Report

Abstract:
This project proposes an explainable approach to abstractive summarization for the Hindi language by incorporating extractive summarisation contexts into a fine-tuned sequence-to-sequence model. We explore the potential of fine-tuning the IndicBART and other models on our news articles and headlines dataset and investigate the use of cosine similarity between the article's title and individual sentences to inform the attention mechanism.

Introduction:
This project focuses on developing an abstractive summarisation model specifically for Hindi. We aim to explore the effectiveness of our proposed approach in generating high-quality summaries for Hindi texts, taking into account the challenges posed by code-mixing and script mixing.
To evaluate the performance of our model, we employ two widely-used metrics: ROUGE score and BERT score. The ROUGE score measures the quality of a summary by comparing it to a set of reference summaries.
We establish baseline scores using three state-of-the-art models: mT5, IndicBART, and IndicBARTSS. These models serve as a benchmark for evaluating the performance of our proposed approach.

Related Work:
Both extractive and abstractive summarisation are well-explored problems in the English language context. A lot of datasets are available in English. Pubmed, arXiv, CNN/Daily Mail are to name a few. Guo et al. extended the T5 model to take long text as input and performed summarisation over the PubMed dataset. PRIMERA is another model that uses the Longformer model and achieves state-of-the-art results on datasets like arXiv summarisation data, MultiNews, and WCEP datasets. Hasan et al. introduced a multilingual dataset named XL-Sum comprising 44 languages. They experimented with the mT5 model to perform abstractive summarisation and report results. In (Rush et al., 2015), the researchers used convolution models to encode the input and context-sensitive feed-forward network with the attentional mechanism. They showed better results for Gigaword and DUC datasets. (Chen, 2015) have produced a sizeable Chinese dataset for short text summarisation (LCSTS),
Aries et al.performed multilingual and multi-document summarisation by clustering sentences into topics using a fuzzy clustering algorithm. They score each sentence based on
the topic coverage and then create a summary using the highest-scoring sentences. Ladhak et al. proposed WikiLingua, an article-summary pair multilingual dataset available in 18 languages for cross-lingual abstractive summarisation. They fine-tuned mBART in their experiments.

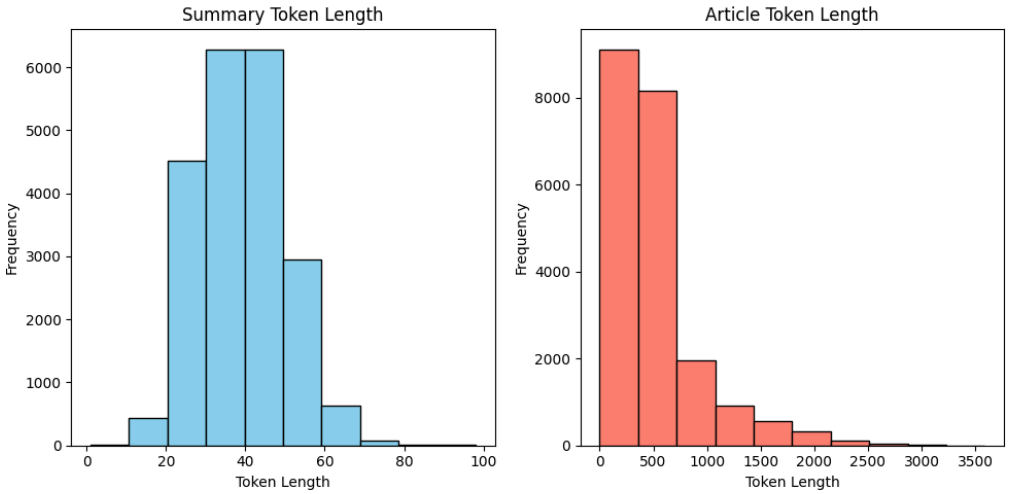
For our specific task, i.e., 2023 - ILSUM, we used a fine-tuned Indic-Bart-SS,mBART, mT5, and Indic-Bart, which was done earlier by Yadav et al.

Methodology:

We have incorporated extractive text summarisation to improve the results of the generated abstractive test summaries by first computing the cosine similarity between the **sentences in every article** with the **title of the article** (The sentences were split by the पूर्ण विराम ('।')). We further use these similarities to calculate a weighted sum of the final hidden representations of the encoders of the fine-tuned IndicBARTSS baseline model. The IndicBARTSS pre-trained model is available on Hugging Face. The weighted encoded representation is then passed to the decoder model along with the attention masks to auto-regressively generate the summary. Since we had low availability of training resources, we performed this "post-fine-tuning" on only 20% of the training dataset. We believe that by incorporating the similarity scores into the encoded representation, the model will "learn" to discern the relative portions of the input article text. Note that it is our assumption that the title of the text always contains contextual information about the article.

Dataset:

We use the Hindi dataset available for task 1 at **https://ilsum.github.io/ilsum/2023/dataset.html.** It has 21,225 samples in the training dataset and 3000 samples in the testing dataset. The length of the token on the training set is given below:



Note: The testing dataset does not have a "Summary"; thus, we split the training set to get the output.

Experimental Setup:

The same experimental setup was performed for fine-tuning all the baseline models, including the IndicBARTSS model, which was used in our Sentence Similarity Abstractive Summarization model.

Learning Rate: 0.00005, Epochs: 3, Optimiser: Adam, Weight Decay: 0.01
We have used HuggingFace's Trainer for performing the fine tuning.

Metrics:

1. ROGUE Score = ∑ (Recall of n-grams) where this score can be made into ROUGE-1 (unigram) measures the overlap of single words, ROUGE-2 (bigram) measures the overlap of two-word sequences, and so on. ROUGE-N is often used to evaluate the grammatical correctness and fluency of the generated text

2. ROUGE-L: ROUGE-L measures the longest common subsequence (LCS) between the candidate and reference texts. It computes the precision, recall, and F1 score based on the length of the LCS. ROUGE-L is often used to evaluate the semantic similarity and content coverage of generated text, as it considers the common subsequence regardless of word order.

3. BERTScore leverages the pre-trained contextual embeddings from BERT and matches words in candidate and reference sentences by cosine similarity. It has been shown to correlate with human judgment on sentence-level and system-level evaluation. Moreover, BERTScore computes precision, recall, and F1 measures, which can be helpful in evaluating different language generation tasks.

Results:

Baseline

Model Name	**BERT-SCORE**	**ROGUE-1**	**ROGUE-2**	**ROGUE-L**
indic-bart	P: 0.7994 R: 0.7972 F1: 0.7977	Fmeasure: 0.18350	Fmeasure 0.0860	Fmeasure 0.1824
indic-bartSS	P: 0.7507 R: 0.7549 F1: 0.7519	Fmeasure 0.1593	Fmeasure 0.0734	fmeasure 0.158
google/mt5-small	P: 0.6219 R: 0.5792 F1: 0.5994	fmeasure: 0.0048	fmeasure: 0.0007	fmeasure: 0.0056

Finetuned:

Model Name	BERT-SCORE	ROGUE-1	ROGUE-2	ROGUE-L
Custom indic-bart	P: 0.8263 R: 0.8376 F1: 0.8312	fmeasure=0.2722	Fmeasure 0.1279	fmeasure=0.272
Custom indic-bartSS	P: 0.7676 R: 0.801 F1: 0.707	fmeasure: 0.3073	fmeasure: 0.1896	fmeasure: 0.3033
Someman/Bart-Hindi	P: 0.6967 R: 0.6724 F1: 0.6840	fmeasure: 0.0825	fmeasure: 0.0654	fmeasure: 0.0996

Discussion

1. From the training data, we see that often, the beginning of the summary starts with English words; on analysis with the best of our model, we couldn't make it translate the starting words to English. As we use a multilingual tokeniser and embedding model, both the translation and the Hindi sentence are equal in the embedding space, giving out less loss, but the ROGUE score remains high.

2. The fine-tuned model for the summarisation task performed better. Still, the FineTuned Model on simply the heading of a text article also performed well with better NER recognition in general and more verbs in the text.

Conclusion and Future Work.

- Finetuning a multi-lingual model worked well with the similarity scores in our model. This can be improved by considering whether a sentence contains multiple Named Entities, which need a higher score than a simple sentence with fewer named entities.

- Translation: We can encourage our model to translate sentences provided it is a multi-lingual, Specifically Hindi and English tokeniser with a weight during loss calculation so that our model outputs an English translation.

- We didn't add other metrics like coverage, repetition_penalty etc which can be used to add variation and generate a better summary.

Models Link:
https://drive.google.com/drive/folders/1f-A2qmpIhsDYcArtpWXikw_loPHL083Q?usp=sharing

References:

1. Summarising Indian Languages using Multilingual Transformers-based Models from https://arxiv.org/pdf/2303.16657.pdf**.**

2. IndicBART: A Pre-trained Model for Indic Natural Language Generation from https://arxiv.org/pdf/2109.02903.pdf

3. Automatic Text Summarization Methods: A Comprehensive Review from https://arxiv.org/pdf/2204.01849

4. Hindi Text Summarization using Sequence to Sequence Neural Network from https://assets.researchsquare.com/files/rs-2036546/v1/be822c26-1573-41f5-9389-f92591338ae5.pdf?c=1664130567

5. Finetuning Summarization at https://huggingface.co/docs/transformers/en/tasks/summarization

6. Summarisation using T5 https://huggingface.co/learn/nlp-course/en/chapter7/5

7. BERT Score from https://huggingface.co/spaces/evaluate-metric/bertscore and ROGUE score from ROGUE LINK