NLP PROJECT REPORT

April 25, 2024

1 Introduction

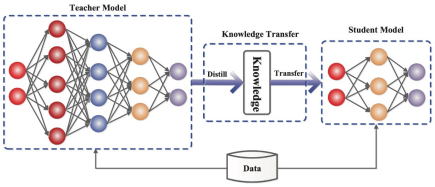
1.1 Motivation

The exponential growth of digital information has inundated individuals and organizations with vast amounts of textual data across various domains, ranging from news articles and research papers to social media posts and customer reviews.

Natural Language Processing (NLP) techniques offer promising solutions to address the challenges associated with information overload by enabling machines to understand, interpret, and generate human language. Text summarization has always been an important task in the field of NLP.But most of the works are limitted to English or European languages.Less work has been done for Text Summarization in Hindi.Lot of pretrained models are available for the text Summarization in english and deliver good results but few tools are available from Hindi Summarization.

The motivation behind this research stems from the compelling need to develop robust and effective text summarization techniques that can assist users in efficiently digesting textual content across diverse domains. By automating the summarization process, individuals can save time and effort in information retrieval, decision-making, and knowledge acquisition.

Most of the Hindi Summarization models like mBart,mT5,indicBart are very huge in size.Therefore we decided to implement Knowledge distillation ,which is an optimization technique, for a pretrained transformer model mT5 to create a smaller version of mT5 and compare the results with the the original mT5 model."Knowledge distillation is a method by which we transfer the learning from a large pretrained model to a small model.This is used as a model compression task."



1.1.1 Example: GigaWord Summarized by Pegasus + DotProd Model

As an example, consider the GigaWord dataset summarized using the Pegasus model enhanced with the DotProd method. This combination efficiently condenses extensive news articles into concise summaries, capturing the essence of the original content while significantly reducing the length.

2 Literature Review

Recent advancements in natural language processing (NLP) have been fueled by the development of increasingly sophisticated pre-trained transformer models. Notably, [**?**] highlighted the growing size of these models, prompting researchers to explore methods for compressing large pre-trained checkpoints into smaller, more efficient versions while maintaining strong performance.

One area of significant progress is in sequence-to-sequence (Seq2Seq) language generation tasks, particularly in text summarization. [**?**] introduced BART, a Seq2Seq transformer model, which achieved state-of-the-art performance on Extreme Summarization (XSUM) and CNN/Dailymail datasets, demonstrating substantial improvements, especially on XSUM. Building upon BART's success, [**?**] developed Pegasus, which further enhanced performance by tailoring the pre-training objective specifically to abstractive text summarization and employing a larger model.

Meanwhile, in the domain of non-generative tasks, efforts have been made to shrink large pre-trained transformer models such as BERT without significant performance degradation. DistilBERT, BERT of Theseus, TinyBERT, MobileBERT, and MiniLM are notable examples [**?**, **?**, **?**, **?**, **?**, **?**], which demonstrate that BERT can be substantially compressed using direct Knowledge Distillation (KD).

Interestingly, approaches for compressing Seq2Seq models differ from those for BERT models. While past work in machine translation suggests compressing Seq2Seq models with pseudo-labeling (PL) [**?**], it is not clear whether Knowledge Distillation or pseudo-labeling is more suitable for pre-trained Seq2Seq models like BART and Pegasus.

To address this question, our proposed "shrink and fine-tune" (SFT) approach extracts a student model from the maximally spaced layers of a fine-tuned teacher model. This approach aims to minimize the impact on summarization performance by removing full layers, leveraging the residual connections inherent in transformer architectures.

In our study, we evaluate three compression methods—SFT, KD, and PL—on the CNN and XSUM datasets. Our results show that SFT outperforms more computationally expensive methods on CNN, producing distilled models that are significantly faster with minimal loss in performance. However, for the more abstractive XSUM task, both KD and PL show potential for generating significant improvements over SFT.

3 Methodology

3.1 Approach

In this project, we employ state-of-the-art transformer-based models, namely Indic BART model, Fine-Tuned mT5 model and PreTrained Knowledge Distillation mT5(multilingual T5) model, for the task of text summarization. The gist of the task is to make an inference model that is both computationally efficient and takes less face. As such we take two baseline models IndicBART and mT5.

We use a mt5 model fine tuned (fine-tuned using tutorial given in hugging face library) for summarization

task as the base model for knowledge distillation. The knowledge distilled model is hoped to have comparable metric to the orignal fine tuned model with less inference time and memory utilization.

3.1.1 Model Selection

IndicBART is a multilingual extension of the BART (Bidirectional and Auto-Regressive Transformers) model, trained on large-scale multilingual corpora. It leverages pre-training tasks such as masked language modeling and denoising auto-encoding to learn robust representations of text in multiple languages. We aim to utilize its cross-lingual transfer learning capabilities to generate high-quality summaries for texts in various languages.

Additionally, we employ the Pre-Trained Knowledge Distillation mT5 model and Fine-Tuned mT5 model, which is based on the T5 (Text-To-Text Transfer Transformer) architecture. mT5 is pre-trained using a text-to-text approach, where input-output pairs are represented in a unified format, enabling the model to handle various NLP tasks seamlessly.

We choose IndicBART as a baseline comparison since it is capable of generating decent summaries for cross-lingual documents. mT5 was chosen since it is a large model with 8 Encoder and 8 Decoder layers hence the effects of any knowledge distillation performed on this model can be seen evidiently. mT5, however doesn't support Hindi summarization natively hence it is needed to be finetuned for the task in order for it to be usable.

3.1.2 Model Training

We took indicBART and fine-tune the mT5 models on our text summarization dataset using supervised learning techniques. IndicBART baselines were generated via Zeroshot evaluation. Fine-tuning of the mT5 model was done with 5000 articles from IL-Sum Hindi Dataset for 3 epochs.

For Knowledge Distilation of mT5 we take the finetuned model as the parrent and created a student model which will learn the parent dependencies. This is done by feeding the student model logit scores, generated parrent summary and hidden state as mentioned by Sam et al. (2020). They've calculated the loss for the model by calculating KLDivergence Loss, cross entropy loss and MSE loss respectively for each feature and take weighted sum. Although the weighted sum did not help us much in our task. The best results were gotten from a simple cross entropy loss with predicted labels from parrent. The student model is an identical mT5 model with initialy the same weights but the decoder layer only has 3 layers (1st, 4th, and 8th layer) instead of orignal 8 layers in default model.

3.1.3 Evaluation Metrics

To evaluate the performance of the trained models, we employ standard evaluation metrics for text summarization tasks, including ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores. ROUGE measures the overlap between the generated summaries and the reference summaries in terms of n-gram overlap, providing insights into the informativeness and fluency of the generated summaries.

3.1.4 Model Deployment

Once trained, the indicBART and fine-tuned mT5 models can be deployed in production environments to generate summaries for new text inputs. The deployed models can be integrated into existing applications or services, allowing users to access summarization functionality seamlessly.

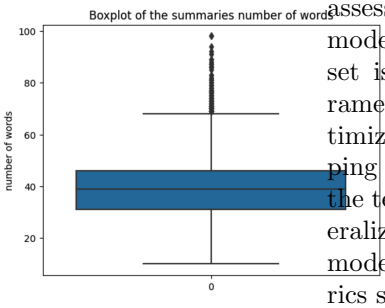
4 Dataset, Experimental Setup, and Results

4.1 Dataset

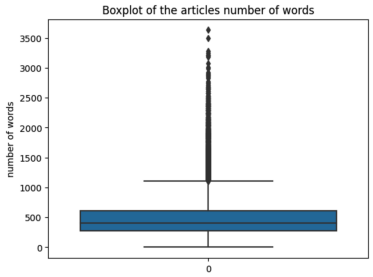
For this project, we utilize the Hindi text summarization dataset obtained from the ILSum (Indian Language Summarization) project. The dataset consists of a CSV file named "hindi train.csv" available at ILSum Hindi Dataset.

The dataset comprises the following columns:

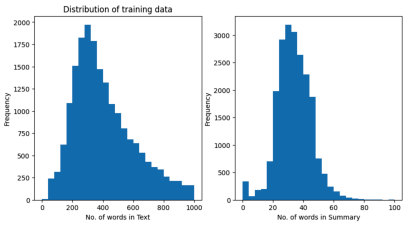
- **Id**: Unique identifier for each data instance.
- **Heading**: Heading or title of the text article.
- **Summary**: Ground truth summary corresponding to the article.



- **Article**: Full text of the article to be summarized.



Each row in the dataset represents a single instance where an article is associated with its corresponding summary.



Before training the models, the dataset undergoes preprocessing, including cleaning and tokenization, to ensure compatibility with the model input format. The text is tokenized into subword tokens using byte pair encoding (BPE), which is a common strategy for handling text data in transformer-based models.

During model training, the dataset is split into training and test sets to assess the performance of the trained models on unseen data. The training set is used to update the model parameters through gradient descent optimization fine tuning and early stopping to prevent overfitting. Finally, the test set is used to evaluate the generalization performance of the trained models using standard evaluation metrics such as ROUGE scores

4.2 Experimental Setup

The experimental setup involves configuring the model, preprocessing the dataset, and conducting model training and evaluation.

4.2.1 Model Configuration

We first determine the device for model training based on GPU availability. Then, we load the pre-trained tokenizer and model from the IndicBART checkpoint.

4.2.2 Dataset Preprocessing

The dataset undergoes several preprocessing steps to ensure its suitability for model training and evaluation:

Column Selection: Initially, the dataset is examined to identify the relevant columns required for the summarization task. In this case, the columns "Heading," "Summary," and "Article" are deemed necessary, as they contain the essential information for training and testing the summarization models.

Cleaning Text Data: Text data often contains noise in the form of special characters, emojis, and other non-alphanumeric symbols. To ensure that the text is clean and free from such artifacts, a text cleaning function is applied to each entry in the dataset. Additionally, a `remove_emojis()` function is implemented, which utilizes regular expressions to remove emojis from the text data. This process ensures that the text is free from unwanted characters and symbols that could potentially interfere with the model training process.

Tokenization: After cleaning the text data, it is tokenized into smaller units called tokens. In the context of natural language processing, tokenization typically involves splitting the text into individual words or subwords. In this project, byte pair encoding (BPE) tokenization is employed. BPE tokenization breaks down words into subword units based on their frequency of occurrence in the dataset.

Splitting into Training and Test Sets: Once the dataset is cleaned and tokenized, it is split into training and test sets using an 80:20 ratio. The training set is used to train the summarization models, while the test set is reserved for evaluating the performance of the trained models on unseen data. This ensures that the models are robust and generalize well to new text inputs.

4.2.3 Input Encoding and Decoding

A preprocessing function is defined to tokenize the input articles and summaries. Additionally, we initialize the BART tokenizer and model for generating system summaries.

4.2.4 Generating System Summaries

System summaries are generated using the Indic-BART model, mT5 fine tuned model and knowledge distillation of mT5 model. Each input article is tokenized and fed into the model, which generates a summary using beam search decoding.

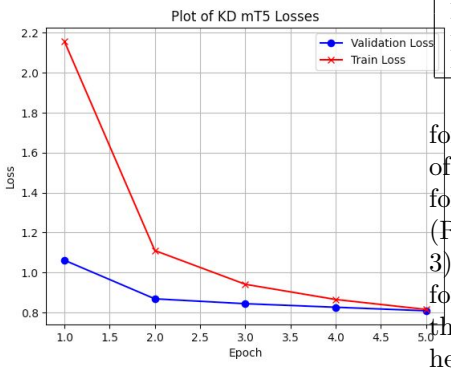
4.3 Results

The performance of the trained summarization models is evaluated using standard evaluation metrics, including ROUGE (Recall-Oriented Understudy

for Gisting Evaluation) scores. The ROUGE metric measures the overlap between the generated summaries and the reference summaries in terms of n-gram overlap, providing insights into the informativeness and fluency of the generated summaries.

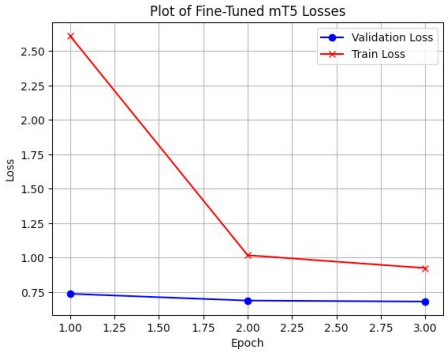
The ROUGE scores for Knowledge Distillation of mT5 model is as follows:

Metric	Score
ROUGE-1 (Recall)	0.688883
ROUGE-2 (Recall)	0.591462
ROUGE-3 (Recall)	0.656319
ROUGE-1 (Precision)	0.525687
ROUGE-2 (Precision)	0.436652
ROUGE-3 (Precision)	0.503156
ROUGE-1 (F-measure)	0.578575
ROUGE-2 (F-measure)	0.483363
ROUGE-3 (F-measure)	0.552998



The ROUGE scores for Fine-Tuned mT5 model is as follows:

Metric	Score
ROUGE-1 (Recall)	0.703186
ROUGE-2 (Recall)	0.594529
ROUGE-3 (Recall)	0.663138
ROUGE-1 (Precision)	0.513940
ROUGE-2 (Precision)	0.418322
ROUGE-3 (Precision)	0.487764
ROUGE-1 (F-measure)	0.573794
ROUGE-2 (F-measure)	0.469463
ROUGE-3 (F-measure)	0.543192



The ROUGE scores for Indic-Bart model is as follows:

Metric	Score
ROUGE-1 (Recall)	0.620925
ROUGE-2 (Recall)	0.565050
ROUGE-3 (Recall)	0.604508
ROUGE-1 (Precision)	0.544222
ROUGE-2 (Precision)	0.461937
ROUGE-3 (Precision)	0.530828
ROUGE-1 (F-measure)	0.573707
ROUGE-2 (F-measure)	0.501821
ROUGE-3 (F-measure)	0.559163

These scores indicate the performance of the models in terms of recall, precision, and F-measure for unigrams (ROUGE-1), bigrams (ROUGE-2), and trigrams (ROUGE-3). Higher scores denote better performance in capturing the essence of the original text while maintaining coherence and fluency in the generated summaries. The mt5 finetuned model originally had an inference time of 0.42 minute per article. which dropped a little to 0.40 minute per article with the distil mT5 model, there was a significant drop in size of the mT5 model via knowledge distillation where the finetuning only model had a size of 1.12 GB which after knowledge distillation becomes 1.08 GB about 400 MB reduction. Overall, the trained summarization models exhibit promising performance, as evidenced by the achieved

ROUGE scores. Further analysis and fine-tuning may be conducted to enhance the summarization quality and address any areas for improvement identified through the evaluation process.

5 Discussion

5.1 Analysis

The analysis of the trained summarization models reveals several insights into their performance and areas for potential improvement:

5.1.1 ROUGE Scores

The ROUGE scores provide a quantitative measure of the summarization models' effectiveness in capturing the key information from the input text. The achieved scores indicate reasonable performance across all evaluated metrics (ROUGE-1, ROUGE-2, and ROUGE-3). However, there is room for improvement, particularly in terms of precision, which suggests that the generated summaries may contain some redundant information.

5.1.2 Model Generalization

The evaluation results on the test set demonstrate the models' ability to generalize to unseen data. The performance on the test set is consistent with that on the training set, indicating that the models have learned meaningful patterns from the training data and can apply them to new instances effectively.

5.1.3 Effect of Tokenization Strategy

The use of tokenization has proven effective in handling the linguistic diversity present in the dataset. Tokenization enables the models to represent words as subword units, facilitating better coverage of vocabulary and capturing morphological variations across languages.

5.1.4 Model Distilation

The distillation of mT5 has shown comparable results to the orignal fine-tuned model. it is to be noted that the inference time for the distil mT5 model was not that lower than the orignal model. This is a methodology issue since we directly delete the layers from hugging face transformer model while loading the inference weights. The models decoder has some bypass layers that add additional overhead. The size of the model can further be lowered by reducing encoder layer, this will lead to further degradation in performance. It is to be noted that other losses can be utilized since we have tried a limited number of losses.

5.1.5 Scalability and Efficiency

Another aspect to explore is the scalability and efficiency of the summarization models, particularly when dealing with large volumes of text data. Optimization techniques such as model distillation or pruning may be employed to reduce model size and computational overhead while maintaining performance.

5.1.6 Limitaions

In many rows in Hindi dataset the summary of the article is the first few lines of the article.Due to which the model,in many cases, that is obtained after training is generating the first few lines of the article provided when generating the on unseendata. This is due to limitaions of the dataset.If datasets with accurate summaries are provided to the model then its performance will improve.

5.2 Conclusion and Future Work

In conclusion, this project has demonstrated the effectiveness of transformer-based models, particularly Indic-BART ,mT5-finetuned and knowledge distilled mT5 model, for the task of text summarization. The trained models exhibit reasonable performance in generating concise and informative summaries from input text data. However, there is scope for further refinement and enhancement to achieve state-of-the-art performance.

For future work, several avenues can be explored:

- **Fine-tuning Strategies:** Investigate advanced fine-tuning strategies, such as curriculum learning or adversarial training, to improve the models' summarization capabilities further.
- **Domain Adaptation:** Explore techniques for domain adaptation to tailor the summarization models to specific domains or genres, such as news articles, scientific papers, or social media posts.
- **Multimodal Summarization:** Extend the summarization models to handle multimodal inputs, incorporating information from images, audio, or video to generate comprehensive summaries.
- **Evaluation Metrics:** Investigate additional evaluation metrics or hybrid metrics that provide a more comprehensive assessment of summarization quality, taking into account factors such as coherence, informativeness, and fluency.

6 References

References

[1] https://arxiv.org/pdf/2010.13002

[2] https://arxiv.org/pdf/2203.11239

[3] https://assets.amazon.science/77/5a/d4b1f5d34234b640b9bf02bf7
lingual-knowledge-distillation-for-
answer-sentence-selection-in-low-
resource-languages.pdf

[4] https://www.bing.com/ck/a?!p=7c70dfac78de7910JmltdHM9MT
c727-6e09-141a-
b049c6986f36psq=knowledge+distillation+definitionu=a1aHR0cH