An Extractive Approach for Automated Summarization of Indian Languages using Clustering Techniques

Kirti Kumari[1,*], Ranjana Kumari[2]

[1]*Indian Institute of Information Technology Ranchi, Ranchi, Jharkhand, India.*
[2]*Katihar Engineering College, Katihar, Bihar, India*

Abstract

Automated summarization for English language has been an active area of research for quite a long time, but very less works have been done for multilingual and regional languages specially for the Indian languages. A summarization system consists of reduction of a text document to generate a new form which conveys the key meaning of the information contained in the text. This technique is useful due to the information overload and useful information being just a tiny fraction of it. Summarization reduces the time overload of reading the entire document and thus improves the readability and efficiency. Summarization can be understood as targeted dissemination of information for too long; didn't read (TLDR) texts. In this paper we have presented an extractive technique of text summarization employing K-means clustering on Indian Language Summarization (ILSUM) 2022 dataset and achieved satisfactory results by the team $IIIT\_Ranchi$.

Keywords
Text Summarization, Clustering, K-Means, Autoencoder, Cosine similarity

1. Introduction

Text Summarization task was very popular in past decades but requirement was different as compared to now a days. At that time storage space was so scarce. Large documents needed to be replaced with summaries for storage. There are now many inexpensive storage devices available compared to that time, but because there is such a vast volume of data readily available, it is necessary to transform that data into knowledge and information [1]. When a user submits a query, the system responds with a collection of lengthy Web pages that contain so much information that it would be nearly difficult for a reader to read them all. Due to the exponential development in the quantity and complexity of information sources on the internet, research in automatic text summarization has attracted a lot of attention in the quickly evolving world of today [2]. Applications of the acquired information and expertise include company management, production control, market analysis, engineering design, and scientific investigation. A summary is a text created from one or much more texts that contains the essential details from the original

---

Forum for Information Retrieval Evaluation, December 9-13, 2022, India

[*] Corresponding author.

kirti@iiitranchi.ac.in (K. Kumari); ranjanak3663@gmail.com (R. Kumari)
0000-0003-3714-7607 (K. Kumari)

CEUR Workshop Proceedings (CEUR-WS.org)

text but isn't even close to half as long as the original text.

The two different techniques to summarization are extractive and abstractive. Abstractive is human way of summarization which requires knowledge outside of the current text too. Abstractive techniques are resource heavy and are yet to reach an accuracy comparable to a human, therefore Extractive techniques which, as the name suggests, generate summary by selecting words from the parent text, dominate the field of text summarization. Utilizing this strategy, summaries are produced by removing essential text segments from the text after analysing variables like term frequency, sentence placement, and many others to pinpoint the sentences that need to be eliminated. Most of the previous works on summarization for monolingual and native English but there is shortage of non-native English and regional language specially Indian languages. Indian Language Summarization (ILSUM) 2022 [1] task given the opportunity for the researchers to cope with code-mixing and script mixing different multilingual Indian languages.

For the current work, we applied Extractive approach for the ILSUM 2022 task by the team $IIIT\_Ranchi$. We tried for all the given tasks which are in English, Hindi and Gujarati languages and achieved the satisfactory results. A very few teams submitted on all the given shared tasks [3, 4].

The paper organized as follows: Section 2 gives a short review on previous work done in the area of summarization. Section 3 discussed the dataset and Section 4 provides the detail about methodology. Section 5 presents the results and finding of our work. Finally, we concluded in Section 6 with highlighted the future scope.

2. Related Work

Automatic text summarization is an active area from the Natural Language Processing (NLP) research community [2, 5]. Many works have been done in the text summarization area can be seen in article [1]. Clustering techniques [6] are mainly used for forming clusters of alike data using unsupervised approach. When we don't have prior knowledge about the data and its labels, but based on some similarity measure either cosine similarity or distance based measures we try to predict the likeliness of data. As discussed in the paper [7] based on centers and the euclidean distance plane sentences can be vectorized and be presented in a hyperplane. Then based on the euclidean distance those vectors can be clustered all together and similar sentences from these clusters thus can be taken for final summary. Another approach based on cosine similarity can be seen in [8]. Where the cosine similarity between two vectors(sentences) is calculated which is presented in the hyperplane. A similar approach can be seen in [9] where they have used vectorization technique based on Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer. Which is based on the number of occurrences of words in a specific sentences. The approach presented by us uses K-means clustering to create extracts from the parent text. The number of clusters depend upon the size of the input text. In broader view though, too less number of clusters are unsuitable as they may change the meaning of the parent text entirely. Similarly large number of clusters would mean that the size of the extracted text is large which contradicts the purpose of summarisation. Keeping in mind the

---

[1] https://ilsum.github.io/ilsum/2022/index.html

Table 1
Distribution of Datasets of different languages

Language	Training sample	Validation sample	Testing sample
Hindi	7958	507	2843
Gujarati	8458	606	3021
English	12566	899	4488

various constraints , the maximum number of clusters has been set to 6 for texts containing more than 15 sentences, 5 for texts having sentences greater than 6. Text less than this size has been left as it is as it is perceived to be already summarised.

The main challenge was posed by the fact that any Indian written snippet is multilingual, in this instance, Hindi, English and Gujarati. A lot of datasets are available for english language but not for Hindi and Gujarati, in addition the script is much more complex adding to the difficulties. Each cluster thus can only be composed of words from the same language.

3. Dataset

In this section, we presented the dataset used in this work. There are variety of datasets are available for any single language like German, English, Frech, Spanish and many more languages but there is no dataset for Indian regional languages. Another problem with those dataset are they are not public and size of the datasets are also small. Therefore, through the ILSUM 2022 dataset [3, 4], organizer committee anticipated to cross the existing gap by creating reusable public datasets for Indian Language Summarization. The dataset comprised of articles collected from various news articles online from different websites. Overall languages wise distribution of dataset is 29% of Gujarati , 27% of Hindi and 44% of English. The detail distribution of dataset can be seen in Table1. More details about the datasets and tasks can be seen in articles [3, 4].

4. Methodology

In this section, we preset the detailed description of our approach used for summarization tasks.

The approach presented uses K-Means clustering to create extracts from the parent text. The number of clusters depend upon the size of the input text. In broader view though, too less number of clusters are unsuitable as they may change the meaning of the parent text entirely. Similarly large number of clusters would mean that the size of the extracted text is large which contradicts the purpose of summarization. Keeping in mind the various constraints, the maximum number of clusters has been set to 6 for texts containing more than 15 sentences, 5 for texts having sentences greater than 6. Text less than this size has been left as it is as it is perceived to be already summarised.

The main challenge was posed by the fact that any Indian written snippet is multilingual, in this instance, Hindi, English and Gujarati. A lot of datasets are available for English language but not for Hindi and Gujarati, in addition the script is much more complex adding to the difficulties. Each cluster thus can only be composed of words from the same language.

The following steps were used:

- Splitting and Tokenization
- Word Vectorization
- Sentence Vectorization
- K-means Clustering

4.1. Splitting and Tokenization

For tokenization and training purposes we have used NLTK library[2] for splitting the words. We have designed custom tokenizer for all English, Gujarati and Hindi languages. Separate splitters have been used for English, Gujarati and Hindi words as end of the sentence delimeters are different in the 3 languages. We have also used sentence tokenizers based on this splitters to tokenize the sentences for further analysis.

4.2. Word Vectorization

After receiving the tokenized words from previous sentences we have trained a Word2Vec model. Each word is converted into a vector which represents its similarity index with other words in the text corpus. This forms basis for distance calculation in cluster formation based upon the trained centers.

4.3. Sentence Vectorization

Now that we have the word vectors we concatenate them together to form the sentences. Now each sentence is flattened into its constituent words to make it suitable for neural net input. The dimensions of the sentences are reduced using Autoencoders. A total of 5 layers including the flattening layer are used. We first reduce the dimensions of sentence and then increase them again to the original which is the basis for Autoencoder approach for lower dimensional representation.

4.4. K-means Clustering

K-Means Clustering is one of the most popular and simple to implement unsupervised machine learning algorithms. Typically, such algorithms make inferences from an unlabelled dataset as opposed to the supervised algorithms. The objective of K means is to group similar data points together on the basis of underlying patterns. K means creates K clusters in which the data points are grouped. In other words, K means identifies K centroid points, all the other data points are identified to one of the centroids on the basis of least distance. The centroids are not fixed and they may change over as frequently as every next iteration, this is done to ensure small centroids and hence tighter clusters.

Finally K-means Clusters the cleaned text. The number of clusters is decided by the size of the input text. On an average we have 13 words per sentence so based on that if we have to

---

[2] https://www.nltk.org/

Table 2
Validation ROUGE F1 Scores for Hindi, Gujarati and English languages

Language	ROUGE-1-F1	ROUGE-2-F1	ROUGE-3-F1	ROUGE-4-F1
Hindi	0.2985	0.1429	0.1118	0.0999
Gujarati	0.1777	0.0866	0.06512	0.0544
English	0.2765	0.1268	0.10349	0.0961

Table 3
Testing ROUGE F1 Scores for Hindi, Gujarati and English languages

Language	ROUGE-1-F1	ROUGE-2-F1	ROUGE-3-F1	ROUGE-4-F1
Hindi	0.3272	0.1739	0.1395	0.1257
Gujarati	0.1763	0.0846	0.06287	0.0532
English	0.3209	0.1861	0.1627	0.1540

take 75 words (as provided by the organizer) in final summary, so we have come up with the solution below:

- Taken 6 for texts having more than 15 sentences
- Taken 5 for texts having more than 6 sentences
- Texts having less number of sentences were left unmodified

5. Result

In this section, we present our results and findings of our approach. Table 2 and Table 3 are showing the results of four different ROUGE F1 Scores of validation and testing sets for Hindi, Gujarati and English languages.

6. Conclusion

Extractive text summarization techniques have been focused upon. K-means further complements the efficiency of extractive techniques as it is quick and suitable for small samples as well as large. Though the performance metric scores were not upto the mark, the method was elegant and hence tweaking the parameters to improve it may prove to be fruitful.

As future scope, the current work can be extended by utilizing the Abstractive approach for the summarization task.

Acknowledgments

We are thank full to our undergraduate students Shaury Srivastava and Pallav Garg for their contribution in prepossessing steps. A very special thanks to academics and Management of Indian Institute of Information Technology Ranchi for providing the necessary resources.

References

[1] W. S. El-Kassas, C. R. Salama, A. A. Rafea, H. K. Mohamed, Automatic text summarization: A comprehensive survey, Expert Systems with Applications 165 (2021) 113679.

[2] P. Mehta, P. Majumder, Effective aggregation of various summarization techniques, Information Processing Management 54 (2018) 145–158. URL: https://www.sciencedirect.com/science/article/pii/S030645731630632X. doi:https://doi.org/10.1016/j.ipm.2017.11.002.

[3] S. Satapara, B. Modha, S. Modha, P. Mehta, Findings of the first shared task on indian language summarization (ilsum): Approaches, challenges and the path ahead, in: Working Notes of FIRE 2022 - Forum for Information Retrieval Evaluation, Kolkata, India, December 9-13, 2022, CEUR Workshop Proceedings, CEUR-WS.org, 2022.

[4] S. Satapara, B. Modha, S. Modha, P. Mehta, Fire 2022 ilsum track: Indian language summarization, in: Proceedings of the 14th Forum for Information Retrieval Evaluation, ACM, 2022.

[5] S. Ghodratnama, A. Beheshti, M. Zakershahrak, F. Sobhanmanesh, Extractive document summarization based on dynamic feature space mapping, IEEE Access 8 (2020) 139084–139095.

[6] N. K. Nagwani, Summarizing large text collection using topic modeling and clustering based on mapreduce framework, Journal of Big Data 2 (2015) 1–18.

[7] K. Shetty, J. S. Kallimani, Automatic extractive text summarization using k-means clustering, in: 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), 2017, pp. 1–9. doi:10.1109/ICEECCOT.2017.8284627.

[8] M. Jain, H. Rastogi, Automatic text summarization using soft-cosine similarity and centrality measures, in: T2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), IEEE, 2020, pp. 1021–1028.

[9] R. Khan, Y. Qian, S. Naeem, Extractive based text summarization using kmeans and tf-idf, International Journal of Information Engineering and Electronic Business 11 (2019) 33–44. doi:10.5815/ijieeb.2019.03.05.