Implementing Deep Learning-Based Approaches for Article Summarization in Indian Languages

Rahul Tangsali[1,3,*,†], Aabha Pingle[1,3,†], Aditya Vyawahare[1,3,†], Isha Joshi[1,3,†] and Raviraj Joshi[2,3]

[1]*SCTR's Pune Institute of Computer Technology, Pune*
[2]*Indian Institute of Technology Madras, Chennai*
[3]*L3Cube, Pune*

Abstract
This paper presents a summary of the work by our team Next Gen NLP, submitted at ILSUM 2022- a shared task in collocation with FIRE 2022, focused on the Indian language summarization of news articles. The shared task was to carry out either extractive or abstractive summarization of news articles written in Indian English, Hindi, and Gujarati. We achieved rank 2 in English, rank 4 in Hindi, and rank 3 in Gujarati in the validation phase of the competition. Whereas for the test phase of the competition, we achieved rank 3 in English and rank 4 in both Hindi and Gujarati. We experimented with pre-trained models in our work and fine-tuned those models with the ILSUM 2022 datasets. In our case, the fine-tuned SoTA PEGASUS model worked the best for English, the fine-tuned IndicBART model with augmented data for Hindi, and again fine-tuned PEGASUS model along with a translation mapping-based approach for Gujarati. Our scores on the obtained inferences were evaluated using ROUGE-1, ROUGE-2, and ROUGE-4 as the evaluation metrics.

Keywords
Extractive text summarization, Indian Languages, NLP, Pretrained models

1. Introduction

Text summarization is a trending research domain that has gained popularity with a plethora of emerging use cases seeking its application [1, 2]. The last few decades have witnessed tremendous growth in NLP research, especially text summarization. Text summarization has applications in a wide range of domains, including medicine, politics, news, etc. With the massive influx of news data in the form of newspaper articles, digital media, social media platforms, and so on, a need exists to automate the news summarization process so that useful insights could be achieved much faster than human workers were employed for the same task. Effective summarization approaches investigated recently have hastened the process and made their mark in the NLP research community by achieving state-of-the-art (SoTA) accuracies.

Forum for Information Retrieval Evaluation, December 9-13, 2022, India
*Corresponding author.
†These authors contributed equally.
rahuul2001@gmail.com (R. Tangsali); aabhapingle@gmail.com (A. Pingle); aditya.vyawahare07@gmail.com (A. Vyawahare); joshiishaa@gmail.com (I. Joshi); ravirajoshi@gmail.com (R. Joshi)

Three distinct types of text summarization techniques exist: extractive, abstractive, and hybrid. In extractive text summarization, key sentences and phrases are picked from the original document and are integrated to generate a final summary [3]. This summarization technique is easier to perform, but it may overlook the text's overall context or omit some essential information. This type of summary text is helpful for taking notes. Abstractive summarization analyses the full text and generates a summary based on the fundamental concepts of the text [4]. This summary is made using an entirely different wording style than the original text. Unlike the extractive summarization methodology, sentences from the original text aren't picked up directly. Abstractive summarization provides an intelligently curated summarization using unique phrases which are not native to the input text. However, with deep learning methodologies, preparing abstractive summaries could be difficult and take a long time with human judgment. The hybrid-based text summarization approach utilizes both extractive and abstractive text summarization methods to generate the final summary [5, 6].

With the emergence of NLP research worldwide, research on text summarization has been conducted in high-resource languages such as English and texts written in Indian subcontinent-based languages. Hindi and Gujarati are two of the most spoken Indian languages. Hindi is the most spoken language in India and is considered the official language in 9 states and 3 union territories and an additional official language in 3 other states across the country. Hindi is also one of the 22 scheduled languages of the Republic of India. Hindi is spoken by approximately 615 million people worldwide and was recorded as the third most spoken language in the world as of 2019. Gujarati is an Indo-Aryan language spoken predominantly by the Gujarati people in the Indian state of Gujarat. It is the sixth most spoken language in India and is spoken by around 55 million people worldwide. Hindi and Gujarati are spoken by a considerable percentage of the population across the world. Yet, there has been a backfoot witnessed in NLP research in these languages compared to high-resource languages spoken worldwide.

Text summarization research stretches back to 1958 when the first paper on the subject was published [7]. Since then, various methodologies have been presented for both abstractive and extractive text summarization in English. These include statistical-based, clustering-based, graph-based, semantic-based, machine learning, and deep learning-based approaches. Deep learning-based approaches, which focus on training neural nets, include work done by Mohsen et al. [8], Xu [9], Alami et al. [10], and Anand and Wagh [11]. In addition, encoder-decoder models have been proposed, with attention mechanisms incorporated in several proposed methodologies.

In comparison to English, lesser research has been done on text summarization research in Hindi and Gujarati. There is a significant shortage in dataset resources, preprocessing methodologies, and other research for many Indian languages, especially Gujarati, compared to English. This motivated us to develop system pipelines that could perform efficient extractive summarization for articles written in Hindi and Gujarati and achieve decent accuracy for the generated summaries. Many organizations are leveraging their services to Indian language speakers, and we aim to solve a small part of this challenge by performing summarization research in two of the widely spoken languages in India.

For this shared task [12], we implement pre-trained models [13] and tweak the conventional pipelines along with fine-tuning with new data to obtain better results than previously implemented systems. For English, we implement the PEGASUS [14], BRIO [15], and T5 [16] models and also leverage the SentenceBERT model for extractive summarization purposes [17, 18]. For Hindi, we implemented fine-tuning of IndicBART [19] with a right-shift operation (augmenting the original dataset by shifting the last sentence of the article to the top), XL-Sum [20], and mBART [21] models. For Gujarati, we implemented extractive summarization by translating each sentence in the Gujarati article to English, and by creating a corresponding mapping between the Gujarati and translated English sentences, and applying fine-tuned PEGASUS model for English to the resultant English article to generate the English summary. The generated extractive summary in English is then translated back to Gujarati by a back-mapping mechanism to get the final Gujarati summary. We also fine-tuned XL-Sum and mBART models for Gujarati article summarization.

2. Related Work

Text summarization research dates back to 1958 when the first article on the topic [7] was published. Since then, numerous rule-based and deep learning-based techniques have been presented. Rule-based approaches include work done by Baxendale [22], which selects sentences for a summary based on word position and heading of the article, and that by Oliviera in 2016 [23], which used scoring criteria such as lexical similarity, sentence centrality, text rank, and so on for text summarization.

Research on deep-learning approaches for text summarization picked up the pace when encoder-decoder [24] and attention-based architectures [25] were proposed. Yu [26] suggested methods for creating one-sentence summaries of news stories that use recurrent neural network models like LSTM [27] and GRU [28], as well as with/without attention. In recent years, fine-tuning pre-trained models using domain-specific datasets has been the dominant paradigm in text summarization research. Pre-trained models which implement the BART [29], T5 [16], etc. architectures have been proposed, which are available in the Hugging Face library. Recent research includes the implementation of an importance-based ordering approach implemented by Zhao et al., a cascade approach to abstractive summarization with content selection and fusion proposed by Lebanoff et al [30]., and usage of prompt-based models such as GPT-3 [31], PaLM [32], T0 [33], etc. Many times, articles considered for summarization can be multi-document in nature. Wang et al. [34] suggested a task-specific architecture for multi-document summarization by combining numerous texts into a single graph. Zhong et al. [35] implemented a semantic-based framework for the same.

In the case of Hindi and Gujarati, there has been relatively little research on text summarization. K. Vimal Kumar et al. [36] suggested a graph-based method for summarising text in Hindi. Gulati et al. [37] developed a unique fuzzy inference method for summarising multi-source Hindi literature. Gupta et al. [38] suggested a rule-based method for Hindi that included dead phrase and deadwood reduction strategies. Jain et al. [39] presented a real coded genetic algorithm for Hindi text summarization. For Gujarati, Shah and Patel suggested Gujarati Text Summarizer, which uses Textblob[1] and Gensim[2] to construct summaries from Gujarati text. Patel examines the preprocessing phase for text summarization of Gujarati texts, emphasizing

[1]https://textblob.readthedocs.io/en/dev/
[2]https://radimrehurek.com/gensim/

Table 1
Details of provided dataset for ILSUM Shared Task 2022

	English	Hindi	Gujarati
Train	12565	7958	8731
Validation	898	569	606
Test	4487	2842	3020

related issues and appropriate solutions [40].

3. Dataset Description

The datasets used in our research were provided to us by the ILSUM Shared Task organizers. Datasets were organized in a CSV format, with multiple columns describing each record in the file. These datasets were built using articles and headline pairs from several leading newspapers across India. The columns in the CSV files were- "id": denoting the ID for the article for unique identification, "Link": hyperlink from where the article has been extracted, "Heading": heading/title of the article, "Article": the actual content of the article, and "Summary": gold extractive summary of the article. Each article consisted, on average, of about 9 to 10 sentences, and the extractive summaries, on average, were a single sentence long. For the validation and test CSV files, there were only two columns: "id" and "Article", where it was expected to find the summary of the text present in the "Article" column. The dataset content was raw, with unnecessary punctuations and delimiters hindering the proposed pipeline, hence causing a need for efficient data cleaning.

Table 1 proposes the contents of the training, validation and test datasets in terms of number of records present in each set.

4. Data Preparation

The dataset provided by the organizers was pretty raw, with redundant punctuations and delimiters in the content. Hence, it was necessary to remove those so that the clean data obtained could be further tokenized and passed to the model. In addition, we remove stopwords present in the text [41] to avoid model redundancy towards not-so-useful data and convert the text to lowercase to generalize the model perception towards the text. Out of the five columns present in the CSV file, the "id", "Link" and "Heading" columns were seemingly redundant to be taken into consideration, so we filtered out those columns for the model to get trained only on the articles and their corresponding extractive summaries.

We use the SentencePiece[3] tokenizer for tokenizing the English, Hindi, and Gujarati article texts. SentencePiece is an unsupervised text tokenizer and detokenizer intended specifically for Neural Network-based text generation systems with a preset vocabulary size before neural model training. It extends direct training from raw sentences to incorporate subword units

[3]https://github.com/google/sentencepiece

(e.g., byte-pair-encoding (BPE) [42]) and unigram language model. This tokenizer can be defined implicitly using Hugging Face API[4] for model fine-tuning so that both tokenization and detokenization processes can be carried out without explicit code. First, a vocabulary of all the common words in all articles is created and further utilized to quantify the text to a vectorized format. Additionally, we apply padding to the maximum sequence length in the batch so that sequences of uniform length only would be passed ahead to the model [43].

For the translation+mapping-based approach that we implement as one of the approaches for Gujarati, we first split the sentences using the full stop as a delimiter. Then, each sentence is translated to English using the Google Translate API[5], and then the mapping is created between the original Gujarati sentence and the translated English sentence obtained. Finally, these English sentences from each paragraph are concatenated to get the final translated summaries in English.

5. Systems implemented

5.1. For English

5.1.1. Fine-tuning PEGASUS

PEGASUS stands for "Pre-training with Extracted Gap sentences for Abstractive Summarization", the paper for which was presented at the 2020 International Conference on Machine Learning by Zhang et al.[44]. By masking entire sentences from the text and then appending the gap sentences, the PEGASUS model yields a pseudo-summary of the input text. The PEGASUS model picks sentences that are essential to the model and removes or masks them from the input document. The model is then assigned with recovering those vital phrases, which it accomplishes by constructing the output sequence, including the critical documents entirely from the document's non-essential parts. The advantage of this technique is its self-supervision; the model may generate as many instances as there are documents without the need for human annotation, which is sometimes a bottleneck in fully supervised systems.

We fine-tuned the "pegasus-large" model[6] available on Hugging Face with the training dataset for English. This model is pre-trained on 350 million web pages and 1.5 billion news articles, making its accuracy state-of-the-art in text summarization research. The Hugging Face transformer library was used for fine-tuning purposes, which made the implementation easier. Since the training data was large enough, we decided to fine-tune the model for 1 epoch on the training data, along with a weight decay of 0.01, which took about 3.5 hours for the same. The inferences yielded a significant increase in ROUGE scores as observed to those obtained with the only pre-trained version of the model.

To further increase the ROUGE scores, we tried experimenting with the max-tokens parameter of the model during inference generation, which is the maximum length the generated inference can have. The organizers had specified the standard value of the same to 75. We experimented with a range of max-tokens values around that range, and we got max-tokens=65 to be the ideal

[4]https://huggingface.co/
[5]https://cloud.google.com/translate/
[6]https://huggingface.co/google/pegasus-large

value for the highest ROUGE scores.

We also experimented with augmenting the dataset by adding noise to each record of the dataset so that the model could predict the result better despite the noisy text present. The ROUGE score increase for the same needed to be increased, compared to the highest score we received.

5.1.2. Fine-tuning BRIO

BRIO stands for "Bringing Order to Abstractive Summarization", the paper presented in 2022 by Liu et al.[15]. Maximum Likelihood Estimation (MLE) [45] is often used to train summarization models. MLE presupposes that an ideal model would allocate full probability mass to the reference summary, which may result in poor performance when a model must compare numerous candidates that vary from the reference. Instead of relying on MLE training, BRIO has a contrastive learning component, enabling abstractive models to more precisely assess the likelihood of system-generated summaries.

We fine-tune the "Yale-LILY/brio-cnndm-uncased" version[7] of the BRIO model available on Hugging Face on the English dataset. Since BRIO is an extension to the BART model, we apply BART-based tokenization to the input text, which uses SentencePiece internally. We fine-tuned the English dataset on the model for 1 epoch with a weight decay of 0.01 and even experimented further with adding noisy text to each training record. The model's performance, however, was not as good as the fine-tuned PEGASUS model mentioned earlier.

5.1.3. Leveraging SentenceBERT for extractive summarization

The approach was a tweaked implementation derived from the paper "Fine-tune BERT for Extractive Summarization" presented by Liu in 2019 [46]. Here, extractive summarization is approached as a classification problem by predicting a score between 0 and 1 for each phrase in a text, i.e., by determining whether or not it belongs to the summary. The algorithm then creates a summary based on these scores by picking the phrases with the highest scores determined by certain relevant parameters.

We extract sentences using the SpaCy[8] library for each article in the training dataset. For every sentence in each training example, we assign a label of 1 if it belongs to the final extractive summary, else 0. The original dataset was unbalanced, as most of the sentences are unlikely to be in the summary. We augmented our dataset with new examples that balanced positive and negative examples. This annotated data, along with the labels, constitutes the input to our BERT model.

We fine-tuned the "sentence-transformers/all-mpnet-base-v2" model[9], since it proved to be the fastest among all models available in the sentence-transformers library. We set the batch size for training to 4, and the maximum sequence length of the generated summary to 512. The learning rate for training was set to 0.00001. We fine-tuned the SentenceBERT model for 3 epochs, which took approximately 4.5 hours. The original pre-trained BERT model is modified

[7]https://huggingface.co/Yale-LILY/brio-cnndm-uncased
[8]https://spacy.io/
[9]https://huggingface.co/sentence-transformers/all-mpnet-base-v2

by drop out and a dense layer on top of the BERT model to get the final output label. Finally, we get the inferences from the model by taking **two** sentences with the highest scores obtained from the BERT model, which gave an average summary length of around 70. We add only those sentences in the summary whose length is more than 25 characters.

5.1.4. Fine-tuning T5

The Text-to-Text-Transfer-Transformer (T5) paradigm suggests recasting all NLP tasks as a single text-to-text format with text strings as input and output. The original text of the input and output pairs during T5 pre-training is modified by introducing noise.

We fine-tuned the 'mrm8488/t5-base-finetuned-summarize-news' version[10] of the T5 model, which is pre-trained on 4515 English news articles. We fine-tuned this model on our dataset. We applied T5 tokenization to our dataset and fine-tuned the model for 20 epochs. The maximum length of the summary during the inference phase was set to 75.

5.2. For Hindi

5.2.1. Fine-tuning IndicBART

We used IndicBART[19], a multilingual, sequence-to-sequence pre-trained model. The model focuses on Indic languages majorly, and English as well. IndicBART is based on the mBART architecture and provides support for 11 Indian languages, and can be used to build various natural language generation applications for tasks like machine translation and summarization.

We fine-tuned the 'ai4bharat/IndicBART' version[11] of IndicBART available on Hugging Face on the training dataset for Hindi. The data used for training the model was augmented by adding noise to each record of the dataset. The model gave better results better after training on such a dataset. The model was fine-tuned for 2 epochs. We also experimented with the maximum length parameter while generating the inferences. Inferences obtained with 'max length' set to 60 gave the best ROUGE scores.

5.2.2. Fine-tuning XL-Sum

The paper titled 'XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages' [20] presents a multilingual dataset as well as an mT5 checkpoint fine-tuned on the dataset and proposes fine-tuned mT5 [47] with XL-Sum and experimented on multilingual and low resource summarization tasks. The model was fine-tuned on the 45 languages of the XL-Sum dataset.

We used the 'csebuetnlp/mT5_multilingual_XLSum' checkpoint[12] available on Hugging Face for our summarization task. To get the best results, we fine-tuned this checkpoint on the given Hindi training dataset for 2 epochs. This method gave ROUGE scores comparable to the Indic BART scores.

[10]https://huggingface.co/mrm8488/t5-base-finetuned-summarize-news
[11]https://huggingface.co/ai4bharat/IndicBART
[12]https://huggingface.co/csebuetnlp/mT5$_multilingual_XLSum$

5.2.3. Fine-tuning mBART

Pre-trained on multilingual corpora containing 25 languages, mBART (Multilingual Denoising Pre-training for Neural Machine Translation)[21] can be used for a wide range of tasks, including machine translation and summarization. We used the "facebook/mbart-large-cc25"[13], "GiordanoB/mbart-large-50-finetuned-summarization-V2"[14] and "ARTeLab/mbart-summarization-mlsum"[15] pre-trained models on the dataset.

The results obtained differed minutely. However, the "facebook/mbart-large-cc25" model gave us the best ROUGE scores; hence, we fine-tuned the model on the dataset for 1 epoch.

5.3. For Gujarati

5.3.1. Translation+Mapping+PEGASUS

We implemented the PEGASUS model for Gujarati by fine-tuning the "pegasus-large" model available on Hugging Face. As this model wasn't initially trained for the Gujarati language, we implemented translation and mapping steps to use this model for generating inferences on our Gujarati dataset.

First, we translated the Gujarati validation dataset to English and simultaneously stored the mapping between the English-translated sentence and the Gujarati sentence for each article in a dictionary. For translation, we used the GoogleTranslator module provided by deep-translator[16] library. Then, we generated the inferences on the English-translated validation dataset using the PEGASUS model fine-tuned for English, the max-tokens parameter for which was set to 75 initially. Finally, the generated inferences were back-mapped to give the original Gujarati sentences. As the dataset provided was extractive, we performed the mapping and back-mapping steps mainly to keep the summaries extractive in nature. It should be noted that the translation process was only used once, and the original Gujarati text was retrieved using the mapping developed during the Gujarati to English translation process.

To further increase the ROUGE scores, we experimented with the max-tokens parameter of the model. We observed that the English-translated sentences were longer than the original Gujarati sentences. Therefore, we tested by increasing the max-tokens parameter, and we inferred that max-tokens set to **85** provided the highest ROUGE scores.

5.3.2. Fine-tuning mBART

For this approach, we used the "facebook/mbart-large-cc25"[17] model. After applying the mBART tokenizer on the given Gujarati dataset, we fine-tuned the model for one epoch. This methodology gave us competent ROUGE scores. However, we improved our results by augmenting the dataset by adding noise to each record of the dataset to create a new record so that the model could predict better.

[13]https://huggingface.co/facebook/mbart-large-cc25
[14]https://huggingface.co/GiordanoB/mbart-large-50-finetuned-summarization-V2
[15]https://huggingface.co/ARTeLab/mbart-summarization-mlsum
[16]https://github.com/nidhaloff/deep-translator
[17]https://huggingface.co/facebook/mbart-large-cc25

Table 2
Results obtained in validation phase (English)

Approach Implemented	ROUGE-1	ROUGE-2	ROUGE-4
Fine-tuned PEGASUS	**0.5618**	**0.4509**	**0.4218**
Fine-tuned BRIO	0.4878	0.3723	0.3383
SentenceBERT leveraged for summarization	0.4639	0.3421	0.3156
Fine-tuned T5	0.4851	0.3588	0.3226

Table 3
Results obtained in validation phase (Hindi)

Approach Implemented	ROUGE-1	ROUGE-2	ROUGE-4
Fine-tuned IndicBART	**0.5536**	**0.4572**	**0.4162**
Fine-tuned XL-Sum	0.5281	0.4098	0.337
Fine-tuned mBART	0.5269	0.4271	0.3806

Table 4
Results obtained in validation phase (Gujarati)

Approach Implemented	ROUGE-1	ROUGE-2	ROUGE-4
Translation+Mapping+PEGASUS	**0.2028**	**0.1155**	**0.0835**
Fine-tuned mBART	0.1924	0.1095	0.0723
Fine-tuned XL-Sum	0.1718	0.0718	0.0361

Table 5
Test phase results for submitted models

Language	Model submitted for test phase	ROUGE-1	ROUGE-2	ROUGE-4
English	Fine-tuned PEGASUS	0.5568	0.4430	0.4123
Hindi	Fine-tuned IndicBART	0.5559	0.4547	0.4136
Gujarati	Translation+Mapping+PEGASUS	0.2087	0.1192	0.0838

The ROUGE scores obtained after fine-tuning the mBART model on this dataset were comparable to the Translation+Mapping+PEGASUS model.

5.3.3. Fine-tuning XL-Sum

We used the XLSum model, an mT5 model fine-tuned on the multilingual XLSum dataset. We used the checkpoint 'csebuetnlp/mT5_multilingual_XLSum' available on Hugging Face to generate inferences on the Gujarati dataset. The model was trained for 5 epochs with the max-tokens parameter set to 75.

6. Evaluation Metrics

In our study, the ROUGE Score, which stands for Recall-Oriented Understudy for Gisting Assessment, was chosen as the evaluation metric [48]. For our summary, we recorded ROUGE-1, ROUGE-2, and ROUGE-4 scores. ROUGE-1 calculated the unigram overlap between the candidate and reference summaries, whereas ROUGE-2 assessed the bigram similarities between

Table 6
Best scores obtained by teams in the validation phase

Language	Best performing team	ROUGE-1	ROUGE-2	ROUGE-4
English	MT-NLP IIIT-H	0.5685	0.4592	0.4335
Hindi	HakunaMatata	0.6104	0.5152	0.4755
Gujarati	MT-NLP IIIT-H	0.2620	0.1644	0.1216

Table 7
Best scores obtained by teams in the test phase

Language	Best performing team	ROUGE-1	ROUGE-2	ROUGE-4
English	MT-NLP IIIT-H	0.5583	0.4458	0.4180
Hindi	MT-NLP IIIT-H	0.6072	0.5102	0.4711
Gujarati	MT-NLP IIIT-H	0.2611	0.1651	0.1241

the summaries. All ROUGE scores are graded out of one, with the ROUGE score closer to one, indicating more parallel with the gold summaries.

7. Results

Tables 2, 3 and 4 describe the results obtained on our approaches by testing on the validation data. Table 5 describes the results obtained on the models we submitted during the test phase. We evaluated the performance of the models using ROUGE scores as the evaluation metrics. The best performing approaches were: fine-tuned PEGASUS model with max-tokens set to 65 for English, the IndicBART model with right-shift operation for Hindi, and Translation+Mapping+PEGASUS based approach for Gujarati. We achieved optimum accuracies with these approaches on both the validation and test datasets.

In the validation round of the competition, we were ranked second in English, fourth in Hindi, and third in Gujarati. In the test phase, we were ranked third in English and fourth in Hindi and Gujarati.

8. Competition Results

Tables 6 and 7 depict the best scores obtained in the ILSUM shared task for each language and the team obtaining the same. Our research being motivated by a shared task- there were other teams in the competition to achieve the most optimum results for English, Hindi, and Gujarati. This section of the paper is designed to give readers a thorough comparison between the findings from our study and the top outcomes from the shared task.

9. Conclusion and Future Work

Thus, we have illustrated the findings of our research which we performed in the ILSUM shared task in collocation with FIRE 2022 [49]. We have experimented with text summarization on news articles written in English, Hindi, and Gujarati. We implemented pre-trained models in our research and data manipulation operations performed in some of the operations. Finally, we evaluate the ROUGE scores on the inferences obtained from each system we trained. We achieved decent accuracy on our best-performing models, with accuracies very close to SoTA accuracies. We can conclude from this analysis that there is a lot of scope for improvement in research performed for low-resource Indian languages, such as Gujarati, compared to English. The research foundation for text summarization for English is robust, as there are many pre-trained models and attention-based mechanisms that one can leverage. However, this foundation has to be scaled up drastically in the coming years for Hindi and Gujarati.

In the future, we plan to leverage our work on larger datasets, especially for Hindi and Gujarati, as we believe that clean and well-formatted datasets are one of the significant barriers that cause the gap between text summarization research in English and low-resource Indian languages. Furthermore, we plan to implement our approaches on high-end GPUs and use better preprocessing and tokenization techniques to shorten this research gap.

10. Acknowledgements

This research was accomplished as part of the L3Cube Pune mentoring program. We convey our gratitude to our L3Cube mentors for their continuous assistance and encouragement.

References

[1] A. Vhatkar, P. Bhattacharyya, K. Arya, Survey on text summarization, 2020.

[2] V. Dehru, P. Tiwari, G. Aggarwal, B. Joshi, P. Kartik, Text summarization techniques and applications, IOP Conference Series: Materials Science and Engineering 1099 (2021) 012042. doi:10.1088/1757-899X/1099/1/012042.

[3] N. Moratanch, C. Gopalan, A survey on extractive text summarization, 2017, pp. 1–6. doi:10.1109/ICCCSP.2017.7944061.

[4] N. Moratanch, C. Gopalan, A survey on abstractive text summarization, 2016, pp. 1–7. doi:10.1109/ICCPCT.2016.7530193.

[5] M. Kirmani, N. Hakak, M. mohd, M. Mohd, Hybrid Text Summarization: A Survey: Proceedings of SoCTA 2017, 2019, pp. 63–73. doi:10.1007/978-981-13-0589-4_7.

[6] D. Sahoo, A. Bhoi, R. C. Balabantaray, Hybrid approach to abstractive summarization, Procedia Computer Science 132 (2018) 1228–1237. URL: https://www.sciencedirect.com/science/article/pii/S1877050918307701. doi:https://doi.org/10.1016/j.procs.2018.05.038, international Conference on Computational Intelligence and Data Science.

[7] H. P. Luhn, The automatic creation of literature abstracts, IBM Journal of Research and Development 2 (1958) 159–165. doi:10.1147/rd.22.0159.

[8] F. Mohsen, J. Wang, K. Al-Sabahi, A hierarchical self-attentive neural extractive summarizer via reinforcement learning (hsasrl), Applied Intelligence 50 (2020) 2633–2646.

[9] J. Xu, G. Durrett, Neural extractive text summarization with syntactic compression, arXiv preprint arXiv:1902.00863 (2019).

[10] N. Alami, M. Meknassi, N. En-nahnahi, Enhancing unsupervised neural networks based

text summarization with word embedding and ensemble learning, Expert systems with applications 123 (2019) 195–211.

[11] D. Anand, R. Wagh, Effective deep learning approaches for summarization of legal texts, Journal of King Saud University-Computer and Information Sciences (2019).

[12] 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.

[13] X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huo, J. Qiu, Y. Yao, A. Zhang, L. Zhang, W. Han, M. Huang, Q. Jin, Y. Lan, Y. Liu, Z. Liu, Z. Lu, X. Qiu, R. Song, J. Tang, J.-R. Wen, J. Yuan, W. X. Zhao, J. Zhu, Pre-trained models: Past, present and future, AI Open 2 (2021) 225–250. URL: https://www.sciencedirect.com/science/article/pii/S2666651021000231. doi:https://doi.org/10.1016/j.aiopen.2021.08.002.

[14] J. Zhang, Y. Zhao, M. Saleh, P. J. Liu, Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, in: Proceedings of the 37th International Conference on Machine Learning, ICML'20, JMLR.org, 2020.

[15] Y. Liu, P. Liu, D. Radev, G. Neubig, BRIO: Bringing order to abstractive summarization, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 2890–2903. URL: https://aclanthology.org/2022.acl-long.207. doi:10.18653/v1/2022.acl-long.207.

[16] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, et al., Exploring the limits of transfer learning with a unified text-to-text transformer., J. Mach. Learn. Res. 21 (2020) 1–67.

[17] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, arXiv preprint arXiv:1908.10084 (2019).

[18] Y. Liu, Fine-tune bert for extractive summarization, arXiv preprint arXiv:1903.10318 (2019).

[19] R. Dabre, H. Shrotriya, A. Kunchukuttan, R. Puduppully, M. Khapra, P. Kumar, IndicBART: A pre-trained model for indic natural language generation, in: Findings of the Association for Computational Linguistics: ACL 2022, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 1849–1863. URL: https://aclanthology.org/2022.findings-acl.145. doi:10.18653/v1/2022.findings-acl.145.

[20] T. Hasan, A. Bhattacharjee, M. S. Islam, K. Mubasshir, Y.-F. Li, Y.-B. Kang, M. S. Rahman, R. Shahriyar, XL-sum: Large-scale multilingual abstractive summarization for 44 languages, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online, 2021, pp. 4693–4703. URL: https://aclanthology.org/2021.findings-acl.413. doi:10.18653/v1/2021.findings-acl.413.

[21] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, L. Zettlemoyer, Multilingual denoising pre-training for neural machine translation, Transactions of the Association for Computational Linguistics 8 (2020) 726–742.

[22] P. B. Baxendale, Machine-made index for technical literature—an experiment, IBM Journal of Research and Development 2 (1958) 354–361. doi:10.1147/rd.24.0354.

[23] H. Oliveira, R. Ferreira, R. Lima, R. D. Lins, F. Freitas, M. Riss, S. J. Simske, Assessing shallow sentence scoring techniques and combinations for single and multi-document

summarization, Expert Syst. Appl. 65 (2016) 68–86. URL: https://doi.org/10.1016/j.eswa.2016.08.030. doi:10.1016/j.eswa.2016.08.030.

[24] K. Aitken, V. V. Ramasesh, Y. Cao, N. Maheswaranathan, Understanding how encoder-decoder architectures attend, 2021. URL: https://arxiv.org/abs/2110.15253. doi:10.48550/ARXIV.2110.15253.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[26] H. Yu, Summarization with attention-based deep recurrent neural networks, 2017.

[27] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–80. doi:10.1162/neco.1997.9.8.1735.

[28] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014. URL: https://arxiv.org/abs/1412.3555. doi:10.48550/ARXIV.1412.3555.

[29] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. URL: https://aclanthology.org/2020.acl-main.703. doi:10.18653/v1/2020.acl-main.703.

[30] L. Lebanoff, F. Dernoncourt, D. S. Kim, W. Chang, F. Liu, A cascade approach to neural abstractive summarization with content selection and fusion, in: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, Suzhou, China, 2020, pp. 529–535. URL: https://aclanthology.org/2020.aacl-main.52.

[31] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[32] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck,

J. Dean, S. Petrov, N. Fiedel, Palm: Scaling language modeling with pathways, 2022. URL: https://arxiv.org/abs/2204.02311. doi:10.48550/ARXIV.2204.02311.

[33] V. Sanh, A. Webson, C. Raffel, S. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, A. Raja, M. Dey, M. S. Bari, C. Xu, U. Thakker, S. S. Sharma, E. Szczechla, T. Kim, G. Chhablani, N. Nayak, D. Datta, J. Chang, M. T.-J. Jiang, H. Wang, M. Manica, S. Shen, Z. X. Yong, H. Pandey, R. Bawden, T. Wang, T. Neeraj, J. Rozen, A. Sharma, A. Santilli, T. Fevry, J. A. Fries, R. Teehan, T. L. Scao, S. Biderman, L. Gao, T. Wolf, A. M. Rush, Multitask prompted training enables zero-shot task generalization, in: International Conference on Learning Representations, 2022. URL: https://openreview.net/forum?id=9Vrb9D0WI4.

[34] D. Wang, P. Liu, Y. Zheng, X. Qiu, X. Huang, Heterogeneous graph neural networks for extractive document summarization, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 6209–6219. URL: https://aclanthology.org/2020.acl-main.553. doi:10.18653/v1/2020.acl-main.553.

[35] M. Zhong, P. Liu, Y. Chen, D. Wang, X. Qiu, X. Huang, Extractive summarization as text matching, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 6197–6208. URL: https://aclanthology.org/2020.acl-main.552. doi:10.18653/v1/2020.acl-main.552.

[36] K. V. Kumar, D. Yadav, A. Sharma, Graph based technique for hindi text summarization, in: J. K. Mandal, S. C. Satapathy, M. Kumar Sanyal, P. P. Sarkar, A. Mukhopadhyay (Eds.), Information Systems Design and Intelligent Applications, Springer India, New Delhi, 2015, pp. 301–310.

[37] A. N. Gulati, S. D. Sawarkar, A novel technique for multidocument hindi text summarization, 2017 International Conference on Nascent Technologies in Engineering (ICNTE) (2017) 1–6.

[38] M. Gupta, N. K. Garg, Text summarization of hindi documents using rule based approach, 2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE) (2016) 366–370.

[39] A. Jain, A. Arora, J. Morato, D. Yadav, V. Kumar K, Automatic, J. Szymanski, H. Mora, D. Logofătu, A. Sobecki, D. Jain, Automatic text summarization for hindi using real coded genetic algorithm, Applied Sciences 12 (2022). doi:10.3390/app12136584.

[40] P. Patel, Pre-processing phase of text summarization based on gujarati language, International Journal of Innovative Research in Computer Science Technology ISSN (2014) 2347–5552.

[41] S. Sarica, J. Luo, Stopwords in technical language processing, PLOS ONE 16 (2021) e0254937. URL: https://doi.org/10.1371%2Fjournal.pone.0254937. doi:10.1371/journal.pone.0254937.

[42] R. Sennrich, B. Haddow, A. Birch, Neural machine translation of rare words with subword units, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 1715–1725. URL: https://aclanthology.org/P16-1162. doi:10.18653/v1/P16-1162.

[43] M. Dwarampudi, N. V. S. Reddy, Effects of padding on lstms and cnns, 2019. URL: https://arxiv.org/abs/1903.07288. doi:10.48550/ARXIV.1903.07288.

[44] J. Zhang, Y. Zhao, M. Saleh, P. J. Liu, Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2019. URL: https://arxiv.org/abs/1912.08777. doi:10.48550/ARXIV.1912.08777.

[45] X. Wang, W. Chen, M. Saxon, W. Y. Wang, Counterfactual maximum likelihood estimation for training deep networks, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems, volume 34, Curran Associates, Inc., 2021, pp. 25072–25085. URL: https://proceedings.neurips.cc/paper/2021/file/d30d0f522a86b3665d8e3a9a91472e28-Paper.pdf.

[46] Y. Liu, Fine-tune bert for extractive summarization, ArXiv abs/1903.10318 (2019).

[47] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, C. Raffel, mT5: A massively multilingual pre-trained text-to-text transformer, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 483–498. URL: https://aclanthology.org/2021.naacl-main.41. doi:10.18653/v1/2021.naacl-main.41.

[48] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.

[49] 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.