A Review: Abstractive Text Summarization Techniques using NLP

Pooja Batra
Amity School of Engineering and Technology
Amity University
Gurgaon, Haryana
pbnagpal@ggn.amity.edu

Sarika Chaudhary
Amity School of Engineering and Technology
Amity University
Gurgaon, Haryana
schaudhary@ggn.amity.edu

Kavya Bhatt
Amity School of Engineering and Technology
Amity University
Gurgaon, Haryana
Kavyabhatt28aug@gmail.com

Saloni Varshney
Amity School of Engineering and Technology
Amity University
Gurgaon, Haryana
salonivarshney572@gmail.com

Srashti Verma
Amity School of Engineering and Technology
Amity University
Gurgaon, Haryana
srashtiverm@gmail.com

***Abstract*—Today's world is getting flooded with an increasing amount of articles and links to choose from. As this data grows, the importance of semantic density does as well. How can one say the most important things in the shortest amount of time? Having a generated summary lets one decide whether they want to deep dive further or not. Conversion of lengthy texts into short and meaningful sentences is the main idea behind text summarization. To achieve this, various algorithms are present. Machine Learning models are trained, first to understand the given document and then create a summary of it. These models achieve this task either by extracting important words out of the document or by creating human-like sentences to form the summary.**

Keywords—Abstractive text summarization, LSTM, RNN, Attention Mechanism

I. Introduction

Text summarization is the process of generating a coherent and accurate summary of the original document by extracting essential information and reducing the length of the source document.. The summary helps in easy and fast information retrieval by retains the gist of the document as mentioned by Dalwadi et al. [1]. According to Arun Krishna Chitturi et al. [2].The text summarization aim's at consolidating the source document into optimized form, preserving the overall idea and information intact. In laymen language conversion of lengthy texts into short and meaningful sentences is the main idea behind text summarization. The need for text summarization is continuously increasing as today's world is getting flooded with a growing amount of articles and links to choose from with the expansion of the internet. Human being tends to read the whole document to develop an understanding of it and generate a summary by keeping the main points in mind. It is getting extremely difficult to obtain the required information from this pool of words and sentences in a short period. Going through all the documents, articles, and different forms of information to manually summarize is extremely time-consuming and exhausting for humans. Summarization helps in saving valuable time and conveys the main essence from which the reader can decide if they want to dig deeper. The first automatic text summarizer came into existence in the 1950s and since then summarization has been enhancing.

Text summarization can be used for various purposes like email summary, reviews of movies, news headlines, outline of student notes, summarize information for businessman and government officials, and businessman, summarize the medical data for doctors, summarize the legal document, novel or book summary helping consumer decide to read it for not and code summarizer as acknowledged by Deepali K. Gaikwad at al. [3].

Removing the important information because of the large addition of it on the internet, as self explanatory briefing will be of more value. Following this there is an immeasurable amount of energy which concerns the age of the already programmed content summary structure which helps to establish the abstracts that follow consequently from the content, web, and the organization messages that are related

978-1-7281-9785-2/20/$31.00 ©2020 IEEE

with their own satellite's substances. This review contains and show, how the swarm intelligence advancements is carried out to accomplished the task of content summarization effectively [4–7].

Text summarization approaches can be broadly divided into extractive and abstractive text summarization.

A. Extractive Text Summarization

An extractive text summarization chooses words, sentences, and paragraphs from the original document on the measure of importance and concatenates these important parts of the document to form a summary. It does not modify the original text. Extractive text summarization uses statistical, linguistic, and heuristic methods to identify and extract important text which was referred by Mehdi Allahyari et al. [8] in their paper. Extractive summarization is easy to achieve but it faces certain problems like ambiguity and miscommunication in summary.

Fig. 1. Extractive Text summarization

B. Abstractive Text Summarization

Abstractive text summarization is similar to the way humans summarize the document. Abstractive text summarization firstly understands the document and then generates the summary introducing new words, sentences, and rephrasing. Abstractive text summarization produces a more relevant and accurate summary with reduced ambiguity. It uses a complex heuristic algorithm. It has a good compression rate and reduces redundancy. Achieving abstractive text summarization is harder than extractive text.

Fig. 2. Abstractive Text summarization

II. Related Work

Rule-based Artificial Intelligence did not do a good job for Abstractive Text Summarization. But recently the new algorithms in Deep Learning gave a boost to how we can handle Abstractive Text Summarization. In one of the research papers by Thomaidou et al. [9], a deep learning-based framework was used which after taking in the landing webpage of the website as input, generated promotional short texts of advertisements. While training a model for this task. The main aim of the model is to learn an internal language representation and paraphrase the intent of the text rather than extracting the words from the text.

It requires data processing which involves the following:

- Filter unnecessary characters/sentences.
- Tokenize articles into words that is we do not care about the sequence or order of the words, we just have a bunch of words, and all of those words are input for one model.
- Create word embeddings to represent words numerically.

A. Why Word Embeddings?

A human would know that there is some relevance between Spain and Madrid but a computer would not know that they are related in any way. That is why we require word embeddings. Word embeddings give a semantic power and make it easy to find a connection between two words.

An abstractive text summarization technique was used by Huong Thanh Le et al. [10] in which they generated sentences using a keyword. This was governed by discourse rules and constraints that were syntactic. Word graph was used to generate sentences that were conceptually correct and hence the summary was created.

B. Recurrent Neural Networks

Traditionally, in neural networks, the set of inputs and outputs are independent of each other. This creates a problem when we need the model to predict the next word of the sentence. To do so we require the model to remember the previous words. RNN provides a solution to this dilemma by using a hidden layer. In RNN, the output generated from the previous move is used as an input to the current move. The same task is performed for every element in the sequence and the ouput depends on previous computations, that is why they are called recurrent neural network.

Doing so gives rise to a hidden state, this results in a very deep neural network and when backpropagation is applied to such a deep neural network, it results in vanishing or exploding gradients.

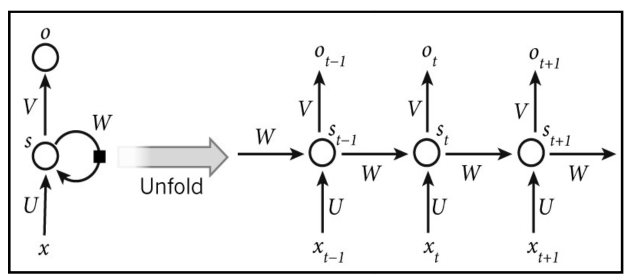

Fig. 3. An unfolded recurrent neural network.

The above is a depiction of an RNN being unfolded into a whole network. For instance, if the input sequence is of let's say n words, then the network would be seperated into n-layered neural network in which one layer denotes each word.

This creates a problem when it comes to long text input as the neural network can become too complex and might crash.

C. LSTM (Long Short Term Memory) Networks

Long Short Term Memory networks are a special category of RNN which solves the problem long text input and long term dependencies.

Chandra et al. [11], used a deep learning-based framework that generated a summary for products for an e-commerce website. In an unsupervised way, they carried out following steps: (a) for data aggregation, keywords and descriptions for a specific item were collected (b) removal of duplication to avoid redundancy and creating a dictionary for information about specific item (c) filtering of data according to user's requirement (d) for sentence generation, RNN with LSTM were used (d) to get ordered output, TextRank was implemented. The implementation of these steps was done using abstraction and extraction.

Ilya Sutskever et al. [12] used a method where they used multilayered LSTM to create a fixed dimensional word to vector sequence and to decode the target sequence, a deep LSTM is introduced. They used the WMT-14 dataset to translate English sentences to French and it received a BLEU score of 34.8 on the whole test set. Because of out-of-vocabulary words, this result was penalized.

LSTMs, similar to RNNs, also have a chain-like sequence, but the repeating module has a different structure. Four neural network layers interact uniquely, unlike the traditional networks which have only one neural network layer.

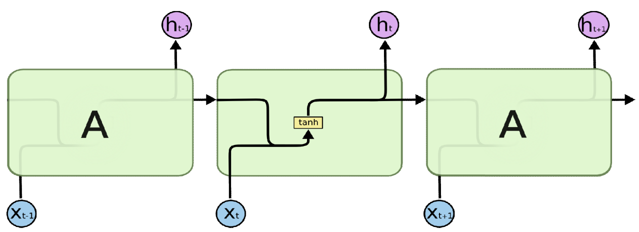

Fig. 4. The repeating module in a standard RNN contains a single layer.

Source: Adapted from [5]

The above figure shows that in a traditional RNN, there is only one layer in its repeating module. The figure below shows that in a repeating module of an LSTM, there are four interacting layers.

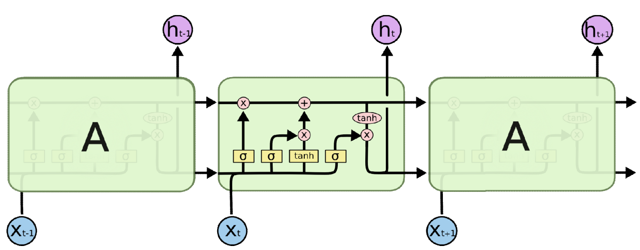

Fig. 5. The repeating module in an LSTM contains four interacting layers.

Source: Adapted from [5]

D. Encoder-Decoder Model with Attention

Ramesh Nallapati et al. [13] worked on an abstractive text summarizer that used the encoder-decoder model with attention. The authors used the Gigaword Corpus scripts to preprocess the data. It resulted in 3.8M training examples and 400K validation and test examples from which a subset of 2000 examples was taken for each process. 200-dimensional word2vec vectors trained for model embeddings on the script. For decoding purpose beam search of size 5 was used for generating a summary where the summary was limited to a maximum of 30 words. For evaluation, the full-length F1 variant of Rouge was used considering each highlight to be a separate sentence. The Project resulted in the repetition of words while generating summaries for multiple sentences.

The previous model can handle sequences of the same length. That is if the input is n set of words then the output should also be n set of words. But in the case of a translator, the set of input and output words would logically be different. To handle mappings between sequences of different lengths we've encoder-decoder model.

The problem with this model is that encoder does not depend on time so when we are producing the output for each time in decoder we only have one presentation from the encoder. The solution to this is attention.

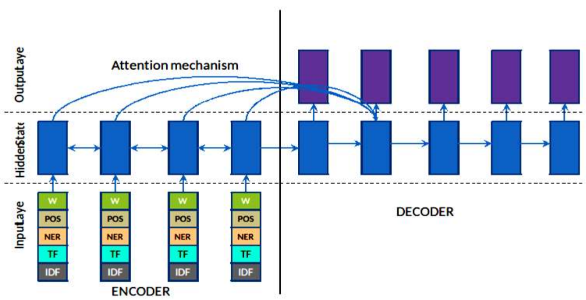

Fig. 6. Working of Attention Mechanism.

Source: Adapted from [9]

Attention is a type of mechanism that let's neural network focus its attention on some parts of the input. During the time of decoding process, this mechanism retains and utilizes all the hidden states of the input sequence. To fulfill this process a unique mapping is created between each time step of the decoder output to all encoder hidden states. The above diagram shows how the attention mechanism works. This means for each decoder output, it provides the access for whole input sequence and specific elements can be picked from the sequence to generate output.

E. Pointer Generation Mechanism

Abigail et al. [14] worked on a text summarizer that used a pointer generator mechanism to create output. CNN/Daily Mail dataset was used that contained news articles with multiple sentence summaries. The scripts supplied by Nallapati et al. were used which consisted of 287,226 training pairs, 13,368 validation pairs, and 11,490 test pairs. In this project, the word embeddings were learned from scratch during training.

Training took 3days and 4 hours and less than 230,000 training iterations. For evaluation, the F1 scores for ROUGE-1, ROUGE-2, and ROUGE-L were taken that measures the word-overlapping. The evaluation was also done with the METEOR package both in Exact Match Mode and Full Mode. The project resulted in good summaries by handling out-of-vocabulary words easily and copying factual details correctly.

Yang Liu et al. [15] explained the pre-trained model-Bidirectional Encoder Representations from Transformers (BERT) which is enhancing the natural language processing tasks. Paper describes the complete working of BERT in both extractive text summarization and abstractive text summarization.

Li Dong et al. [16] presented a new model called the Unified pre-trained Language Model (UniLM) which comes pre-trained using the following modeling tasks: sequence to sequence, unidirectional, and bidirectional prediction. The task of unified modeling is made functional by using specific self-attention masks to manage the flow of context the prediction takes upon and a shared Transformer network. It achieved an improved result of 40.51 in ROGUE-L on abstractive summarization of CNN/DailyMail data set.

TABLE I

Summarized Related Work

Year	Authors	Techniques	Description
2013	I. Lourentzou, P. Katsivelis-Perakis, S. Thomaidou, and M. Vazirgiannis	Sentiment Analysis and Natural Language Generation.	In this paper, inputting a landing webpage of the website, generated a short text of the advertisement.
2013	HuongThanh Le and Tien Manh Le	Abstractive text summarizer using syntactic constraints and word graph.	Word graphs were used to generate sentences using keywords.
2014	Sutskever, Ilya, Vinyals, Oriol, Le, and Quoc	Multilayered LSTM and deep LSTM.	Multilayered LSTM was used to create a fixed dimensional word to vector sequence and to decode the target sequence.
2016	Ramesh Nallapati, Cicero Nogueira dos Santos, Bowen Zhou, CaglarGulcehre and Bing Xiang	Encoder decoder model with attention and a large vocabulary.	Employed the framework of Attentional Encoder-Decoder Recurrent Neural Network outperforming the model on two different corpora.
2017	Abigail See, Christopher D. Manning and Peter J. Liu and	Abstractive Text Summarizer using pointer generator network and coverage mechanism.	Used a pointer-generator network that would copy words from the source text through pointing and Coverage was used to keep track of data which had been summarized.
2019	Yang Liu and MirellaLapata	Text Summarization with the pre-trained model (BERT).	For abstractive summarization, different optimizers were used for the encoder and the decoder to alleviate the mismatch between the two.
2019	Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou and Hsiao-Wuen Hon	Text Summarization with pre-trained language model (UniLM).	This paper presents the UNIfied pre-trained Language Model (UniLM) which can be utilized for both natural language understanding and generation tasks. The model is pre-trained using three types of language: unidirectional, bidirectional, and sequence-to-sequence prediction.

III. Research Methodology

This section describes the methodology adopted for the review. To carry out a systematic review following steps are required.

- Framing of a research question.
- Defining of search protocols. (inclusion/exclusion and selection of primary studies)
- Data extraction

A. *Research Question*

RQ1: What are popular techniques that have been utilized in literature for text summarization using NLP?

RQ2. Which technique of text summarization provides more accurate results?

B. *Search Protocols Criteria*

To present a systematic review, a proper selection of studies is required. These various protocols are defined to figure out the most relevant studies in the field of text summarization. Inclusion exclusion criteria are defined to reject irrelevant papers and are discussed in table II.

TABLE II

Tabular Representation of Inclusion and Exclusion

Criterion	Inclusion	Exclusion
Language	Papers published in English.	Papers published in other languages than English.
Publication Period	Papers published between 2013-2019.	Papers published before 2013.
Techniques	Abstractive technique papers.	Extractive technique papers.
Peer-review	Original research papers and reviews published in the journal.	Book chapters, reports, conference proceedings were excluded.
Outcome	Papers which describe the outcomes of used technique with proper	Papers with unclear outcomes and evaluation.

	evaluation.	

Data Extraction: Finally Fig.7 represents the year wise selection of paper utilized in this review.

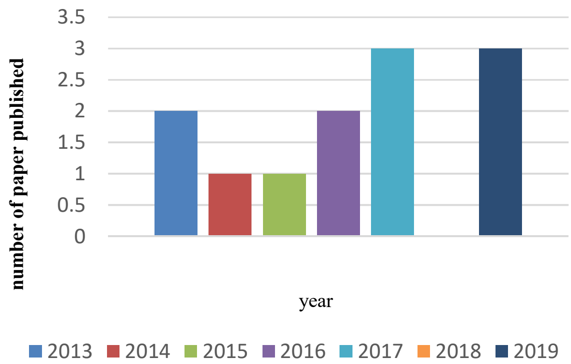

Fig. 7. Yearwise Selection of Papers

IV. Result And Discussion

TABLE III
Tabular Representation of Popular Techniques And Their Results

Models	**Methods**	**Result**
Encoder-Decoder Model with Attention	Attention is a type of mechanism that let's neural network focus its attention on some parts of the input. In the decoding process, this mechanism memorizes and utilizes all the hidden states of the input sequence. It is done by creating a different mapping between each time step of the decoder output and the hidden states of the encoder.	ROGUE 1: 35.46 ROGUE 2: 13.30 ROGUE L: 32.65
Pointer Generator	A hybrid network can copy words from the source text via pointing, which helps in accurate reproduction of input while retaining the ability to produce narrative words through the generator.	ROGUE 1: 36.44 ROGUE 2: 15.66 ROGUE L: 33.42
Pointer Generator with coverage	It is similar to pointer generator but it adds coverage mechanism that is it keeps track of the summary to avoid repetition.	ROGUE 1: 39.53 ROGUE 2: 17.28 ROGUE L: 36.38
BERTSU MABS	This applies bidirectional training of transformer to language modeling this results in a deeper sense of language context and flows as compared to single direction models.	ROGUE 1: 41.72 ROGUE 2: 19.39 ROGUE L: 38.76
UniLM	In this, the model is trained using three different techniques those are uni and bidirectional prediction and sequence to sequence prediction. The unification is achieved by using a shared transformer network and a unique self-attention mask.	ROGUE 1: 43.33 ROGUE 2: 20.21 ROGUE L: 40.51

Table III shows the popular models that have been utilized in literature for text summarization using NLP. For results various ROGUE (Recall-Oriented Understudy for Gisting Evaluation) scores were compared based on their output on CNN/Daily Mail dataset. ROGUE 1 and ROGUE 2 refer to the overlap of unigram and bigram between the system and reference summary respectively. ROGUE L is to calculate the longest common subsequence based statistics.

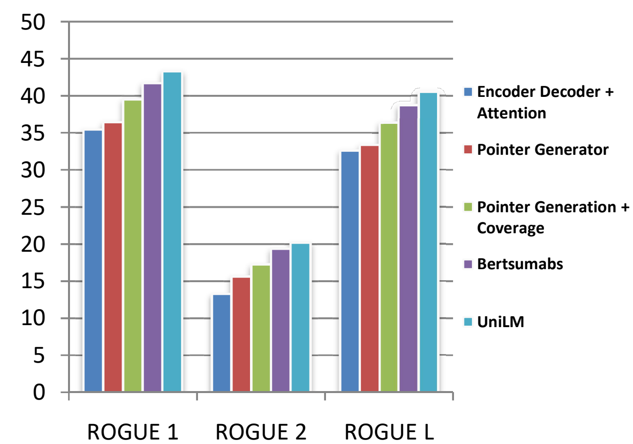

Fig. 8. Accuracy of Popular Techniques

Fig.8 shows the comparison between the popular techniques of abstractive text summarization. According to which UniLM provides the best output summaries.

V. Conclusion

As the usage of the internet is continuously increasing so is the volume of information present on it. Manually finding out the right information from this huge pile of information present is not easy but if we have a gist of the document we are getting in, it will become a lot easier. Just like the abstract of this review paper gives its insight. Text summarization does the same and generates a comprehensive summary. Thus, there is an immense need for text summarization to save time and effort. The better it gets the more we will be able to apply it to more complex language like that in a scientific paper or even an entire book.

In this paper, fundamental concepts and approaches to automatic text summarization have been discussed. The paper starts with a brief introduction to automatic text summarization and the work done in the past and present. This paper emphasizes on the various methods of abstractive text summarization like a recurrent neural network, long short term memory network, encoder-decoder model, and pointer generator mechanism.

REFERENCES

[1] Dalwadi, Bijal & Patel, Nikita, and Suthar Sanket, "A Review Paper on Text Summarization for Indian Languages".IJSRD - International Journal for Scientific Research & Development, Vol. 5, Issue 07, 2017.

[2] Arun Krishna Chitturi and Saravanakumar Kandaswamy, "Survey on Abstractive Text Summarization using various approaches". International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, No.6, November – December 2019.

[3] Deepali K. Gaikwad and C. Namrata Mahender, " A Review Paper on Text Summarization". International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 3, March 2016.

[4] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, "Hybrid clustering analysis using improved krill herd algorithm". Appl. Intell. 2018.

[5] L.M. Abualigah, A.T. Khader, M.A. Al-Betar, O.A. Alomari, "Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering". Expert Syst. Appl. 84, 24–36 2017.

[6] L.M. Abualigah, A.T. Khader, M.A. AlBetar, E.S. Hanandeh, "Unsupervised text feature selection technique based on particle swarm optimization algorithm for improving the text clustering". Eai International Conference on Computer Science and Engineering 2017.

[7] L.M. Abualigah, A.T .Khader, M.A. Al-Betar, Z.A.A. Alyasseri, O.A. Alomari, E.S. Hanandeh, "Feature selection with β-hill climbing search for text clustering application", Palestinian International Conference on Information and Communication Technology (PICICT) IEEE, 2017.

[8] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut, "Text Summarization Techniques: A Brief Survey". In Proceedings of arXiv, USA, July 2017.

[9] S. Thomaidou, I. Lourentzou, P. Katsivelis-Perakis, and M. Vazirgiannis, "Automated Snippet Generation for Online Advertising", Proceedings of ACM International Conference on Information and Knowledge Management (CIKM'13), San Francisco, pp.1841- 1844, USA, 2013.

[10] Chandra Khatri, Sumanvoleti, Sathish Veeraraghavan, Nish Parikh, Atiq Islam, Shifa Mahmood, Neeraj Garg, and Vivek Singh, "Algorithmic Content Generation for Products", Proceedings of IEEE International Conference on Big Data, Santa Clara, pp.2945-2947, CA 2015.

[11] Huong Thanh Le and Tien Manh Le, "An approach to Abstractive Text Summarization", In proceeding of International Conference of Soft Computing and Pattern Recognition (SoCPaR), Hanoi, Vietnam, Dec 2013.

[12] Sutskever, Ilya & Vinyals, Oriol and Le, Quoc, "Sequence to Sequence Learning with Neural Networks", Advances in Neural Information Processing Systems, 2014.

[13] Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos Santos, Caglar Gulcehre, and Bing Xiang, "Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond", The SIGNLL Conference on Computational Natural Language Learning (CoNLL), 2016.

[14] Abigail See, Christopher D. Manning, and PeterJ.Liu from, "GetToThePoint: Summarization with Pointer-Generator Networks", Association for Computational Linguistics, 2017.

[15] Yang Liu and Mirella Lapata, "Text Summarization with pre-trained encoders", Institute for Language, Cognition and Computation School of Informatics, University of Edinburgh, 2019.

[16] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon, "Unified Language Model Pre-Training for Natural Language Understanding and Generation". Conference on Neural Information Processing Systems, 2019.