MultiLingual(Hindi,Gujarati,Kannada) Text Summarisation

Manish Agrawal[1], **Nij Padariya**[2], **Yashwanth Holla**[3]
[1]231110028, [2]231110032, [3]231110060
[1]CSE, [2]CSE, [3]CSE
amanish23@iitk.ac.in,nijbp@iitk.ac.in, yashwanthh23@iitk.ac.in

1 Introduction

1. In the context of text summarization, the process involves condensing a given text while ensuring that the essential information and overall meaning are preserved. This task becomes particularly challenging in a multilingual setting, where text may be in different languages such as Hindi, Gujarati, and Kannada. In our project on Multilingual text summarization, we explored two distinct pipelines to address this complexity.

1.1 First Pipeline:Multilingual Summarization:

This pipeline is designed to handle data in multiple languages, including Hindi, Gujarati, and Kannada. The objective is to generate summaries in the corresponding language of the input data. For instance, if the input is in Hindi, the summary should also be in Hindi; likewise for Gujarati and Kannada inputs. This pipeline requires robust language processing capabilities to understand and summarize text in different languages accurately.

1.2 Second Pipeline: Hindi-only Summarization :

In contrast to the multilingual pipeline, this pipeline focuses specifically on processing data in Hindi and generating summaries in Hindi. Within this pipeline, we experimented with various approaches to text summarization. These approaches include rule-based methods, TF-IDF (Term Frequency-Inverse Document Frequency), mT5 (multilingual T5 model), and IndicBart.

2 Multilingual Summarization

1. The models used to implement this pipeline are IndicTrans and Pegasus.

2.1 IndicTrans Model:

The IndicTrans model is a machine translation model specifically designed for translating Indian languages into English and vice versa. It leverages transformer-based architectures and large-scale training data to achieve accurate and fluent translations between Indian languages and English.

2.2 Pegasus Model:

The Pegasus model is a state-of-the-art abstractive text summarization model developed by Google Research. It is based on the transformer architecture and is trained using a massive corpus of diverse text data. Pegasus excels in generating coherent and informative summaries by understanding the context and key information in the input text.

2.3 Implementation

2. In this pipeline, we implemented the multilingual text summarization task using a sequential approach involving machine translation and summarization techniques, followed by another round of machine translation.

3. Firstly, we utilized the IndicTrans model, which is specifically designed for translating Indian languages, to convert the input text from Indian languages (such as Hindi, Gujarati, or Kannada) into English. This step ensures that the text is in a language that can be processed by the subsequent summarization model.

   Next, we employed the Pegasus model for text summarization.We have used samsum dataset to finetune for English summarization.We have finetuned pegasus model on various parameters and select that model which have best score among all.

After generating the summary in English using Pegasus, we used the IndicTrans model again to translate the English summary back into the corresponding input language (e.g., Hindi, Gujarati, Kannada). This final translation step ensures that the generated summary is in the same language as the original input text, completing the multi-lingual summarization process. We also evaluated its results on the ground truth.

3 Hindi only Summarization

1. In our Hindi text summarization project, our primary goal is to find the best-performing model for summarization tasks on Hindi text. To achieve this, we have selected pretrained models specifically trained on the Hindi language, such as mT5 and IndicBERT, for direct summarization. These models are well-suited for understanding the nuances of Hindi text, which can lead to more accurate summaries.

2. Additionally, we are exploring other algorithms that may not be originally designed for Hindi but could still provide valuable insights. For example, we are considering using TF-IDF, a commonly used algorithm for summarization based on word frequency, and rule-based approaches. While these methods are more commonly associated with English text, we believe they are worth exploring to see how they perform on Hindi text. By running these algorithms on our system and analyzing the results, we aim to identify the most effective approach for Hindi text summarization.

3.1 Rule-based Summarization:

Approach: Rule-based summarization relies on predefined rules and heuristics to extract important sentences or phrases from the input text. In this approach, specific rules are defined to identify key sentences based on criteria such as sentence length, presence of keywords, importance of sentences in the text structure, etc.

3.2 TF-IDF (Term Frequency-Inverse Document Frequency):

Approach : TF-IDF is a statistical technique used to evaluate the importance of a word in a document relative to a corpus of documents. In TF-IDF, each word's importance is calculated based on its frequency in the document (Term Frequency) and its rarity across the corpus (Inverse Document Frequency). Words with higher TF-IDF scores are considered more significant in representing the content.

3.3 mT5 (Multilingual T5 Model):

mT5 is a variant of the T5 (Text-to-Text Transfer Transformer) model that supports multilingual text processing tasks.mT5 is pre-trained on a diverse range of languages and can generate text in multiple languages, making it suitable for multilingual summarization tasks. It uses a text-to-text format, where input text is converted into a specific text format representing the summarization task, and the model generates the corresponding summary.
Approach : We have used the tokeniser and the model from t5 checkpoint : csebuetnlp/mT5-multilingual-XLSum. After the tokenisation is done using the tokeniser, these tokens are passed to the model for text summarisation.

3.4 IndicBART

IndicBART, developed by AI4Bharat, is a variant of the BART (Bidirectional and Auto-Regressive Transformers) model specifically tailored for the Indian language context. It is pre-trained on a large corpus of multilingual data, including various Indian languages, making it suitable for tasks such as text summarization in Indian languages
Approach : We have used tokenisers and the model of IndicBART.Firstly we tokenised the input text so that the model can understand it. Once the input text is tokenized and encoded, pass it to the IndicBART model for summarization.

4 Datasets

1. The datasets that we have used for our tasks are :

2. For the pipeline 1 we have used samsum dataset for the finetuning of pegasus model for text summarisation task.

3. For the pipeline 2, the dataset available in the kaggle has only headline generation which was not suitable for our task so we went on

with creation of our dataset with the help of CNN DailyMail dataset.

4.1 Dataset-Creation :

Firstly we extracted the 3000 lines from CNN dataset and then we passed it into IndicTrans model, this model has converted the dataset from English language to Hindi language.

5 Results

All the results mentioned below are generated from the dataset of CNN translated to Hindi language with 3000 rows of data.

The Rouge1,Rouge2,RougeL and Cosine similarity scores for the different implementations are given below :

- **Rule Based Summary:**

  Rouge-1: Recall: 0.235651, Precision: 0.412866, F-measure: 0.291755
  Rouge-2: Recall: 0.076136, Precision: 0.160638, F-measure: 0.100882
  Rouge-L: Recall: 0.159748, Precision: 0.278538, F-measure: 0.197031

- Average similarity score: 0.7296695890987738

- **TF-IDF based summary:**

  Rouge-1: Recall: 0.406837, Precision: 0.226605, F-measure: 0.279576
  Rouge-2: Recall: 0.154813, Precision: 0.071947, F-measure: 0.094720
  Rouge-L: Recall: 0.281068, Precision: 0.157232, F-measure: 0.192962

- Average similarity score: 0.7313403066471219

- **IndicBart based summary:**

  Rouge-1: Recall: 0.457595, Precision: 0.239545, F-measure: 0.306385
  Rouge-2: Recall: 0.185096, Precision: 0.078773, F-measure: 0.108186
  Rouge-L: Recall: 0.332287, Precision: 0.17379, F-measure: 0.22209

- Average similarity score: 0.733336230430752

- **mT5 based summary:**

  Rouge-1: Recall: 0.191032, Precision: 0.399062, F-measure: 0.253969
  Rouge-2: Recall: 0.050734, Precision: 0.121811, F-measure: 0.070238
  Rouge-L: Recall: 0.141892, Precision: 0.296661, F-measure: 0.188658

- Average similarity score: 0.7261072570346296

- **MultiLingual Text summarisation:**

  Rouge-1: Recall: 0.234996, Precision: 0.401981, F-measure: 0.2869
  Rouge-2: Recall: 0.085709, Precision: 0.152956, F-measure: 0.105887
  Rouge-L: Recall: 0.215126, Precision: 0.369035, F-measure: 0.262987

- Average similarity score: 0.6197199414601785

- The graphical representation of Rouge1,Rouge2,RougeL scores for the different implementations are also attached in the Plots from Results section:

6 Finetuning of Pegasus Model

6.1 Before Finetuning

On the samsum dataset test part, we made a run of text summarization in English language before finetuning the Pegasus model. The rouge scores for this method is given as : rouge1 :0.015558, rouge2:0.000295, rougeL: 0.015537, rougeLsum:0.01554.

The similarity score for this model obtained before fine tuning is : 0.6370636393432221.

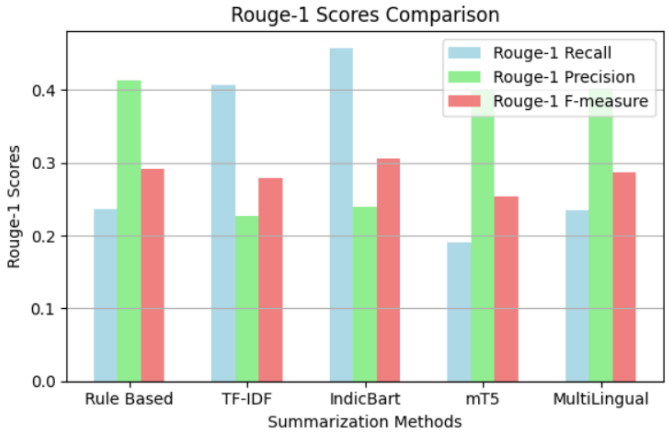
6.2 After Finetuning

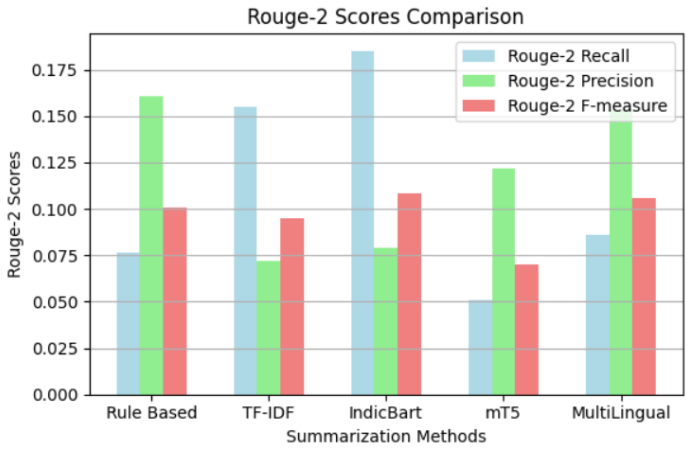
On the samsum dataset test part, we made a run of text summarization in English language after finetuning the Pegasus model. The rouge scores for this method is given as : rouge1 :0.018748, rouge2:0.000387, rougeL: 0.018638, rougeLsum:0.018662.

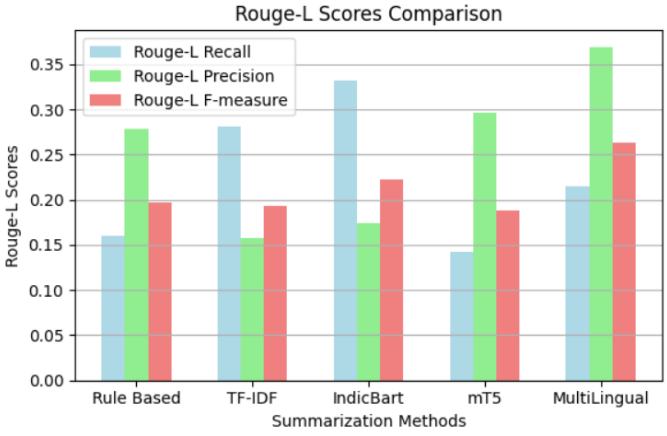
The similarity score for this model obtained before fine tuning is : 0.7356697076559067.

From the above scores we can observe that finetuning the model has improved the similarity scores.

7 Plots from Results







These are the plots obtained by comparing the rouge scores of different approaches that we followed for the text summarization are :

8 Conclusion

From the scores we can observe that IndicBART performed better in the text summarisation for the given dataset which is followed by TF-IDF,rule-based,mT5 but multilingual pipeline approach couldn't perform well.