Deep Learning Model for Abstractive Automatic Text Summarization in Hindi

Vishwa Sharma, Rakesh Kumar Tiwari, Ishan Awasthi, Shivendra Kaushal
Prof. Rushali Patil Department of Computer Engineering Army Institute of Technology Pune 411015

Abstract:
Text summarization is a process in which long texts are compressed and condensed into smaller summaries. Only the crux ideas of the document are fetched from the main document and included in the final piece, which is cohesive. As the amount of data is soaring exponentially. The need for a tool that summarizes text specifically for Indian languages is also pertinent. Using a variety of techniques, we strive to construct both extractive and abstractive approaches for text summarization of Hindi text in this research. The abstractive method is based on seq-to-seq networks and the attention model. A summary of all Indian regional languages cannot be generalized by a single approach. This is so that each language may be treated separately because every language has unique linguistic characteristics.
Keywords: Recurrent Neural Network, long short-Term Memory, Term Frequency, Inverse Document Frequency, Word Embedding, Word Vector, Continuous Bag of words.
DOI: 10.24297/j.cims.2023.6.9

1. Introduction

Overview

Text summarising methods may be roughly divided into two parts, extractive summarization and abstractive summarization, based on the methodologies utilised. The technique of text summarization by extractive method includes words and phrases into the newly formed summary after selecting them from the source text or documents. The text’ s key ideas serve as the foundation for the summary. To find the necessary phrases and sentences, extractive techniques use statistical factors including position of sentence, numerical data, grammatical subjects like nouns, topic token frequency, normalized sentence length, etc.

2. Objectives

- To design a method for summarizing Hindi text that can summarize a single document using certain Extractive and Abstractive techniques which include seq-2-seq [10] model, Page Ranking Algorithm.

- Parts of the original text are condensed and paraphrased as part of the abstraction method. When abstraction is employed for text summarization in deep learning tasks, the grammatical issues of the extractive approach can be resolved.

3. Related Works

A. DOUBLE ATTENTION POINTER NETWORK BASED TEXT SUMMARIZATION [14]

Z. Li, Z. Peng, S. Tang, C. Zhang and H. Ma [14], described the main objective of the papers where the main points of the text should be covered in a solid document summary. An encoder-decoder based on a double attention pointer network [14] was suggested in the paper that was submitted. The self-attention mechanism, pointer network, and soft attention provide core material that is more coherent, and the combination of these two processes produces accurate summaries, according to the approach utilised in the study for text summarising. By adopting the improved coverage mechanism, the repetition problem is also resolved and the quality of the summaries that are produced is raised. Scheduled sampling and reinforcement learning (RL) are coupled together to produce new training methods for the model’s optimization. The paper’s conclusion is that there is a decrease in the possibility of repeating and an increase in the performance of summarising.

B. TWO-STAGE TRANSFORMER-BASED APPROACH [6]

Su, Wu and Cheng, [6] with respect to the standard BERT model, developed this model which introduces a combination of text segmentation and 2-stage abstractive model. The former divides the given input text in various segments, each of which is then inserted into the 2-stage model. A summary for each segment is received and then the concatenation of the the summary is done. Very limited Chinese database is available for the given. In both objective and subjective [6] assessments, the models in the suggested system performed as well as, or even better than, the competition. On the LCSTS dataset, the suggested variable-length abstractive summarization algorithm has an accuracy rating of up to 70.0%. Finally, an accuracy of maximum 70.0% was revealed on LCSTS dataset by a human subjective evaluation proposed by a multilength abstractive summarization method.

C. SEMANTIC REWARD-BASED REINFORCEMENT LEARNING FOR OPTIMIZED ABSTRACTIVE TEXT SUMMARIZATION [5]

The following paper [5] talks about how a neural network produces a text summary be using an encoder and decoder structure in the model that are currently there for abstractive text summarization and talks about how these methods are not deemed effective on the current metrics system. The paper [5] introduces a methodology for improving the quality of the summary produced by the system [5] by introducing a reward function which will be used in the summarization of the text which is hinged on reinforcement learning. The study modifies the ROUGE metric function and produces two function which are ROUGE – SIM and ROUGE – MD. the general paper states that when the reward function is used the model generated has the return functions which are less biased than the sequence to sequence based model the [5] paper states that the following proposed model almost annihilates the grammatical errors present in the paper and thus is considered better than many of the already present model such as sequence to sequence and extractive model. The paper [5] helps to understand the various metric system used and modified which can be used in the project implementation for Hindi as well and thus is considered very useful research for automatic text summarization.

D. ABSTRACTIVE TURKISH NEWS SUMMARIZATION BASED ON DEEP LEARNING [17]

A deep learning-based encoder-decoder model [17] was employed to generate Turkish news headlines through an abstractive text summarization method. The system was trained with recurrent neural networks using FastText [15]-generated word embeddings of news texts. Different training scenarios were tested by training the system with the first sentence, first two sentences, and full-text of each news article. The system’s success was measured using ROUGE score and semantic similarity score. The experimental results indicated that the model trained with full-text of news achieved the highest performance compared to the other models.

E. SIMILARITY MEASUREMENT OF BENGALI SENTENCES FOR ABSTRACTIVE TEXT SUMMARIZATION. [18]

The summarization of text is an extensive subject of study in natural language processing which aims to compress large text documents while retaining their primary meaning. Proper text analysis is necessary to comprehend the meaning of larger texts, which is crucial for building a better text summarizer. Abstractive text summarization methods generate summaries that may or may not be present in the source text document. These methods learn from human-provided summaries to produce a text summary. Sentence similarity, which gauges how similar phrases or words are, is an essential indicator for assessing how well a text summarizer performs. In this essay, we investigate several sentence similarity techniques and suggest a technique for locating a more accurate Bengali abstractive text summarizer. To measure similarity, we used human-generated and machine-generated Bengali short text summaries obtained from online and social media. We applied some of the approaches used in English sentence similarity measurement to our Bengali text and achieved satisfactory results. We collected data from various sources, created a summary of the texts, and pre-processed the summary text to generate summaries using our abstractive text summarization model. Our proposed method for measuring similarity between summary sentences proved effective, with all cases yielding optimal results.

4. Proposed Methodology

Abstractive text summarization is a complicated natural language processing problem that includes providing a short and informative summary of a given piece of text while keeping its core content. In this proposed methodology, we will present a step-by-step approach for abstractive text summarization for Hindi language using an attention-based transformer and encoder-decoder architecture.

1) Pre-processing of Text The first step in our proposed methodology is the pre-processing of the input text. The input text is first cleaned to remove any unwanted characters, symbols or punctuations. Then, the text is tokenized into words and sentences using a tokenizer. The next step is to remove any stop words, which are commonly occurring words that do not add significant meaning to the text. Finally, the input text is lemmatized to reduce the inflectional forms of words to their base or root form
2) Word Vector Generation In this step, we generate word vectors for the pre-processed text. Word vectors are numerical representations of words that capture their semantic meaning. We use a pre-trained word embedding model such as Word2Vec [16] or fastText [15] to generate the word vectors. The word vectors are used as input to the attention-based transformer.
3) Attention Based Transformer The attention-based transformer [4] is a neural network architecture that has shown great results tasks that involves processing of natural language like analysis of sentiments, text summarization, and machine translation. In this step, we take the input text and encode it using an attention-based transformer and after that a summary is generated. The transformer architecture consists of several layers of encoder and decoder. After processing the input text, the encoder layer creates a set of context vectors [15] that reflects the semantic meaning of the input. Next, using the context vectors generated in the previous step decoder layer generates a summary which is going to be our final result. While generating the summary, attention mechanism is used to put more emphasis on the relevant parts of the given input text.

4) Encoder-Decoder In this architecture [8] we first take given text for summarization and encode it into a vector representation of fixed length. This vector representation is then decoded into a summary. In this stage, we carefully create a summary using an encoder-decoder architecture. The pre-processed text is provided to encoder as input, which then creates a set of context vectors. Then a summary is generated by the decoder using the context vectors created by encoder. To put more emphasis on the relevant parts of the given input text while generating the summary we use attention mechanism.
5) Evaluation The final step in our proposed methodology is the evaluation of the summary generated. We use the ROUGE-L, ROUGE-1 and ROUGE-2 metrices for evaluating the quality of the summary generated collectively known as ROUGE metrices. The ROUGE metrices determines the similarity between the output summary and the original summary or the summary taken for reference. The ROUGE metric consists of three scores: ROUGE-L, ROUGE-1 and ROUGE-2. ROUGE 1 evaluates the overlapping of the generated summary and the summary taken for reference. The overlapping bigrams between the reference summary and the summary produced is measured by the ROUGE-2 score. The largest subsequence which is common between the output summary and the summary taken for reference is measured by the ROUGE-L score.

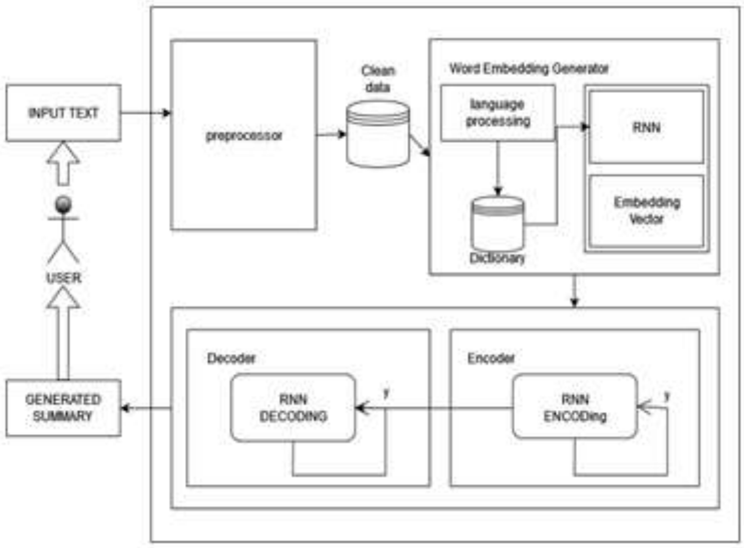

Fig. 1. System Diagram

5. Implementation Details

A. DATA PREPARATION

- Data pre-processing is a critical step in natural language processing (NLP) that involves cleaning, transforming, and encoding raw text data to prepare it for analysis or modelling. Following are some common data pre-processing techniques used in the Project:
- Text Cleaning: Text cleaning involves removing unwanted characters, such as punctuation and special symbols, from the text data. This can be done using regular expressions or specific libraries in programming languages such as Python. Text cleaning can also include removing stop words (such as" the" and" a") and performing stemming or lemmatization for reduction of inflectional forms of words into the base form.
- Tokenization: The process which breaks the given text into individual words or sub-words which are known as tokens, is known as tokenization. This is done to enable further analysis or modelling of the text data at the token level.
- Part-of-Speech Tagging: In part-of-speech (POS) tagging, each token in a text sequence is marked with the appropriate POS tag, such as a noun or a verb or an adjective or adverb. This is helpful in tasks like text classification or named entity recognition since it helps to grasp the grammatical structure of a sentence or document.
- Named Entity Recognition: The process of recognizing and categorising named entities in text which can be a person a company of a location is generally known as entity recognition. This can be done using rule-based approaches or using machine learning models trained on annotated data.
- Word Embeddings: Word embedding is a technique for representing words as dense vectors of real numbers. Word embeddings can be learned from large corpora using neural network models and are used to represent words in NLP Applications such as machine translation and language modelling.
- Padding and Truncation: Padding and truncation are techniques used to ensure that all sequences in a dataset have the same length. This is often necessary in machine learning applications, where data is typically represented as numerical vectors of fixed length.

B. MATHEMATICAL MODEL

Mathematical representation

Let the original document be a string D

Let the summary of the document be string A

Let the length of A be L

The for any string Ai to be the summary of D following two conditions should be met

$Condition1: |length\ (Ai) - L| <= \delta$

$Condition2$: topicset(A_i) ⊂ topicset(D)

Any summary following the two conditions will be a valid summary but there are many summaries which following the valid condition let represent all those summaries with a set S

Where,

$S = A_{i:}$ |length (A_i) − L| ≤ δ & topicset(A_i) ⊂ topicset(D)

The task of automatic text summarization is to find the summary Ai from s which represent or resembles D closely in a contextual manner To do that lets define a function dis () for calculation of the semantic distances between two strings the minimum the value of D is the closely the strings will resemble each other

Therefore,

$Condition3$: A^* = min(dis(topicset(D), topicset(A_i))) where $A_i \in S$ where A^* is the ideal final summary

Assuming that all knowledge may be stated as a series of metaknowledge, where a metaknowledge is a fundamental understanding that cannot be subdivided into lesser understandings, metaknowledge can be assembled into any understanding. Let ϕ be the aggregate of all the metaknowledge therefore it can be represented as $\phi = \{t1, t2, ...tn\}$ where ti is a metaknowledge Therefore:

$topicset(D) = \{td1, td2, ...tdn\}$ where $tdj \subset \phi$ for $1 < j < n$

$topicset(A) = \{ta1, ta2, ta3, ...tam\}$ where $tai \in \phi\ for 1 < i < m \leq n$

Due to condition 1 following definition should hold the condition which is :

$Condition4$: $tak \in topicset(D)\ \forall\ tak \in topicset(A)$ where $1 \leq k \leq m$

Let the weights of metaknowledge of the topicsets be defined as follow:

$weightset(D) = \{wd1, wd2, wd3, ...wdn\}$ where $wdj \subset$

$R \forall\ 1 < j < n$

$weightset(A) = \{wa1, wa2, wa3, ...wam\} textwherewak \subset$

$R \forall\ 1 < k < m \leq n$

We can define the distance between the best summary of a document and a document as
$dis(topicset(A*), topicset(D))$

Replacing the respective topics dis(td1,td2,td3,. . . . .tdn, ta1,ta2,ta3,. . . . .tam)

n∑(j=0) wdj *d(tdj,topicset(Ai)) + m∑k=0 wak*d(tak,topicset(D))

Where function d is a distance function where $d(e,S)$ is defined as

0 $if\ e \in S$

1 $if\ e \in S$

Therefore, the equation reduces to

A *= min(n∑j=0 wdj * d(tdj,topicset(Ai)))

Therefore, a good summary is a summary where the topics in document D but not in A are minimum hence defines the whole model of automatic text summarization mathematically.

V. ALGORITHMS

A. TEXT PRE-PROCESSING

In Hindi text summarising, the preparation phase is critical in preparing the material for the summation process. To clean and prepare the text for analysis, it employs several approaches such as tokenization, stop word removal, stemming, part-of-speech (POS) tagging, and lemmatization. The purpose of pre-processing is to eliminate any extraneous information, lower the size of the text, and standardise the content for improved summarization performance.

Tokenization

The process of breaking down a sentence into individual words or phrases is known as tokenization. [13] In Hindi, tokenization is particularly challenging due to the complex nature of the script and the use of multiple scripts. Tokenization can be performed using various techniques such as rule-based, statistical, or machine learning methods.

Stop words Removal

Stop words are words that do not add significant meaning to the text and are commonly used in nlp tasks such as text summarization. We start by eliminating all of the commonly used stop words from the text. These are the words that don’t have any semantic meaning and don’t contribute any pertinent information to the text; they are thus eliminated from the input text. If they are left in, they could carry greater weight than the words that are relevant if they are left in.

Sentence Segmentation

Next comes segmentation, in this process we break the sentences and find the frequency of words in all the sentences. In Hindi language'|' marks the end of the sentence which is used in segmenting. The phrases are now divided into individual words for tokenization with the use of commas, special symbols, and spaces.

POS Tagging

It is identifying the grammatical function of individual word in a sentence. In Hindi, POS tagging is particularly a challenge because of the script's complexity and the use of several scripts. POS tagging can be performed using various techniques such as rule-based, statistical, or machine learning methods.

Lemmatization

Lemmatization is the reduction of words to their most fundamental form. Lemmatization, which reduces words to their most basic form in Hindi, can assist to lower the percentage of unique words in a text and boost the effectiveness of the summary procedure.

B. WORD VECTORIZATION

Word vector is a numerical representation of a natural language word which is used for natural language processing as machines cannot comprehend. word vectorization is a process to map words into

1) Vectorization Methodologies: there are two types of vectorization techniques which are continuous bag of words [16] and skip grams. There are two types of continuous bag of words models which are for one word context and for multiple word context,

- In CBOW [16] the neural network for the following tasks is trained – We consider a corpus and for every word in the corpus vocabulary we try to pair or predict the next word that can be associated with that word, weights produced at the hidden layer would be the word embedding
- A skip gram model also uses a fake task approach but in nature it is a complete contrast to CBOW [16] model in skip gram model [15] also a fake task is used for this model the

task used is Provided a corpus for each and every vocabulary present in that corpus we try to calculate the probability distribution of each word of being the next word of the vocabulary in the sentence or the corpus

2) Word Vectorization Algorithm: python library fasttext [15] can be used to create vector word embeddings following and word vectors can be generated using following algorithms

```
import fasttext.util
ft = fasttext.loadmodel(path)
to get the word embedding following command can be used
ft.getword vector(word).shape[ 0 ]
algorithm to create dictionary is given below
vectorize = model.getwordvector( wordRand ).shape[ 0 ]
w_matrix = np. zeroes ((DICT SIZE, vectorize))
unk words = [ ]
textwords = [ ]
for (word, index) in tokenizer. wordindex. items ():
if index < DICT SIZE:

if word in model. words:

w_matrix [index] = model. getwordvector (word)
textwords. append (word)
else:
w_matrix [ index] = np. random. rand (1, vectorize)
unk_words. append (word)
return matrix
```

3) Summarization Using Transformers: after creating word vectors the summarization of the text takes place, for abstractive summarization a transformer is used which uses a basic encoder and decoder architecture for generating summaries, a transformer is an attention based neural network which generates summary in two different steps which are encoder and decoder defined as follow:

- Encoder: The Encoder component of the Transformer takes in the given text and creates a set of representations of the given input text which are encoded. These encoded representations are then passed on to the Decoder component. N identical layers are stacked to make the Encoder. Furthermore, each individual layer consists of two sub layers which comprises of a multiple head Attention layer and a feed forward fully position wise connected layer which allows the encoder to have attention on different elements of the text provided in the input to capture its contextual meaning.

- Decoder: The Decoder components takes the encoded form of the input text then processes it to produce the summary of the input text. N identical layers are stacked to make the Decoder. Furthermore, there are three sub layers present in each individual layers which are: Masked Multiple Head Attention layer, a Multiple Head Attention layer, and a Fully Connected position wise Feed-Forward Network layer. The Masked Multiple Head Attention layer in the Decoder ensures that during training, the decoder can only attend to positions that have already been generated. This is crucial to avoid the model cheating by scanning the text before training. In Decoder the Multi-Head Attention layer allows it to attend to the different parts of the encoded input text, and the Fully Connected positional wise Feed-Forward Network layer produces the final summary output.

6. Testing Methodologies And Results

A. EVALUATION MATRICES

In the following project certain evaluation metrics are taken into account which are important to evaluate the performance of the model, these evaluation matrices broadens the understanding of the performance and efficiency of the model which later helps in result evaluation.

The evaluation matrices used to assess the effectiveness of the model created for the project are listed below:

- Precision: It is defined as the fraction of true positives in the total number of positive samples predicted by the model. A high accuracy grade suggests that the model produces a minimal amount of false positive predictions.

$$\text{Precision} = \frac{\text{True Positives}}{\text{False Positives} + \text{True Positives}}$$

- Recall: The measures of the number of true positives among the total number of true positive samples is known as recall. If the recall score is high, it would imply that the given model can identify most of the positive samples.

$$\text{Recall} = \frac{\text{True Positives}}{\text{False Positives + True Negatives}}$$

- F1-Score: The weighted harmonic mean of recall and precision, which takes into account both metrics. It is defined as 2 * (recall*precision) / (recall + precision). F1-score provides a balanced assessment of the model's performance and is often used when precision and recall are both important.

$$F1 = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

- ROUGE: ROUGE is an abbreviation for Recall-Oriented Understudy for Gisting Evaluation, it is a prominent collection of measures for measuring the quality of summarization models. ROUGE measures and expresses the amount of overlap between the model's summary and one or more reference summaries in terms of n-grams. ROUGE can account for the degree of coverage of the input material, as well as the level of abstraction and informativeness of the summary, by using n-grams, which are continuous sequences of words of a specified length.

ROUGE has several variants, including ROUGE-L, ROUGE-2, and ROUGE-1, each of which focuses on a particular amount of n-gram overlap. The ROUGE 1 score measures the unigrams which overlaps between the output summary and the summary taken as reference. The ROUGE 2 score measures th bigrams which overlaps between the output summary and the summary which is taken for reference. The ROUGE-L score measures the biggest common subsequence between the reference summary and the created summary.

To calculate the ROUGE scores, we first tokenize the reference summary and the generated summary in n-grams. Then, it is calculated how much of the generated summary's n-grams coincide with those of the reference summaries, and the resulting values are used to compute various statistics, such as recall, precision, and F1-score. These statistics are then merged to provide the final ROUGE score, which shows the degree of similarity between the reference and created summaries.

7. Results

- ROGUE-1 Score for the following model are as follows:

F – *score:* 0.666

Precision: 0.857

Recall: 0.545

- ROGUE-2 Score for the following model are as follows:

F – *score:* 0.235

Precision: 0.285

Recall: 0.2

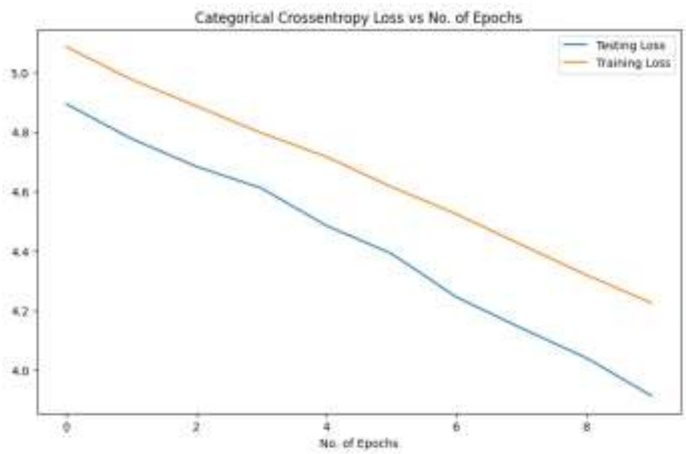

Fig. 2. Categorical Cross entropy loss vs no of epochs

- ROGUE-L Score for the following model are as follows:

G – *score:* 0.696

Precision: 0.847

Recall: 0.549

Based on the ROUGE scores, the summarization model evaluated in this study performed reasonably well in terms of ROUGE-1 scores, but had lower performance for ROUGE-2. An F-score of 0.666, precision of 0.857, and recall of 0.545 for both ROUGE-1 and ROUGE-L was attained by the model. However, for ROUGE-2, the F-score was only 0.235, with precision of 0.285 and recall of 0.2.

These findings suggest that the model could benefit from improvements in capturing longer n-grams in order to improve ROUGE-2 performance. Additionally, it may be useful to explore additional techniques such as attention mechanisms or more advanced deep learning architectures to further enhance the model’s performance.

The difference between the predicted summary and the true summary for a given input text is measured by the cross-entropy loss. The loss function typically decreases during training as the model learns to better approximate the true summary.

In Fig. 2. curve for the training data would indicate that the model is effectively learning from the training data and improving over time. As the number of epochs increases, the model is able to better approximate the true summary, resulting in a decrease in the loss.

For the testing data, the curve would indicate that the model is generalizing well to new data. This means that the model is able to perform well on the training data as well as on the unseen data. A decrease in the testing loss over time suggests that the model is not overfitting to the training data and is instead learning useful patterns in the input data.

Overall, for both the training and testing data the curve suggests that the text summarization model is learning effectively and generalizing well. However, it’s important to carefully monitor the curves during training to avoid overfitting, and to validate the model’s performance on a separate validation dataset to ensure that it is performing well on unseen data.

8. Conclusion

In this research paper, we used transformer-based model for developing Hindi text summarization application and identified the shortcomings of various existing text summarization approaches for the Hindi language. We have presented a novel approach for Hindi text summarization, which achieved competitive results in terms of ROUGE scores.

The transformer-based model used in our approach was able to effectively learn the important features of the input text and generate a concise and informative summary. Our results demonstrate that our model was able to achieve a 0.666 F-score on ROUGE-1, a 0.235 F-score on ROUGE-2, and a 0.696 F-score on ROUGE-L, which implies that our model is able to effectively summarize Hindi text.

We are certain that the model and technique proposed in the research may be enhanced even further by combining new language variables and using a bigger dataset. Furthermore, additional research into various transformer-based architectures and pre-trained language models may yield even better results for Hindi text summarization.

Overall, our findings show that transformer-based models have the potential to be employed in abstractive text summarization for Hindi, which has crucial implications for increasing the accessibility and usability of huge volumes of Hindi text data. We believe that our work will stimulate more research into the use of deep learning techniques for Hindi text summarization since it establishes the groundwork for future study and advancement in this field.

References

1. Elsaid, A. Mohammed, L. F. Ibrahim and M. M. Sakre," A Comprehensive Review of Arabic Text Summarization," in IEEE Access, vol. 10, pp. 38012-38030, 2022, doi: 10.1109/ACCESS.2022.3163292.
2. Gupta, Rahul, I. Khatri and Monika," A Review on Various Techniques of Automatic Text Summarization," 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2020, pp. 1379-1384, doi: 10.1109/ICECA49313.2020.9297641.
3. A. Hern´ andez-Casta´ neda, R. A. Garc˜ ´ıa-Hernandez, Y. Ledeneva and C.´ E. Millan-Hern´ andez," Extractive Automatic Text Summarization Based´ on Lexical-Semantic Keywords," in IEEE Access, vol. 8, pp. 4989649907, 2020, doi: 10.1109/ACCESS.2020.2980226
4. Umakant Dinkar Butkar, et.al "Accident Detection and Alert System (Current Location) Using Global Positioning System" JOURNAL OF ALGEBRAIC STATISTICSVol. 13 No. 3 (2022) e-ISSN: 1309-3452. https://publishoa.com/index.php/journal/article/view/591.
5. Jung, Sukhwan & Segev, Aviv. (2022). Semantic Similarity Analysis between Future Topics and Their Neighbors in Topic Networks for Network-based Topic Evolution. 5952-5961. 10.1109/BigData55660.2022.10020287.
6. M. -H. Su, C. -H. Wu and H. -T. Cheng," A Two-Stage TransformerBased Approach for Variable-Length Abstractive Summarization," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 2061-2072, 2020, doi: 10.1109/TASLP.2020.3006731.

7. R. Alqaisi, W. Ghanem and A. Qaroush," Extractive Multi-Document Arabic Text Summarization Using Evolutionary Multi-Objective Optimization With K-Medoid Clustering," in IEEE Access, vol. 8, pp. 228206-228224, 2020, doi: 10.1109/ACCESS.2020.3046494.
8. Mr. Umakant Dinkar Butkar, Manisha J Waghmare. (2023). Novel Energy Storage Material and Topologies of Computerized Controller. Computer Integrated Manufacturing Systems, 29(2), 83–95. Retrieved from http://cims-journal.com/index.php/CN/article/view/787
9. S. G. Jindal and A. Kaur," Automatic Keyword and Sentence-Based Text Summarization for Software Bug Reports," in IEEE Access, vol. 8, pp. 65352-65370, 2020, doi: 10.1109/ACCESS.2020.2985222.
10. X. Wang and W. Lu," Automatic Text Summarization Technology of Keyword Replacement Based on Seq2Seq," 2021 7th International Conference on Systems and Informatics (ICSAI), Chongqing, China, 2021, pp. 1-6, doi: 10.1109/ICSAI53574.2021.9664167.
11. Alahmadi, Dimah & Wali, Arwa & Alzahrani, Sara. (2022)." TAAM: Topic-aware abstractive arabic text summarisation using deep recurrent neural networks." Journal of King Saud University - Computer and Information Sciences. 34. 10.1016/j.jksuci.2022.03.026.
12. Mr. Umakant Dinkar Butkar, Dr. Pradip Suresh Mane, Dr Kumar P K, Dr. Arun Saxena, Dr. Mangesh Salunke. (2023). Modelling and Simulation of symmetric planar manipulator Using Hybrid Integrated Manufacturing. Computer Integrated Manufacturing Systems, 29(1), 464–476. Retrieved from http://cims-journal.com/index.php/CN/article/view/771
13. Y. Du and H. Huo," News Text Summarization Based on Multi-Feature and Fuzzy Logic," in IEEE Access, vol. 8, pp. 140261-140272, 2020,
14. doi: 10.1109/ACCESS.2020.3007763
15. Z. Li, Z. Peng, S. Tang, C. Zhang and H. Ma," Text Summarization Method Based on Double Attention Pointer Network," in IEEE Access, vol. 8, pp. 11279-11288, 2020, doi: 10.1109/ACCESS.2020.2965575.
16. F. Zhang, W. Gao, Y. Fang and B. Zhang," Enhancing Short Text Topic Modeling with FastText Embeddings," 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Fuzhou, China, 2020, pp. 255-259, doi: 10.1109/ICBAIE49996.2020.00060.

17. Umakant Dinkar Butkar, Dr. Nisarg Gandhewar. (2022). ALGORITHM DESIGN FOR ACCIDENT DETECTION USING THE INTERNET OF THINGS AND GPS MODULE. Journal of East China University of Science and Technology, 65(3), 821–831. Retrieved from http://hdlgdxxb.info/index.php/JE_CUST/article/view/313
18. E. Karakoc¸ and B. Yılmaz," Deep Learning Based Abstractive Turkish News Summarization," 2019 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 2019, pp. 1-4, doi: 10.1109/SIU.2019.8806510.
19. A. K. M. Masum, S. Abujar, R. T. H. Tusher, F. Faisal and S. A.
20. Hossain," Sentence Similarity Measurement for Bengali Abstractive Text
21. Summarization," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-5, doi: 10.1109/ICCCNT45670.2019.8944571.
22. C. Kwatra and K. Gupta," Extractive and Abstractive Summarization for Hindi Text using Hierarchical Clustering," 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 2021, pp. 1-6, doi: 10.1109/ICSES52305.2021.9633789.
23. K. -Y. Chen et al.," Extractive Broadcast News Summarization Leveraging Recurrent Neural Network Language Modelling Techniques," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 8, pp. 1322-1334, Aug. 2015, doi: 10.1109/TASLP.2015.2432578.
24. M. Yang, C. Li, Y. Shen, Q. Wu, Z. Zhao and X. Chen," Hierarchical Human-Like Deep Neural Networks for Abstractive Text Summarization," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 6, pp. 2744-2757, June 2021, doi: 10.1109/TNNLS.2020.3008037.
25. F. You, S. Zhao and J. Chen," A Topic Information Fusion and Semantic Relevance for Text Summarization," in IEEE Access, vol. 8, pp. 178946178953, 2020, doi: 10.1109/ACCESS.2020.2999665.
26. S. Ghodratnama, A. Beheshti, M. Zakershahrak and F. Sobhanmanesh," Extractive Document Summarization Based on Dynamic Feature Space Mapping," in IEEE Access, vol. 8, pp. 139084-139095, 2020, doi: 10.1109/ACCESS.2020.3012539.
27. A. Shimada, F. Okubo, C. Yin and H. Ogata," Automatic Summarization of Lecture Slides for Enhanced Student Preview Technical Report and User Study," in IEEE Transactions on Learning Technologies, vol. 11, no. 2, pp. 165-178, 1 April-June 2018, doi: 10.1109/TLT.2017.2682086.

28. M. Y. Saeed, M. Awais, R. Talib and M. Younas," Unstructured Text Documents Summarization with Multi-Stage Clustering," in IEEE Access, vol. 8, pp. 212838-212854, 2020, doi: 10.1109/ACCESS.2020.3040506.