semantics are the information item method, the multimodal semantic technique, and the semantic graph-based method [5].

The first use of deep learning techniques for Abstractive text summarization was in 2015, with an encoder-decoder architecture-based model proposed. Deep learning methods have been widely used for these purposes due to their great performance in recent years [6]. There were three criteria used to categorize summarization tasks: input, purpose and

979-8-3503-6974-8/24/$31.00 ©2024 IEEE

output. Both Mahajani and Dong et al. examined a mere 5 models of abstract summarization. In contrast, Mahajani et al. [7] examined the architecture of multiple abstractive summarization models as well as their datasets and training methods. There was no discussion, however, of the resulting summary's quality with regard to the various methods or evaluation metrics.

In order to construct a new group of phrases that summarize the original text, abstractive summarization focuses on the most important information. This method requires picking out important details, figuring out what they mean in context, and then recreating them in a different way. The difficulty of automatically producing coherent prose and the difficulty of extracting pertinent information from documents have made abstractive summarization a more difficult task than extractive summarization. There is a tremendous and ever-increasing amount of digital textual data in the modern technological age. A time-saving and effective method of dealing with long text material is made possible by automatic summarization systems. The goal of these approaches is to produce concise summaries that capture all the important points about a topic without sacrificing clarity or thoroughness. Some common uses for text summarization include snippets produced by search engines as a result of document searches and headlines generated by news websites to facilitate surfing.

II. LITERATURE REVIEW

Deep learning simplifies decision-making by analyzing complicated problems. By using feature extraction at varying degrees of abstraction, deep learning makes an effort to mimic the capabilities of the human brain. In general, lower-level layers tend to have more details than higher-level ones. After receiving information from the input layer, the output layer will transform it nonlinearly in order to provide an output. Deep learning's hierarchical structure facilitates learning. Since the output of one layer is used as an input by the next, the degree of abstraction increases linearly with the level of abstraction of the preceding layer. [8]. In addition, the number of layers determines the depth, which in turn effects the degree of learning. Several problems in natural language processing make use of deep learning because it enables the learning of multilevel hierarchical representations of data using numerous data processing layers of nonlinear units. Abstractive summarization has made use of a number of deep learning models, such as RNNs, CNNs, and sequence to sequence models. In this section, we will go into deeper detail on deep learning models.

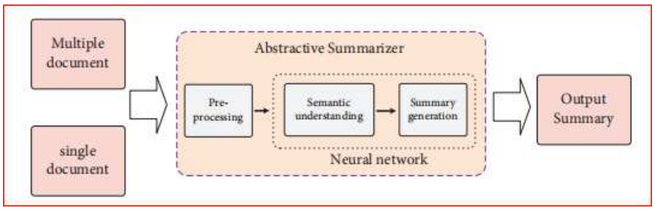

Fig 1. Architecture of Deep Learning based Aabstractive Summarizer

A three-stage procedure, including pre-processing, semantic comprehension, and summary creation, makes up DL-based ABS, as shown in Figure 1. In the pre-processing stage, linguistic techniques like as division of sentences, tokenization of words and removal of stop words etc., are commonly used to organize the input text. In the next stage, neural networks are trained to identify and capture deep semantics for the given input data using vector space. Finally, a fusion vector is produced. The generator then uses the vocabulary to create summary terms by transferring the vector space representation. The fusion vector supplied in the preceding phase is subsequently fine-tuned..

A. *Deep Neural Networks:*

The backbone of deep learning, which employs complex mathematical techniques to train different models, are deep neural networks (DNNs). A multi-layer perceptron (MLP) is another name for it because of the number of hidden layers it has. Graph neural networks (GNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) are some of the DNNs that are frequently employed in ABS.

B. *Recurrent Neural Network:*

The idea behind RNN stems from the commonly held belief that "human's cognition is based on experience and memory." The operation of RNN relies on the interdependence of neighboring items and the existence of a linear relationship in the sequence.The input features from the previous and current time steps are used by the e-network to anticipate the output to the next step. In RNN, the hidden layer nodes are linked to one another.The output produced by the input layer plus previous hidden layer make up the input of the hidden layer [9]. Using RNN to process sequential data is a great idea. In data, it can extract semantic and temporal information.Thus, RNN based DL models have achieved significant strides in addressing several difficult natural language processing (NLP) issues, including recommender systems,IE,, text summarization , time analysis, machine translation.

C. *Convolutional Neural Network:*

One type of deep feed forward neural network is convolutional neural network (CNN)[10]. Three dimensions—depth, breadth, and height—are used to arrange the neurons in a convolutional neural network (CNN). Instead of being totally coupled, neurons in various layers are now just linked in very localized regions. In order to deal with inputs of varied sizes, CNN have most distinctive features—sparse interactions, equivariant representations, parameter sharing are employed. Three structures make up a basic convolutional neural network (CNN): activation, pooling, and convolution. Using the concept of maximum pooling features are extracted at regular intervals, CNN can acquire features of varying levels of complexity, from simple to sophisticated, by employing the convolution kernel to extract features from the data item.Through the use of a convolution filter and pooling processes, it is possible to minimize the parameters, simplify the input matrix, and extract its most salient features. M convolutional layers followed by b pooling layers make up one of the convolution blocks. Convolutional neural networks (CNNs) consist of N sequentially stackable convolutional blocks and K final fully connected layers.

D. Graph Neural Network:

Graph-specific neural networks, or GNNs [11], are a kind of neural network. Nodes are embedded according to their local neighbors, which is a fundamental principle of GNN. A neural network, as it seems at first glance, aggregates the properties of each node and the nodes linked to it. Presently, GNNs can be broadly classified into four types: GCNs, GANs, GGNNs, and GGNs.

When it comes to document categorization and document-based language models (LM), hierarchical neural models have proven to be very effective. A simple hierarchical ABS model was put out by Li et al.[12] in 2015, and further refined by Jadhav and Rajan [13]. Not only that, but their method produces much more informative and readable summaries than competing methods. Drawing inspiration from graph-based natural language processing methods, Tan et al.[14] introduced a new graph based attention technique in the hierarchical encoder decoder architecture. Using word encoder forward encoding and a sentence encoder for short sentence encoding, they built a hidden state network using the hidden states of the sentences. The sentence's hierarchical attention value can be found using a concealed state graph.

The most representative ATS model that relies solely on CNN was proposed by Gehring et al. [15] in 2017 with the model ConvS2S. Both the encoder and the decoder in this model employ CNN. The model takes word embeddings as input and also includes position vectors for each input token in the input layer. Then, the final word embeddings are formed by concatenating the word and position embeddings. This allows CNN based models to understand word order similarly to RNN and use convolution module to perform nonlinear transformations and convolutions on the embeddings. Furthermore, they implemented residual connections between layers to address the issue of gradient disappearance and explosion. On the DUC-2004 and Gigaword datasets, their approach produces results comparable to RNN based models, while training at a significantly faster pace.

To regulate the generated summary's design and satisfy user customization requests, Fan et al.[16] presented a model that may determine the summary's length, style, and entities, among other high-level features. Their model's encoder and decoder are built using CNN. They took their cue from Gehring et al. [15] and expanded intra-attention to include many hops. In addition, they made use of the decoder's self-attention mechanism to make use of the decoded data from before. Using discrete bins to quantify summary length was the first step in controlling the generated summary's length. Then, they added specific word types to the input vocabulary and trained with a marker to show how long the ground-truth summary was.

An extreme ATS system was built by Narayan et al. [17] with the purpose of producing a one sentence title that addresses the query "What is the article about?" They used convolutional neural networks (CNNs) for both the encoder and decoder in the model. While the decoder manages the prediction of each token, the convolutional encoder captures whether or not it represents the document's salient information by associating it with a topic embedding. As an extra input for the encoder and decoder, they used the LDA topic model to extract word and document topic embeddings.

A model was built by Gulcehre et al. [18] to process out-of-view words (OOV words) using an attention-based pointing mechanism. Two softmax layers were used by their model to forecast the next words that would be generated: one layer to forecast the word's location in the given sentence and replicate the output, and the other is to forecast word's position in the vocabulary list that was shortlisted. They choose the softmax to utilize for word generation in each prediction phase using Multilayer Perceptron (MLP). Simultaneously, a big vocabulary technique (LVT) is implemented, which streamlines decoding by decreasing the size of the decoder side's softmax layer. They drew inspiration from a well-known phenomenon in human psychology: when faced with a name that they do not know, individuals often resort to making assumptions about it based on its context and history. The challenge of producing OOV terms is greatly reduced by their approach.

Based on the encoder decoder architecture, Gu et al. [19] presented a novel ATS model (CopyNet) that integrates copying procedure to the decoding procedure. This model does a good job of integrating the decoder's standard word generation process with a enhanced copy mechanism that can pick out phrases and words from the input text and insert them into the right places in the summary that is generated. In particular, they tested their models on simulated and actual datasets, and the outcomes proved that their algorithms successfully reduced the OOV word problem.

Using a multi-task learning approach, Li et al. [20] added textual entailment to the ABS task. In particular, their model is built on top of the attention-based encoder, decoder framework. Using the NLI dataset for training, they construct an entailment relationship classifier by combining the ATS model's encoder with a softmax layer. This way, they can share the encoder with the entailment recognition system. Encoders are able to understand the entailment relationship and the source document's substance in this way. Along with using a Reward Augmented Maximum Likelihood (RAML) for training the model, they also adjusted loss function during decoding for reducing the degree of the produced text summary. This made the decoder entailment conscious.

The encoder decoder are combined using Transformer blocks in the Transformer based encoder decoder model (FASum) described by Zhu et al. [21]. To get information about entities and their relationships from the initial text, they employed the open-source OpenIE program. .Each topic, object, and relation makes up a triple, and these triples represent the extracted knowledge. They formed two undirected edges between subjects and relations and between relations and objects for each triple (subject, relation, object). This is how the input document's knowledge graph—an undirected graph—is constructed by adding edges to all the triples. Each node in the knowledge graph has a unique characteristic that is represented by a feature that is extracted using a graph attention neural network. Lastly, the knowledge of the graph's information is incorporated into the process of decoding to influence the development of text summary by building a cross attention layer on the decoder side.

III. CHALLENGES AND DISCUSSIONS

Many obstacles have been encountered by methods that attempt to summarize texts. While some have been resolved, there are still others that require attention. These difficulties and ways to overcome them are addressed in this section.

E. Unavailability of the Golden Token during Testing:

During training, there are golden tokens (also known as reference summary tokens) that can be used to input prior headline tokens into the decoder. Unfortunately, the golden tokens won't be accessible while testing is underway. as a result, the decoder's subsequent steps can only take the output word that was formed earlier as input. Various approaches have been suggested to resolve this problem, which gets more complicated when dealing with tiny datasets. Take reference as an example; it makes use of the DaD model. At each stage of DaD, a token is used in the training or testing, or the preceding step is used for both training and testing and depending on the outcome of a coin flip. This way, the inputs for the training and testing steps are identical. The ⟨EOS÷ token is always used as the first input to the decoder, and the loss is computed using the same formula.

F. Out Of Vocabulary (OOV) Words:

The main terms in the test data might be uncommon or not seen in the training phase, these words are called out-of-the-word (OOV) words, and they might be a problem during testing. To deal with out-of-word terms, a switching decoder/pointer was used, which used pointers to indicate where the words originally appeared in the source document. You can use the switch on the decoder side to switch between using pointers and generating words. By utilizing the pointer, the decoder will transfer the word from the source to memory once the switch is switched off. To activate the decoder, simply flip the switch; it will then produce a word from the specified vocabulary.

G. Repetition of Sentences and erroneous Data in Summary:

The produced output summary has two problems that need to be thought about: the development of incoherent phrases and the repeating of words. Producing lengthy descriptions using attention based encoder decoder RNN and summarizing lengthy materials are two separate but related difficulties. To combat this, we used the coverage model to generate a coverage vector that aggregated attention from all prior time steps; this allowed us to avoid duplication. Encoder intra temporal attention keeps track of the weights of prior attention for each input token as part of the key attention mechanism, which was used to alleviate repetition in. Additionally, the intra temporal attention makes advantage of the decoder's hidden states at a certain time step.

H. Fake Facts:

An issue with abstractive text summarizing is that it can provide summaries with false facts; in reality, this happens in 30% of the cases. In the case of false facts, the subjects and objects of the predicates might not match. Hence, to address this problem, facts are collected using open information extraction and dependency parsing, such as OpenIE. As a result, we suggested a sequence-to-sequence framework that pays equal attention to both the input text and the description of the extracted facts; this would condition the output summary. In order to implement the suggested method with OpenIE and the dependency parser, Stanford CoreNLP was used. OpenIE makes it easier to extract entities from relations. In addition, copying and coverage methods were employed by the decoder.

I. Other Challenges:

In the dataset for abstractive text summarization, the summary's quality is the primary concern. The news in the CNN/Daily Mail dataset is best summarized in the reference section. Since each highlight stands in for a sentence in the summary, we can easily calculate the total number of sentences is equal to total number of highlights. On sometimes, the items that are highlighted fail to cover all the important aspects of the summary. so much work must be put in before a high-quality dataset can be made available. There is also a lack of a multi-sentence dataset for abstractive summarization in Arabic and other languages. You can pay for a service that summarizes abstractive Arabic material in a single statement. The application of ROUGE for assessment is another concern with abstractive summary. When it comes to extractive summarization, ROUGE delivers acceptable results. Since ROUGE relies on word-for-word matching, it is insufficient for abstractive summarization.

In light of recent developments and assessment outcomes, we have concluded that the BERT pre-trained model is the most promising feature. The transformer based models are of high quality and should produce encouraging outcomes.

IV. CONCLUSION

The tremendous amount of information is available in the Internet has enhanced the importance of text summarization in recent years. There are two main approaches to text summarization: extractive and abstractive. A summary using an abstractive text summarization approach would rewrite the original text to provide a summary with new phrases, whereas an extractive text summarization method would use linguistic and statistical aspects to extract words and phrases from the source text. The methods, datasets, and assessment metrics for abstractive text summarization using deep learning were examined in this study. In addition, we dissected and examined the problems that arose while using different methods. Several findings emerged from the summary of the examined methods. The two most popular deep learning methods were e RNN and attention mechanism. When dealing with the gradient vanishing issue that occurred when using an RNN, some methods used LSTM, while others used a GRU. Abstractive summarization also made use of the sequence-to-sequence paradigm. The New York Times, CNN/Daily Mail, and

Gigaword were among the datasets used. For summarizing single sentences, we used Gigaword, and for summarizing multiple sentences, we used CNN/Daily Mail. The models that used Transformer had the greatest results. The most prevalent problems encountered when summarizing were: a golden token not being available during testing, out-of-the-word terms, inaccurate sentences, repeated summary sentences, and false information. Furthermore, abstractive summarization involves a number of factors that need to be thought about, such as the dataset, assessment metrics, and the quality of the produced summary.

References

[1]. A. Khan, M. A. Gul, M. Zareei et al., "Movie ReviewSummarization Using Supervised Learning and GraphBasedranking Algorithm," Computational intelligence andneuroscience, vol. 2020, Article ID 7526580, 2020.

[2]. G. C. V. Vilca and M. A. S. Cabezudo, "A study of abstractivesummarization using semantic representations and discourse level information," Text, Speech, and Dialogue,pp. 482–490, 2017.

[3]. B. VeeraSekharReddy, K. S. Rao and N. Koppula, "Named Entity Recognition using CRF with Active Learning Algorithm in English Texts," 2022 6th International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India, 2022, pp. 1041-1044.

[4]. K. Shen, P. Hao, and R. Li, "A Compressive Sensing Modelfor Speeding up Text Classification," Computational Intelligence and Neuroscience, Article ID 8879795, 2020.

[5]. Thatha, Venkata&Donepudi, Swapna&Safali, Miriyala& Praveen, s & Nguyen Trong, Tung & Nguyen, Ha. (2023). Security and risk analysis in the cloud with software defined networking architecture. International Journal of Electrical and Computer Engineering (IJECE). 13. 5550. 10.11591/ijece.v13i5.pp5550-5559.

[6]. Bhumireddypalli, VSR, Koppula, SR, Koppula, N. "Enhanced conditional random field-long short-term memory for name entity recognition in English texts", in Concurrency Computation: Practice and Experience. 2023; 35(9):e7640.

[7]. J. Dan and H. Jin, "Text Semantic Classification of LongDiscourses Based on Neural Networks with Improved FocalLoss," Computational Intelligence and Neuroscience, ArticleID 8845362, 2021.

[8]. R. Rzepka, S. Takishita, and K. Araki, "Language Modelbased context augmentation for world knowledge bases," inProceedings of the 34th Annual Conference of the JapaneseSociety for Artificial Intelligence, June 2020.

[9]. J. P. Cheng, L. Dong, and M. Lapata, "Long short-termmemory networks for machine reading," in Proceedings ofthe 2016 Conference on Empirical Methods in Natural Language Processing, pp. 551–561.

[10]. Phalguna Krishna E S, Venkata Nagaraju Thatha, Gowtham Mamidisetti, Srihari Varma Mantena, Phanikanth Chintamaneni, Ramesh Vatambeti, Hybrid deep learning model with enhanced sunflower optimization for flood and earthquake detection, Heliyon, Volume 9, Issue 10, 2023, e21172, ISSN 2405-8440,

[11]. Y. Kim, "Convolutional neural networks for sentence classification," in Proceedings of the 2014 Conference on EmpiricalMethods in Natural Language Processing, pp. 1746–1751,Doha, Qatar, 2014.

[12]. K. Xu, W. Hu, J. Leskovec, and J. Stefanie, "How powerfularegraph neural networks," in Proceedings of the 7th International Conference on Learning Representations, New Orleans,USA, 2019.

[13]. J. Li, M. T. Luong, and D. Jurafsky, "A hierarchical neuralautoencoder for paragraphs and documents," in Proceedingsof the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conferenceon Natural Language Processing of the Asian Federation ofNatural Language Processing, pp. 1106–1115, Beijing, China,2015.

[14]. A. Jadhav and V. Rajan, "Extractive summarization withswap-net: sentences and words from alternating pointernetworks," in Proceedings of the 56th annual meeting of theassociation for computational linguistics, pp. 142–151, Melbourne, Australia, 2018.

[15]. J. Tan, X. Wan, and J. Xiao, "Abstractive document summarization with a graph-based attentional neural model,"inProceedings of the 55th Annual Meeting of the Association forComputational Linguistics, pp. 1171–1181, Vancouver,Canada, 2017.

[16]. J. Gehring, M. Auli, and D. Grangier, "Convolutionalsequence to sequence learning," in Proceedings of InternationalConference on Machine Learning, pp. 1243–1252, Sydney,Australia, 2017.

[17]. A. Fan, D. Grangier, and M. Auli, "Controllable abstractivesummarization," in Proceedings of the 2nd Workshop onNeural Machine Translation and Generation, pp. 45–54,Melbourne, Australia, 2018.

[18]. B. Veerasekhar Reddy, K. Srinivas Rao, K. Neeraja; "Named entity recognition on different languages: A survey". AIP Conference Proceedings 22 May 2023; 2492 (1): 030043.

[19]. Venkata Nagaraju Thatha; A. SudhirBabu; D. Haritha, Privacy-preserving smart contracts for fuzzy WordNet-based document representation and clustering using regularised K-means method. International Journal of Ad Hoc and Ubiquitous Computing, 2022 Vol.40 No.1/2/3, pp.2 – 9.

[20]. VeeraSekharReddy, B., Rao, K.S. &Koppula, N. "An Attention Based Bi-LSTM DenseNet Model for Named Entity Recognition in English Texts", in Wireless Personal Communications 130, 1435–1448 (2023).

[21]. C. Zhu, W. Hinthorn, and R. Xu, "Boosting FactualCorrectness of Abstractive Summarization with KnowledgeGraph," 2020, https://arxiv.org/, Article ID 08612.