Extractive summarization in Hindi using BERT-based ensemble model

Aravind Dendukuri [§]
Manipal Institute of Technology
Manipal Academy of Higher Education
Hyderabad, India
denarvi@gmail.com

Sagar Goyal [§]
Manipal Institute of Technology
Manipal Academy of Higher Education
Prayagraj, India
shgoyal33@gmail.com

Jannat Arora
Thapar Institute of Engineering and Technology
Chandigarh, India
jannatarora21@gmail.com

Abhinav Pradeep
Manipal Institute of Technology
Manipal Academy of Higher Education
kochin, India
abhinavp2301@gmail.com

***Abstract*—The past few years have seen a massive growth in the number of daily internet users whose primary language of communication is Hindi. Hindi is now one of the most spoken languages in the world and the official language of the Indian Government. Given this considerable rise in the amount of data in Hindi, managing, analyzing, and summarizing documents becomes a significant task with many applications. But language models and Natural Language Processing tasks catering to this demographic have been very limited in scope. Even state-of-the-art multilingual models cannot handle the nuances of the language. To bridge this gap, the MuRIL [37] language model was implemented and trained on large-scale Indian text corpora. The present work focuses on the summarization task for Hindi documents. We leverage the power of the MuRIL model and develop a novel extractive summarization-based solution using the language model's embeddings. Newspaper articles spanning several categories are extracted as our training data, and comprehensive testing shows that our model exceeds the performance of the previous baselines on the accuracy metric.**

***Index Terms*—summarization, MuRIL, BERT, Hindi, Extractive, Embeddings, Clustering**

I. Introduction

Text summarization refers to the process of condensing the contents of a text document into a shorter version while maintaining the essential gist of the input. These documents help in the decision-making process as only the core information about the document is maintained and presented. This field of research was first discussed in the 1950s [1] and finds its applications in the varied areas of Natural Language Processing (NLP) like Search Engines; Financial research; Automatic content creation, and Question-Answering bots.

Text-based summarization usually follows two different approaches: Abstractive and Extractive. Abstractive algorithms create summaries by paraphrasing the original document, with a vocabulary set often different from the initial input. It works by creating a semantic representation of the document and using its innate vocabulary to present a compressed version that substitutes for the major points in the input. Hence, this requires several text modifications and sentence generation steps to be performed. On the other hand, Extractive summarizers perform their function by identifying the key points in the input passage and directly extracting it from the document. These key sentences are then concatenated to act as the summary. In the present work, extractive summarization is explored.

The crux of extractive summarization is the selection of the sentences from the document that will form the summary. One particular method for sentence selection is assigning numerical weights to the sentences and ranking them to select the most appropriate ones [7]. Another way is by summing up the weights of the individual term or phrases that make up a sentence in a process called term weighting [8].

While English text summarization has seen a lot of research [2]–[5], work on resource-limited languages like Hindi has been severely limited, with only a few works exploring the topic [9]. Since Hindi is a free order language finding linguistic features in the sentence structures is challenging. Hence, modern methods have primarily focused on exploring the statistical features in a rule-based setting [10]. Modern methods use a mixture of word-level features like term frequency, length of words, and word occurrences in headlines and sentence-level features like sentence length, sentence position, and similarity to headline to select the critical sentences [11]. But, the focus of our model is the use of Hindi-based Language models [37]. There exist many neural models for Extractive summarization in English [12]–[14], with BERTSUM [17] and its follow up [15] achieving the state-of-the-art ROUGE score for both Abstractive and Extractive summarization. But, the use of such language models is minimal in Hindi. Hence, through our current work, we aim to explore the usage of the MuRIL language model on the Hindi extractive summarization task and create a baseline for future work in this area.

[§]Equal contribution

The rest of the paper is organized as follows. We discuss the related research in the next section. In section 3, we describe our dataset and our collection process. In section 4, we present our model and our methodology. Following that, in section 5, we display the result of our model, and finally, in section 6 contains the conclusion and scope for future work.

II. Related research

The first reported work on Extractive summarization was by Luhn Et.al [1]. Their seminal work operated on word frequencies to isolate the key phrase for the summary. After finding the significant terms in a document, the summary was compiled by sorting all the words based on their frequency and seeing the number of significant terms within a candidate sentence. Follow up work by Lin et al. [16] explored the importance of sentence position in choosing a key sentence. Following this, researchers worked on automating the summarization task. [31] studied the graph scoring method for sentence selection while Parveen et al. [32], [33] worked on coherence-based summary generation. It was only recently that the focus has shifted to Neural networks for the summarization task [34], [35].

In the present work, the Transformer [18] based pre-trained language models are used to solve the extractive summarization problem.

A. Pretrained Language Models

BERT [19] uses the Transformer's attention mechanism to understand contextual relationships between words/sub-words in a text. Through its bidirectional training, it was able to trump the performance of the previous single-direction language models and allowed for state-of-the-art results in various downstream tasks like question-answering, sentence-classification and intent classification with minimal finetuning.

BERT can obtain such performances because of the two unsupervised tasks it has been pretrained with: Masked LM (MLM), where a portion of the input is masked and the model is trained to predict the hidden words and Next Sentence Prediction (NSP), where a pair of sentences are provided and the model predicts which sentence follows the other.

One problem with BERT is that it has been trained exclusively on a massive English corpus. Hence, after the development of BERT, multiple languages variations have been developed. mBERT is multilingual that has been pretrained in 104 languages. But, due to the high number of languages on which the model is trained, resource lean languages do not receive enough representation, and hence, their performance on benchmarks suffers. Therefore, multiple BERT variations based on regional languages have also been proposed. Some examples include AraBERT [20] for Arabic and GottBERT [21] for German.

For our research, we explore the MuRIL model [37]. MuRIL supports 17 Indian languages and is one of the few Language models trained in Hindi. Due to this, it can beat mBERT's performance on all tasks in the cross-lingual XTREME benchmark [22]. It has also been pretrained on the same MLM and NSP objectives as BERT.

The other disadvantage of the BERT architecture is that while it has been trained to output word-level embeddings, no independent sentence-level embeddings are computed. To overcome this obstacle in linear time, the SBERT [23] architecture was proposed. This model appends a pooling operation to the BERT model's output and derives a fixed-sized sentence embedding.

Another avenue of research explored in this work is summarization through unsupervised clustering. Through the years, a vast amount of research has been performed on this [24]–[26]. It was only recently that researchers used embeddings from BERT-based language models. [27] explores k-means clustering leveraging the BERT model embeddings. Candidate summary sentences are chosen from the embedded sentences closest to the centroid.

B. Hindi based research

One of the first works on Hindi summarization was performed by Thaokar et al. [28]. They proposed a model for summarization using semantic graphs and the particle swarm optimization algorithm. They used the Hindi wordnet embeddings to obtain part-of-speech details about the input text and used statistical and linguistic features of the text to form the final summary. Follow-up research also explored the linguistic aspects of the text using a rule-based approach [29]. [30] uses a combination of attributes like sentence position, number of words in a sentence along with linguistic features like the presence of proper nouns in a sentence to choose its candidates. To the best of our knowledge, no significant work has been conducted in the use of language models or deep neural networks for the Hindi summarization task, making our model the first of its kind.

Dataset size (articles)	59313
Training set size (articles)	47450
Mean article length (words)	597.55
Mean summary length (words)	55.22
Compression Ratio	10.81
Total Vocabulary Size (words)	493248
Occurring 10+ times	89214

TABLE I: Dataset Description

III. Dataset

The Hindi Text Short and Large Summarization Corpus [1] is the only opensource benchmark for Hindi text summarization available to us. It consists of 1,42,168 articles in the training set. But, out of those, only 69,030 articles have a gold summary associated with them. To complement this dataset, data has also been collected from inshort's Hindi website

[1]https://www.kaggle.com/disisbig/hindi-text-short-and-large-summarization-corpus

[2]. Inshort is a news summarization website where native speakers manually summarise news articles in under 60 words. Since language specialists have performed the work, these summaries act as gold data for our research. This approach is similar to the dataset collection stage of [11].

To maintain consistency and to clean the data compiled, the following steps are taken for both the datasets:

1) All articles written primarily in English are removed from the corpus.
2) All articles below 500 characters are removed.
3) Entries leading to non-text-related article sources are ignored in the final dataset. This includes video sources and tweets.
4) Author details, external links, non-Hindi characters are removed.

These steps finally lead to 44,133 entries from the kaggle corpus and 15,180 entries from the inshorts corpus.

In both cases, the summaries attached to the articles are abstractive. Hence, they are not suitable for an extractive summarization model. Therefore, a greedy algorithm is used to generate the oracle summaries. The algorithm measures the ROUGE score between the lines in the summary and the article. The line from the article with the highest score is then added in the oracle summary [38].

IV. Methodology

Our summariser works through an ensemble of a Supervised MuRIL-based Siamese model, an unsupervised k-means centroid based model and rule-based approach based on the statistical and linguistic structure of the text.

A. Sentence Embeddings using Siamese MuRIL network

Based on the SBERT model [23], we use the sentence embeddings from the MuRIL-based Siamese network. This configuration proves to be uniquely suitable for various pair-based regression tasks, even in a low-resource setting. In the present model, we treat the extractive summarization problem as a binary classification one. Each sentence in the input document is given a label of 0 or 1 based on its absence or presence in its respective documents summary. This formulation allows us to pass the sentences through the MuRIL Siamese network and obtain a score.

Nallapati et al. [38] first established this classification setup, where a RNN based Sequence model was used. Based on our research, apart from [36], no other study has taken place exploring the use of Siamese BERT-based networks for summarization.

Similar to [38], each sentence in a document is approached sequentially, and a binary classification is performed to decide if the statement belongs in the summary or not. We treat this task as a query-relevance problem statement, that is, given a query document, train our model to find the most relevant sentences. Here, we compute relevancy as the presence of a statement in the extractive summary for a given query document. We perform this objective by outputting likelihood scores that a sentence belongs in a document's summary.

We explore the Siamese Sentence-BERT architecture to obtain the sentence representations. This architecture is then fine-tuned with the Indic MuRIL model. The network can derive sentence/text embeddings by averaging, through mean-pooling, all contextualized word embeddings MuRIL provides. This provides a 768-dimensional vector output irrespective of the input length [23]. Therefore, we can obtain a shared embedding space for the document and the sentences in the text and find the most similar sentences during inference time.

After concatenation of our text representations from the sentence and the document, we feed the result to a classification head. The final layer for this is a linear classifier with Sigmoid activation function to decide whether the sentence should be included or not with a Binary Cross Entropy loss function, as defined below:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{n} y_i \log(\hat{y_i}) + (1 - y_i) \log(1 - \hat{y_i})$$

B. k-means centroid using sentence embeddings

We leverage the sentence-transformer architecture presented in [23] to create sentence embeddings. The MuRIL model is used again to initialize this architecture. Though, in this case a Siamese network is not used. A linear MuRIL architecture is implemented that averages the results of the contextual embeddings through mean-pooling to derive the sentence embeddings. This model is built per the training instructions presented on the sbert website [3].

As is standard with all BERT-based models, we can choose any of the model's layers for its embeddings. The most common practice is to use the first vector associated with [cls] which creates the required N x E matrix for the clustering, here N is the number of sentences, and E is the embedding dimension. Other layers have a N X W X E matrix, where W is the number of tokenized words. Our experimentation found that the embeddings from the [cls] layer can produce quality results.

After the required embeddings are generated for each of the sentences in the article, we group them into clusters such that sentences belonging to the same cluster have similar semantic meanings. The final number of such clusters is determined as hyperparameter K. This K equals the number of sentences in our final summary. We use nltk's implementation of k-means clustering for this work.

Finally, the sentence closest to the centroid of a given cluster is chosen as a candidate for that cluster. The similarity or the distance between a sentence S, and the cluster C is done using Cosine similarity.

$$Sim(C, S) = \frac{(C) \cdot (S)}{||C|| \cdot ||S||}$$

After the sentences have been ranked based on their distance from their respective cluster's centroid, the top-ranked

[2]https://www.inshorts.com/hi/read

[3]https://www.sbert.net/docs/training/overview.html

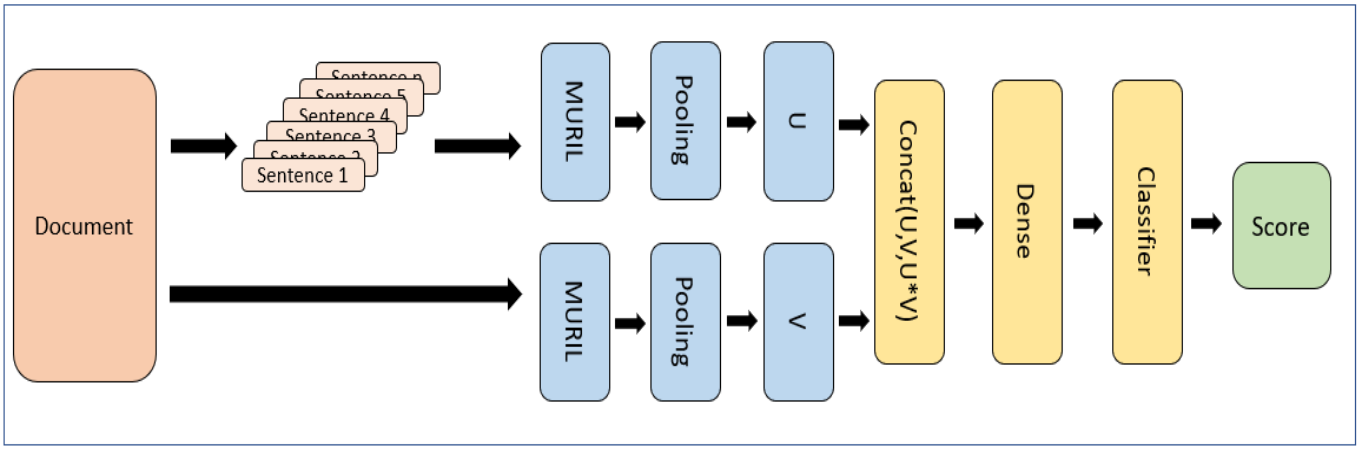

Fig. 1: Siamese-MuRIL-based classification

sentences are appended to the final summary. To ensure richer and significant document compression, we measure the cosine distance between a candidate sentence and the sentences already present in the summary. Any sentence that crosses our sentence threshold is discarded. All summary sentences are then sorted based on their position in the input article and the final summary generated.

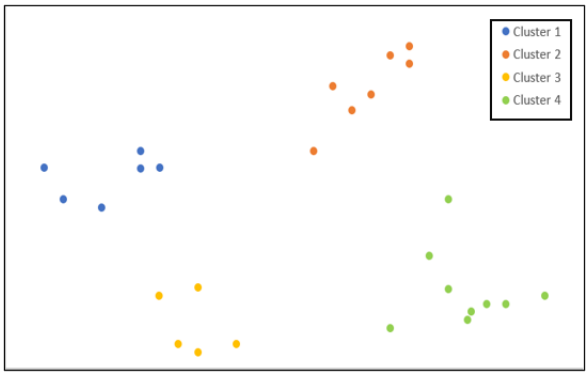

Fig. 2: Clustered embeddings from the [cls] layer

C. Statistical and Linguistic Approach

In this model, candidate sentences are chosen based on the statistical (based on frequencies of the elements in the text) and linguistic (related to the text's language-based intricacies) aspects of the core text.

After the necessary preprocessing steps, detailed in the next section, have been performed on the text word level and sentence-level features are extracted from the text to obtain the final summary. These features have been inspired from [11].

1) Word-level features
   - Frequency-based (TF-IDF): Term frequency (TF) is defined as the regularity of a word in a document, divided by how many words there are and Inverse Document frequency (IDF) for a single document implementation measures the uniqueness of a term by dividing the number of paragraphs in a document by the number of documents with the given term in it. The required measure is the product of these measures. We calculate the score for all unigrams in our text.
   - Presence in the heading of the article: Since our given corpora is derived from news articles, more weightage is given to terms that occur in the headline of the article as well.

   The weights are calculated and normalized.
2) Sentence-level features
   - Number of words in a sentence: The ratio of the number of words in a sentence to the number of words in the largest sentence in the text is taken.
   - Position of the sentence in the article: The sentences at the beginning and at the ending of the text are assumed to have the largest weight and given the greatest score. The scores are reduced sequentially from the beginning of the article to the middle and then sequentially increased till the end.
   - Similarity to the headline: Cosine distance is taken between all the candidate sentences in the text and the article's headline. Sentences with the biggest similarity are given the highest score.
   - Presence of nouns in the sentences: Sentences with nouns are considered more important and given a high score.

The final scores for all the metrics are added, and sentences are sorted based on their accumulated score. The top-k sentences are then taken as the summary for the input.

D. Ensemble model

Our final model is the combination of all the three approaches described previously. Through the final sentences chosen by each of the three models, we propose a hard-voting ensemble. This ensemble works by summing the predictions made by each of the models. The final sentences chosen in the summary are selected by the majority of the models. This

बॉलीवुड लीजेंडरी कोरियोग्राफर पर बायोपिक फिल्म बनाएंगे टीसीरीज के मालिक भूषण कुमार ने आज बड़ी घोषणा करते हुए बताया कि वो देश की पहली महिला कोरियोग्राफर रह चुकी सरोज खान के के जीवन पर एक बायोपिक फिल्म बनाने जा रहे हैं। इसके लिए उन्होंने राइट्स भी हासिल कर लिए हैं। टीसीरीज के मालिक भूषण कुमार ने आज बड़ी घोषणा करते हुए बताया कि वो देश की पहली महिला कोरियोग्राफर रह चुकी सरोज खान के के जीवन पर एक बायोपिक फिल्म बनाने जा रहे हैं। इसके लिए उन्होंने राइट्स भी हासिल कर लिए हैं। आज सरोज खान की पुण्यतिथि के मौके पर उन्होंने उनके फैंस के लिए ये बड़ी घोषणा की है। सरोज खान जिनका असली नाम निर्मला किशनचंद साधू सिंह नागपाल थाउन्होंने अपने नृत्य और अपनी अदाओं की समझ से देश कई प्रतिष्ठित अभिनेत्रियों को नाच सिखाया। इतना ही नहींउन्होंने कई अभिनेताओं को नाचने का गुर सिखाया और लोगों के बीच उन्हें सफलता दिलाई। बॉलीवुड में 80 और 90 का दशक सरोज खान के नाम रहा। श्रीदेवी से लेकर माधुरी दीक्षित तकसरोज ने इन सभी को नृत्यकला का प्रशिक्षण दिया। भूषण कुमार ने सरोज खान बायोपिक फिल्म के लिए उनके बेटे राजू खानबेटी सुकैना खान और हिना खान से राइट्स प्राप्त किये हैं। महज 3 साल की उम्र से हिंदी फिल्म जगत में अपने कदम रखने वाली सरोज खान ने यहां बड़ा नाम कमाया। अपने करियर में उन्होंने तकरीबन 3500 गानों की कोरियोग्राफी की है। अपने काम के लिए वो 3 बार राष्ट्रीय पुरस्कार से भी सम्मानित हो चुकी हैं। उनकी बायोपिक फिल्म को लेकर उनके बेटे राजू खान ने कहामेरी मां को डांस बेहद पसंद था और हमने देखा जिस तरह से उन्होंने अपना जीवन इसमें लगा दिया। मैं खुश हूं कि मैंने इसमें अपना जीवन लगा दिया। मेरी मां को इंडस्ट्री से काफी प्यारा और सम्मान मिला और ये मेरे परिवार के लिए गर्व की बात है कि उनके सफर को दुनिया के सामने पेश किया जाएगा। मैं खुश हूं कि भूषण कुमार जी ने उनके जीवन पर बायोपिक फिल्म बनाने का फैसला किया।

टीसीरीज के मालिक भूषण कुमार ने आज बड़ी घोषणा करते हुए बताया कि वो देश की पहली महिला कोरियोग्राफर रह चुकी सरोज खान के के जीवन पर एक बायोपिक फिल्म बनाने जा रहे हैं। आज सरोज खान की पुण्यतिथि के मौके पर उन्होंने उनके फैंस के लिए ये बड़ी घोषणा की है। श्रीदेवी से लेकर माधुरी दीक्षित तकसरोज ने इन सभी को नृत्यकला का प्रशिक्षण दिया। भूषण कुमार ने सरोज खान बायोपिक फिल्म के लिए उनके बेटे राजू खान बेटी सुकैना खान और हिना खान से राइट्स प्राप्त किये हैं।

Fig. 3: Sample summary generated by our model

formulation allows for lower variance in the final predictions made over the individual models.

Another approach explored in this work is the use of a weighted hard voting scheme. This allows us to assign unequal values to the votes cast by the individual models.

$$Final\_summary = \alpha*(Model_1)+\beta*(Model_2)+\gamma*(Model_3)$$

Here, Model 1 corresponds to the SBERT classification model; Model 2 is the k-means centroid model and Model 3 is based on the statistical and linguistic properties of the text. The exact weights associated with the models are hyper-parameters. For our work, we chose α = 0.4, β = 0.3 and γ = 0.3. This allotment of weights ensures that selections from our best performing model (MuRIL embedding based classification) are taken into consideration except in cases when Model 2 (k-means clustering) and Model 3 (statistical model) select the same candidate, which is also different from the picking of Model 1.

V. Preprocessing and Experimental evaluation

The common preprocessing step across all three models is the breakdown of the article into its component sentences. Following that, we remove sentences from the article below a specific character limit. In this work, we removed all sentences below 15 characters as we observed that these sentences had a negligible presence in the oracle summaries. In the case of the first model, the sentences are then assigned a label of 0 or 1 based on their presence in the summary. Since, in a typical case, the number of sentences not in the article far exceeds the number of sentences in the article, data is undersampled to handle the class imbalance.

Model	**Accuracy**
Classification model	0.71
Centroid Model	0.67
Linguistic features based model	0.62
Ensemble	0.76

TABLE II: Accuracy of the models in our ensemble

	ROUGE-1	**ROUGE-2**	**ROUGE-L**
Oracle Summary	39.53	27.19	36.80

TABLE III: Rouge scores for our oracle summary against the golden summary

Only in the third model, the dimensionality of the data is reduced by removing stop words from the corpora. The words in the sentences are also further stemmed to arrive at the root forms of the terms.

For our implementation, we use PyTorch and the "MuRIL-base-cased" from huggingface as our language model. The MuRIL model was also finetuned over our custom dataset as is described on the model's TFhub page [4]. The model is trained over 5 epochs with a batch size of 16 on Google Colab Pro over NVIDIA's T4 GPU for around 20 hours.

For the evaluation of our results, we choose the accuracy metric. It is defined as the number of summary sentences extracted matching the sentences in the golden summary / The total number of sentences extracted by our model.

Another popular evaluation metric for summarization is the ROUGE score. ROUGE-N measures the count of matching

[4]https://tfhub.dev/google/MuRIL/1

"n-grams" between the reference text and our model-generated text. Another variation of this is ROUGE-L, which measures the Longest Common Subsequence between the reference and model output. However, the ROUGE scores cannot directly evaluate the Oracle summary since these summaries are directly generated from the golden abstractive summaries. The oracle summaries are the upper limit of our model's performance since they equate to the article's top scoring ROUGE sentences. Therefore, the accuracy score between the Oracle summary and our model output proves to be the ideal metric to validate our performance.

VI. Conclusion and Future work

In this paper, we explored how to use the MuRIL language model for extractive summarization in Hindi. We proposed two different models that used the sentence embeddings generated from the language model for classy the outputs as part of the summary or not and for k-means clustering respectively. We also incorporated the statistical and linguistic properties of the text in our third model. These models can be used individually to achieve reasonable performance on the task or be used in a hard-voting-based ensemble. This is the first research exploring the use of BERT-based models for summarization for resource-lean Indian regional languages. We performed our experimentation on a custom dataset created from news articles and reported our accuracy scores which we hope will serve as benchmarks for all future research in this field.

Future work would involve exploring other avenues to deal with class imbalance intrinsic to our task of selecting a small subset of sentences for a large document. We employed undersampling in this work, but other techniques like SMOTE or weighted loss functions could also be used. Another avenue of further research would be using different MuRIL layers to derive the embeddings for the k-means centroid approach.

References

[1] H.P. Lun The automatic creation of literature abstracts IBM J. (1958), pp. 159-165

[2] Elena Lloret and Manuel Palomar. 2012. Text summarization in progress: a literature review. Artificial Intelligence Review 37, 1 (2012), 1–41.

[3] Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. In Mining Text Data. Springer, 43–76.

[4] Horacio Saggion and Thierry Poibeau. 2013. Automatic text summarization: Past, present and future. In Multi-source, Multilingual Information Extraction and Summarization. Springer, 3–21.

[5] Karen Spärck Jones. 2007. Automatic summarising: The state of the art. Information Processing and Management 43, 6 (2007), 1449–1481.

[6] Steinberger J., Ježek K. (2004) Text Summarization and Singular Value Decomposition. In: Yakhno T. (eds) Advances in Information Systems. ADVIS 2004. Lecture Notes in Computer Science, vol 3261. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-540-30198-1_25

[7] Chandra Rakesh, Deepak Sahoo, B. Sahoo, M. Swain. Text Summarization Using Term Weights January 2012International Journal of Computer Applications 38(1):10-14 DOI: 10.5120/4570-6731

[8] El-Khair I.A. (2009) Term Weighting. In: LIU L., ÖZSU M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-39940-9_943

[9] Manisha Gupta, Naresh Kumar Garg Text Summarization of Hindi Documents Using Rule Based Approach. Conference: 2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE). September 2016 DOI: 10.1109/ICMETE.2016.104

[10] Manisha Gupta, Dr.Naresh Kumar Garg. Text Summarization of Hindi Documents using Rule Based Approach. 2016 International Conference on Micro-Electronics and Telecommunicatio n Engineering (ICMETE). Sept. 2016. DOI: 10.1109/ICMETE.2016.104

[11] Vijay, S., Rai, V., Gupta, S., Vijayvargia, A., & Sharma, D. (2017). Extractive text summarization in hindi. 2017 International Conference on Asian Language Processing (IALP), 318-321.

[12] Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In Proceedings of the ACL Conference.

[13] Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In Proceedings of the NAACL Conference.

[14] Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In Proceedings of the ACL Conference.

[15] Liu, Yang and Mirella Lapata. "Text Summarization with Pretrained Encoders." EMNLP/IJCNLP (2019).

[16] Hovy, E. and Lin, C. Y. Advances in Automatic Text Summarization Mani, I. and Maybury, M. T., editors, 81–94. 1999. MIT Press.

[17] Liu, Yang. "Fine-tune BERT for Extractive Summarization." ArXiv abs/1903.10318 (2019)

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

[20] Antoun, Wissam and Baly, Fady and Hajj, Hazem, AraBERT: Transformer-based Model for Arabic Language Understanding, LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020

[21] Raphael Scheible, Fabian Thomczyk, Patric Tippmann, Victor Jaravine, Martin Boeker, GottBERT: a pure German Language Model,https://arxiv.org/abs/2012.02110.

[22] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. arXiv preprint arXiv:2003.11080.

[23] Reimers, Nils and Gurevych, Iryna, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, ssociation for Computational Linguistics, https://arxiv.org/abs/1908.10084.

[24] Dunlavy, D. M., O'leary, D. P., Conroy, J. M., and Schlesinger, J. D. (2007). QCS: A system for querying, clustering and summarizing documents. Information Processing and Management, 43(15), 1588–1605.

[25] Cai, X., Li, W., and Zhang, R. (2013). Combining co-clustering with noise detection for theme-based summarization. ACM Transactions on Speech and Language Processing, 10(4), Article 16), 1–27.

[26] Rasim M. Alguliyev,Ramiz M. Aliguliyev,Nijat R. Isazade,Asad Abdi,Norisma Idris, COSUM: Text summarization based on clustering and optimization, https://doi.org/10.1111/exsy.12340.

[27] Derek Miller, Leveraging BERT for Extractive Text Summarization on Lectures, https://arxiv.org/abs/1906.04165

[28] Chetana Thaokar and Latesh Malik. Test model for summarizing hindi text using extraction method. Information & Communication Technologies (ICT), 2013 IEEE Conference on, 1138–1143. 2013.

[29] Kaur, Dawinder and Kaur, Rajbhupinder Automatic Summarization of Text Documents Written in Hindi Language 2014

[30] Manisha Gupta, Dr.Naresh Kumar Garg, Text Summarization of Hindi Documents using Rule Based Approach, 2016 International Conference on Micro-Electronics and Telecommunication Engineering

[31] Rafael Ferreira, Luciano de Souza Cabral, Rafael Dueire Lins, Gabriel Pereira e Silva, Fred Freitas, George DC Cavalcanti, Rinaldo Lima, Steven J Simske, and Luciano Favaro. 2013. Assessing sentence scoring techniques for extractive text summarization. Expert systems with applications 40, 14 (2013), 5755–5764.

[32] Daraksha Parveen, Mohsen Mesgar, and Michael Strube. 2016. Generating Coherent Summaries of Scientific Articles Using Coherence Patterns.. In EMNLP. 772–783.

[33] Daraksha Parveen, Hans-Martin Ramsl, and Michael Strube. 2015. Topical coherence for graph-based extractive summarization. (2015).
[34] Hayato Kobayashi, Masaki Noguchi, and Taichi Yatsuka. 2015. Summarization Based on Embedding Distributions.. In EMNLP. 1984–1989.
[35] Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. arXiv preprint arXiv:1603.07252 (2016).
[36] Victor Dibia, https://victordibia.com/blog/extractive-summarization, 2021. Accessed - 05/10/2021
[37] Simran Khanuja and Diksha Bansal and Sarvesh Mehtani and Savya Khosla and Atreyee Dey and Balaji Gopalan and Dilip Kumar Margam and Pooja Aggarwal and Rajiv Teja Nagipogu and Shachi Dave and Shruti Gupta and Subhash Chandra Bose Gali and Vish Subramanian and Partha Talukdar, MuRIL: Multilingual Representations for Indian Languages, https://arxiv.org/abs/2103.10730
[38] Ramesh Nallapati and Feifei Zhai and Bowen Zhou, SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents, arXiv, 2016