IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 5, May 2023

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.423

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205109 |

Text Summarization Framework Using Machine Learning

Sairaj Sanjay Bankar, Kadambari Santosh Deshmukh, Dhruv Nandkishor Kale, Jayesh Kailash Sukale, Vanita Babanne

Department of Computer Engineering, RMD Sinhgad School of Engineering, Warje, Pune, India

ABSTRACT: An important natural language processing appli- cation, automatic text summarizing aims to condense a given textual content into a shorter model by using machine learning techniques. As media content transmission over the Internet continues to rise at an exponential rate, text summarization utilizing neural networks from asynchronous combinations of text is becoming increasingly necessary. Using the principles of natural language processing (NLP), this research proposes a framework for examining the intricate information included in multi-modal statistics and for improving the features of text summarization that are currently available. The underlying principle is to fill in the semantic gaps that exist between different types of content. In the following step, the summary for relevant information is generated using multi-modal topic modelling. Finally, all of the multi-modal aspects are taken into account in order to provide a textual summary that maximizes the relevance, non-redundancy, believability, and scope of the information by allocating an accumulation of submodular features.

KEYWORDS: word vectors, word analogies, fast text,Integer linear programming, text summarising, natural language pro- cessing.

I. INTRODUCTION

Now a days, there are large numbers of documents or information that is present related to any particular field. There are many sources out of which we can gather a lot of information that will be pertinent to our field of search. Much information is available at various sources like the internet. But, as we know that a huge amount of information cannot be always considered or taken into use. So, a precise amount of information is always considered and that information is drawn out from the original document that is huge in size. In other words, we can say that we pluck out the summary of the main document. A summary of any document is defined as a collection of essential data by collecting the brief statements accounting the main points of the original document. Therefore, Summarization of a text is a procedure of separating or getting the relevant data out of a very large document. It is the process of shortening the text document by using various technologies and methodologies to create a coherent summary including the major points of the original document. There are various methods by which the summarization process can be carried out. While most summarization systems focus on only natural language processing (NLP), the opportunity to jointly optimize the quality of the summary with the aid of automatic speech recognition (ASR) and computer vision
(CV) processing systems is widely ignored. On the other hand, given a news event (i.e., news topic), multimedia data are generally asynchronous in real life. Thus, Text summarization faces a major challenge in understanding the semantics of information. In this work, we present a system that can provide users with textual summaries to help to acquire the gist of asynchronous data in a short time without reading documents from beginning to end. The purpose of this work is to unite the NLP with neural network techniques to explore a new framework for mining the rich information contained in multimodal data to improve the quality of Text summarization.

Summarization is the process of compressing a long piece of material into a shorter version that retains the essential information. There are two types of summarization methods: extractive and abstractive. Extractive approaches create summaries solely from sections (typically full sentences) extracted straight from the source material, whereas abstractive methods may generate new words and phrases not found in the source text — as a human-written abstract normally does. Because copying huge portions of text from the original document provides baseline levels of accuracy, the extractive approach is easier.Text summarization has previously been split into two subtasks, namely sentence scoring and sentence selection, in prior publications that use extractive approaches. Sentence scoring is a technique for

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205109 |

assigning an importance value to each sentence that has been extensively researched in the past. Extractive methods create summaries by reproducing parts of the source content (typically entire sentences), whereas abstractive methods may generate new words or phrases not found in the source document. Extractive summarization, which is commonly characterised as a sentence ranking or binary classification problem (i.e., sentences that are top ranked or predicted as True are selected as summaries), has received a lot of attention in the past.Content selection in summarization is normally performed by sentence (and, on rare occasions, phrase) extraction. Despite the fact that deep learning models are a significant component of both extractive and abstractive summarization systems, it is unclear how they accomplish content selection with only word and sentence embedding based features as input.One of the most difficult NLP tasks is summarization, which is defined as the process of generating a shorter version of a piece of text while keeping critical context. information. The performance of sequence-to-sequence neural networks on summarization has lately improved significantly.

The availability of large-scale datasets, on the other hand, is critical to the effectiveness of these models. Furthermore, the length of the articles and the variety of styles might add to the complexity. Because news stories have their own distinct characteristics, systems trained solely on news may not be adequately generalised. Recent advances in machine learning have resulted in significant advancements in text summarization. Huge labelled summarization corpora, such as the CNN/Daily Mail dataset, have made it possible to train deep learning models with a large number of parameters. Recurrent neural network (RNN) and Arif Ur Rahman, the associate editor who coordinated the evaluation of this manuscript and approved it for publication, were among them. For text summarization, convolution neural networks (CNN) have been frequently employed. RNN is used in extractive approaches to evaluate sentence importance while simultaneously picking representative sentences.

A. Motivation

Text summarising is a technique for condensing information from a source text into a few representative sentences in order to construct a coherent summary containing relevant information from source corpora.deep neural network-based summarization models have a number of serious flaws. To begin, a significant quantity of labelled training data is re- quired. This is a common issue in low-resource languages where publicly available labelled data is lacking. So that we propose a model, Learning Free Integer Programming Summarizer (LFIP-SUM), which is an unsupervised extractive summarization model.

B.*Objectives*

- To automatically generate a fixed-length textual summary to represent the principle content of the text data.
- To design a graph-based model to effectively calculate the salience score for each text unit, leading to more readable and informative summaries.
- To generate summaries using document text using ma- chine learning.
- Try to improve accuracy.

II. LITERATURE SURVEY

Abigail See et al: In this paper, We demonstrated that a hybrid pointer generator design with coverage lowers inaccuracy and repetition. We tested our model on a fresh and difficult lengthy text dataset and found that it outperformed the abstract state-of-the-art result significantly. Our model has a lot of abstract skills, but getting to higher degrees of abstraction is still a work in progress.

Qingyu Zhou et al: We introduce an unique neural network architecture for extractive document summarization in this paper, which addresses this problem by learning to score and pick sentences simultaneously. The most notable difference between our method and earlier methods is that it integrates sentence rating and selection into a single phase. It rates sentences based on the partial output summary and current extraction state each time it selects one. The suggested combination sentence scoring and selection strategy greatly outperforms the existing segregated method, according to ROUGE evaluation results.

Xingxing Zhang et al: In this paper, We proposed a latent variable extractive summarization strategy that uses a sentence compression model to directly exploit human summaries The proposed approach outperforms a powerful extractive model in experiments, whereas applying the compression model on the output of our extractive system produces inferior results. We intend to investigate approaches to train compression models specifically for our summarising task in the future.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205109 |

Chris Kedzie et al: In this study,An empirical research of deep learning-based content selection algorithms for summarization was given. Our findings imply that such models have significant limits in terms of their capacity to learn robust features for this task, and that more work on sentence representation for summarization is required.

Linqing Liu et al: We suggested an adversarial technique for abstractive text summarization in this study. Experiments revealed that our model was capable of producing more abstract, legible, and diversified summaries.

Jacob Devlin et al: In this paper,Recent empirical gains in language models owing to transfer learning have shown that rich, unsupervised pre-training is an important feature of many language comprehension systems. These findings, in particular, show that deep unidirectional topologies can benefit even low-resource jobs. Our key contribution is to extend these findings to deep bidirectional architectures, allowing a single pre-trained model to solve a wide range of NLP tasks.

T. Boongoen et al: This survey has presented classical and recently developed approaches to cluster ensemble.It begins with the formal terms used to define the problem. Following that, four main categories of consensus clustering approaches are explained in depth with examples. Following that, it describes extensions to three major components of a cluster ensemble framework: ensemble generation, representation and summarization, and consensus function. Many cluster ensemble approaches have been used for a wide range of applications and domain challenges due to their improved

ability to deliver accurate data partitions.

Mahnaz Koupaee et al: In this paper,We present WikiHow,a new large-scale summary dataset made up of a variety of articles from the WikiHow knowledge base The aspects of WikiHow outlined in the research may provide additional challenges to summarization systems. We expect that the new dataset will pique the interest of academics as a viable option for evaluating their systems.

Edouard Grave et al: In this work, we contribute word vectors trained on Wikipedia and the Common Crawl, as well as three new analogy datasets to evaluate these models, and a fast language identifier which can recognize 176 languages.We investigate the impact of several hyper parameters on the trained models' performance, demonstrating how to produce high-quality word vectors. We also show that, despite its noise, employing common crawl data can result in models with greater coverage and better models for languages with little Wikipedia. Finally, we find that the quality of the produced word vectors is substantially lower for low-resource languages, such as Hindi, than for other languages. We'd like to look at further ways for improving the quality of models for such languages in the future.

Zhilin Yang et al: In this study,XLNet is a generalised AR pretraining method that combines the benefits of AR and AE methods by using a permutation language modelling target. XLNet's neural architecture was designed to work in tandem with the AR goal, including the integration of Transformer- XL and the meticulous design of the two-stream attention mechanism. On a variety of tasks, XLNet delivers significant improvements above previous pretraining objectives.

III. PROPOSED SYSTEM

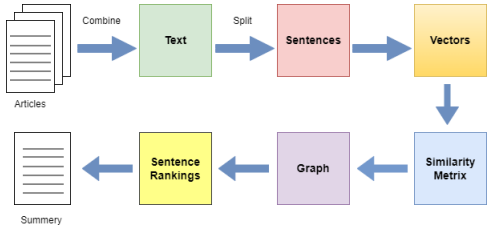

Fig. 1. Proposed System Architecture

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205109 |

It's a two-step process that starts with document repre- sentation and ends with the selection of sample sentences. Distributed vectors pre-trained by sentence embedding models are used to represent a document as a continuous vector. The sentence importance score is then assessed using ILP and PCA, and representative phrases for the summary are chosen. An ensemble of different pre-trained sentence representations
was used to increase model performance further. The first step would be to concatenate all the text contained in the articles.Then split the text into individual sentences.In the next step, we will find vector representation for each and every sen- tence.Similarities between sentence vectors are then calculated and stored in a matrix.The similarity matrix is then converted into a graph, with sentences as vertices and similarity scores as edges, for sentence rank calculation.Finally, a certain number of top-ranked sentences form the final summary.

A. Algorithm

Phase 1 – Data Preprocessing

Apply preprocessing algorithms – Remove unwanted data using preprocessing algorithms

Phase 2 – TFIDF

TF: Term Frequency, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).
IDF: Inverse Document Frequency, which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

IDF(t) = log e(Total number of documents / Number of documents with term t in it).
Example:
Consider a document containing 100 words wherein the word cat appears 3 times. The term frequency (i.e., tf) for cat is then (3 / 100) = 0.03. Now, assume we have 10 million documents and the word cat appears in one thousand of these. Then, the inverse document frequency (i.e., idf) is calculated as log(10,000,000 / 1,000) = 4. Thus, the Tf-idf weight is the product of these quantities: 0.03 * 4 = 0.12.

Phase 3 – Cosine similarity weight

Calculate cosine similarity of sentences. Remove duplicate sentences using cosine similarity weight. Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi- dimensional space. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity.

A. Mathematical Model

- DEEP REPRESENTATION OF A DOCUMENT

To capture the latent meaning of each sentence in this investigation, we used deep representation vectors. A document is assumed to have n sentences in it. We can express a text as a sequence of sentences since phrases contain sequential information:

$$D = [s1,s2, \ldots ,sn]$$

D denotes the document and sk refers to its k-th sen- tence. Let a sentence be represented as a column vector. Thus, Dbasic, the basic representation of D, becomes the following matrix

$$Dbasic = [sv1,sv2, \ldots ,svn]$$

where svk denotes the pre-trained sentence vector ofsk . Dbasic is a d × n matrix, where d is the embedding di- mension. the positional encoding matrix PE is calculated as follows:

$$PE(pos,2i) = \sin(pos/100002i/d) \quad PE(pos,2i+1) = \cos(pos/100002i/d)$$

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205109 |

- PRINCIPAL COMPONENT ANALYSIS

we defined the sentence importance score as follows: imp(si) = X k6=i cos(svdeep,i, PSk) (27)
where imp(si) is the sentence importance score of sentence si and cos denotes cosine similarity

- INTEGER LINEAR PROGRAMMING

1. FORMULATION

The minimal extraction unit in our study is a sentence, not a concept, we formulate the optimization issue similarly. The formulation is based on sentence significance scores and sentence similarity scores. The cosine similarity of the deep sentence representation vectors of the sentences si and sj is defined as follows:

sim(si,sj) = cos(svdeep,i,svdeep,j)

PCA determines the right number of summary sentences for each document automatically. As a result, we change Eq. as follows:

Xn i=1 xi = N

where N is the appropriate number of summary sentences.

2. SENTENCE PRUNING

Sentence trimming is used to lower the temporal complexity of LFIP-SUM. The sentences to be extracted are those with high sentence importance and low redundancy scores, according to the equations above. As a result, we define a sentence's trimming score as:

Pr score(si) = imp(si) 1 n1 P j6=i sim(si,sj) .

The maximum sentence number, l, is the sentence trim- ming hyper-parameter. Sentences are not pruned in texts with lengths less than or equal to l, but in documents with more than l sentences, ILP is applied to the top l sentences based on their Pr scores.

IV. RESULTS AND DISCUSSION

Experiments are done by a personal computer with a configuration: Intel (R) Core (TM) i3-2120 CPU @ 3.30GHz, 4GB memory, Windows 7, MySQL 5.1 backend database and Jdk 1.8. The application is web application used tool for design code in Eclipse and execute on Tomcat server. Some functions used in the algorithm are provided by list of jars like weka jars etc.

Example 1: a microlight aircraft crashed near mumbai in the south island of india late sunday afternoon, killing the pilot. Summarization Result - microlight aircraft crashes in india killing pilot.

V. CONCLUSION

In this paper, we propose an summarization model that does not require parameter training. we used pre-trained sentence vectors, positional encoding, and self-attention to create a deep representation of documents. Following that, we extracted PS vectors that preserve as much information as possible from a document , and we utilised them to select an acceptable number of summary sentences and compute the relevance score of each original sentence. Finally, we constructed an ensemble model based on distinct models with varied pre- trained sentence embedding vectors to produce the final sum- marization results. We confirmed that the suggested model's performance (without training examples) was comparable to that of a state-of-the-art machine learning-based supervised model.

VI. FUTURE SCOPE

In future research we will focus on multi-document and cross-document text summarization.

REFERENCES

1. MYEONGJUN JANG AND PILSUNG KANG, "Learning-Free Unsupervised Extractive Summeriza- tion Model ," 2021, arXiv:9321308 [Online].Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=9321308
2. A. See, P. J. Liu, and C. D. Manning, "Get to the point: Sum- marization with pointer-generator networks," 2017, arXiv:1704.04368. [Online].Available: http://arxiv.org/abs/1704.04368.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2023 ||

| DOI:10.15680/IJIRSET.2023.1205109 |

3. Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, and T. Zhao, "Neural document summarization by jointly learning to score and select sen- tences," 2018, arXiv:1807.02305. [Online]. Available: http://arxiv.org/ abs/1807.02305.
4. X. Zhang, M. Lapata, F. Wei, and M. Zhou, "Neural latent extractive document summarization," 2018, arXiv:1808.07187. [Online]. Avail- able:http://arxiv.org/abs/1808.07187.
5. C. Kedzie, K. McKeown, and H. Daume, "Content selection in deep learning models of summarization," 2018, arXiv:1810.12343. [Online].
6. L. Liu, Y. Lu, M. Yang, Q. Qu, J. Zhu, and H. Li, "Generative adversarial network for abstractive text summarization," in Proc. 32nd AAAI Conf. Artif. Intell., 2018, pp. 1–2.
7. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018,arXiv:1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805.
8. T. Boongoen and N. Iam-On, "Cluster ensembles: A survey of ap- proaches with recent extensions and applications," Comput. Sci. Rev., vol. 28, pp. 1–25, May 2018.
9. M. Koupaee and W. Y. Wang, "WikiHow: A large scale text summarization dataset," 2018, arXiv:1810.09305. [Online]. Available: http://arxiv.org/abs/1810.09305.
10. E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," in Proc. Int. Conf. Lang. Resour. Eval. (LREC), 2018, pp. 1–3.
11. Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and
Q. V. Le,"XlNet: Generalized autoregressive pretraining for language understanding," in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 5753–5763.
12. C. Kedzie, K. McKeown, and H. Daume, "Content selection in deep learning models of summarization," 2018, arXiv:1810.12343. [Online].
Available: http://arxiv.org/abs/1810.12343
13. Z. Li, Z. Peng, S. Tang, C. Zhang, and H. Ma, "Text summarization method based on double attention pointer network," IEEE Access, vol. 8, pp. 11279–11288, 2020
14. J. Tan, X. Wan, and J. Xiao, "Abstractive document summarization with a graph-based attentional neural model," in Proc. 55th Annu. Meeting Assoc. Comput. Linguistics, vol. 1, Jul. 2017, pp. 1171–1181.
15. H. Kim and S. Lee, "A context based coverage model for abstractive document summarization," in Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC), Oct. 2019, pp. 1129–1132.
16. L. Liu, Y. Lu, M. Yang, Q. Qu, J. Zhu, and H. Li, "Generative adversarial network for abstractive text summarization," in Proc. 32nd AAAI Conf. Artif. Intell., 2018, pp. 1–2.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.423

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

Scan to save the contact details