Exploring Text Summarization Models for Indian Languages

Shayak Chakraborty[†], Darsh Kaushik[†], Sahinur Rahman Laskar[†] and Partha Pakray[†]

Department of Computer Science and Engineering, National Institute of Technology, Silchar

Abstract

In the task of Indian Language Summarization presented by FIRE 2022, various methods of text, and summarization has been studied and used by the team TextSumEval. The summarization of mixed corpus languages is important as most articles and documents in India contain excerpts from English or other languages. For the summarization of such languages, a variety of summarization techniques has been studied. Finally, LSTM based sequence-to-sequence model, BART model, GPT model, and T5 model have been studied and experimented with and the results have been concluded.

Keywords

Text Summarization, Indian Language Summarization, Abstractive Text Summarization, Deep Learning

1. Introduction

With the advent of time, there has been a lot of textual information that has come into prevalence. Articles, magazines, and other documents contain a lot of insignificant text which might be difficult to read through because of time scarcity. Summarization of text so that the original text is reduced without losing information is hugely beneficial in these scenarios. Creating precise summaries from a long document has been a very important task throughout ages. This task has been simplified by the advent of automatic text summarization tools in recent times.

Automatic text summarization reduces volumes of text data into summaries which would have been very difficult if it had to be done manually. Automatic text summarization is mainly classified into two main types. The first type is extractive text summarization. In this type of model and algorithm, the summary is created by extracting words from the original document which usually have a higher frequency or have some importance in the sentences. The generated summary has almost all the words from the original document. However, because of the extraction process, the generated summary can produce a lot of erroneous sentences.

The other type of automatic text summarization is abstractive text summarization. In this approach, the models change the sentences into shorter sentences while retrieving the complete context from the original document. This method uses deep learning architectures which are

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

*Corresponding author.

[†]These authors contributed equally.

shayak_pg_21@cse.nits.ac.in (S. Chakraborty); darsh_ug@cse.nits.ac.in (D. Kaushik); sahinurlaskar.nits@gmail.com (S. R. Laskar); partha@cse.nits.ac.in (P. Pakray)

https://github.com/ShayakC98 (S. Chakraborty)

able to learn the summarization task from the document to summary pairs. Deep learning models have significantly improved the quality of text summarization over traditional summarization methods such as frequency-based summarization, lexical ranking-based summarization, and Latent Semantic analysis-based summarization.

When it comes to using deep learning models for text generation models, the most commonly used architectures are that of sequence-to-sequence models. Among these kinds of architectures some are as follows:

- Encoder-Decoder architecture - This type of model typically uses LSTM cells with sometimes added attention units. This model is not always useful for processing inputs with long sentences.
- Transformer-based summarization model - This incorporates positional encodings along with attention heads. This improves the working of the LSTM units.
- BERT model trained for summarization - BERT introduced a word tokenizer which reduced the size of the vocabulary. This model has an Encoder layer without a decoder layer. This is useful for language modeling tasks but not suitable for language generation tasks.
- GPT model trained for summarization - This model has a Decoder layer without a proper encoder layer. Although this model is mainly used for text generation, it requires very specific fine-tuning for multilingual corpora.

In this paper, as per the Indian Languages Summarization task under Fire 2022 [1] [2], the multilingual T5 model was used and was fine-tuned for the summarization of the documents. This model uses an encoder-decoder architecture along with a sentence piece tokenizer which is suitable for multilingual corpora and being pre-trained on huge corpora the fine-tuning becomes easier. In the following section, we look at examples of how summarization tasks have been developed. Section 3 discusses the experimentation that have been conducted and the results have been concluded in Section 4.

2. Background

Automatic text summarization was initially done by extracting sentences by scoring them using Bayesian models [3] and term frequency-inverse document frequency models [4]. These methods were good for extracting entire sentences from the document. However, because these methods relied on the frequency of words from a sentence, if a word was a special name or a stop word it would have a very low chance of getting produced in the summary.

With the advent of machine learning algorithms, the problem of sentence summarization changed. Deep learning-based models were built for language modeling tasks. With the advent of Recurrent Neural Networks sequence-to-sequence, models came into existence. Recurrent neural networks were then used for sequence generation [5]. These sequence-to-sequence models were then used to model abstractive text summarization task-based problems [6]. The working of RNN-based models could be improved by using Long Short Term Memory cells [7] as suggested by [8].

However, these architectures had a problem of losing context for sequences that were long. Even LSTM cells are not very capable of keeping context from long sequences. This problem was solved by [9] in 2017. Transformer architectures solved the problem of long sequence processing by using positional encodings and multi-headed attention to keep information from different parts of the sequences. This architecture was soon extended to various other models to improve on the language modeling task.

Bidirectional Encoder Representations from Transformers [10] was a transformer-based architecture that pre-trained the encoder to improve the working on many NLP tasks such as question answering. BERT also used a word piece tokenizer to improve vocabulary maintenance. This model was used further for text summarization by [11]. Soon after BERT, the BART model [12] was developed. This model improves the sequence processing capabilities by denoising the BERT model. The BART model was trained by adding noise to the text and then it was required to reconstruct the text. This made the model more robust than simple BERT model for the specific language modeling tasks.

After the foundation of pretraining was laid, models like GPT [13] and Pegasus [14] were introduced. Although these models were developed for summarization tasks still they did not produce good results for multilingual corpora. The GPT model introduced the strong decoder which enabled it to generate proper sentences as output. However without a strong encoder, the model lacked the ability to work on multiple language modeling tasks.

The T5 [15] model when pre-trained with XL-sum [16] dataset was best suited to produce summaries for multilingual corpora. The T5 model has been trained by systemic transfer learning methodology. By training the model across multiple tasks the T5 model has come to produce the best solutions for various language modelling tasks. The T5 model does not require task specific retraining in most of the scenarios.

The dataset given by the Indian Languages Summarization task has three main subtasks. Each subtask was for a given language. There were three main languages - English, Hindi, and Gujarati. The given dataset contained mixed corpora - implying that the English dataset file contained Hindi/Gujarati words that had to be processed during summarization and similarly for the other datasets other languages were also present. Each dataset file contains two main columns having the Articles and their respective summary. These two columns were mainly used for training the model for summarization tasks. In the following section, the experiments and results have been discussed in detail.

3. Experiments and Results

For each given dataset the two columns were extracted from the dataset - Articles and Summaries. For preprocessing the article and summaries were stripped of HTML tags. Multiple punctuations and emoticons were removed from the columns. Since there could be proper nouns and other characters from different languages the text was not entirely converted to lowercase and neither were the non-English words removed.

For the English dataset, the summarization was tested with four different models. The LSTM with attention model which was a sequence to sequence model was the first model. It was quickly eliminated as it produced completely arbitrary results. The rouge score for the given

Table 1
Validation scores for used models over the English subtask

Model	Rouge-1	Rouge-2	Rouge-3	Rouge-4
mBART	0.38	0.23	0.19	0.17
GPT	0.46	0.38	0.35	0.34
T5	**0.48**	**0.35**	0.33	**0.32**

Table 2
Test values for English subtask (submission id: TextSumEval - t5 small)

Model/Team Name	Rouge-1	Rouge-2	Rouge-3	Rouge-4
MT-NLP IIIT-H (top scorer)	0.56	0.44	0.43	0.42
Ours (T5)	0.48	0.35	0.33	0.32

model was about 0.01%. This behavior was justified as for this particular model the vocabulary was built using a count vectorizer. With having lost a lot of words that occurred less than one or two times the predictions of the model deteriorated. This was followed by three important models - mBART, GPT, and T5 model. Table 1 shows the validation scores of the following models: The mBART model clearly underperforms in the task. This might be as a result of improper tokenization between the multiple languages and the fact that the mBART model which is based on the transformer architecture lacks a strong decoder which does not allow the model to generate proper summaries despite proper training.

The GPT model which has been used is mainly used for generating text. It can be seen that it outperforms the mBART model by a considerable difference in scores. The GPT model was pre-trained on a huge corpus but when it was fine-tuned on the other language dataset, that is for the Hindi dataset, it produced ambiguous output. This was probably a result of not being trained on the specific task of summarization before. Also, the embeddings of GPT model for multilingual corpora are not built well as the GPT model does not have a powerful encoder system. It reduces the language modeling capacity of the model.

The T5 model (marked as bold in Table 1) which outperformed both the mBART model and the GPT model was pre-trained on a huge dataset. The T5 model was also trained for a variety of tasks including summarization. The T5 model was fine-tuned for 20 epochs using a batch size of 4 and truncating the input length of the original articles to a length of 250. The sentence piece tokenizer used by the T5 model produces a byte pair encoded value for words that are not present in the vocabulary. By using this method the T5 model is able to reproduce the proper nouns and other words from the other languages' corpus which is not present in the original corpus of the dataset. The T5 model provided the best scores for the test run for the English subtask with submission name : TextSumEval - t5 small, as shown in Table 2.

Since the T5 model worked best for the English dataset, a multilingual variant of the T5 model was used to fine-tune the Gujarati and Hindi subtask datasets. The mT5 model [16] that was used for fine-tuning was trained on the XL-Sum dataset after it was pre-trained on the mC4 corpus [17]. This model showed the best performance for summarization for Indian Languages. During training, the model achieved a rogue-1 score of 32 upon the Hindi subtask.

Overall using the T5 model and its variants the summarization task for Indian Languages has

shown marginal improvements over pre-existing models which are already being used. The following section concludes the paper.

4. Conclusion

In this task of Indian Languages summarization under FIRE 2022 first, a list of text summarization methods has been studied. The best methods for abstractive text summarization have been chosen. The experiments have been performed and their results have been noted down accordingly. First, the BART model has been studied which shows the least scores. Followed by the GPT model which gives slightly better results. Finally, the T5 model which gave the highest scores was used for the summarization of English-based sentences containing non-English words. This led to the experimentation of using a trained mT5 model for summarization of other Indian Languages subtasks namely Hindi and Gujarati, which produced considerable scores during training.

References

[1] 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.

[2] 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.

[3] T. Nomoto, Bayesian learning in text summarization, in: HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada, The Association for Computational Linguistics, 2005, pp. 249–256.

[4] H. Christian, M. P. Agus, D. Suhartono, Single document automatic text summarization using term frequency-inverse document frequency (tf-idf), ComTech: Computer, Mathematics and Engineering Applications 7 (2016) 285–294.

[5] A. Graves, Generating sequences with recurrent neural networks, arXiv preprint arXiv:1308.0850 (2013).

[6] R. Nallapati, B. Zhou, C. N. dos Santos, Ç. Gülçehre, B. Xiang, Abstractive text summarization using sequence-to-sequence rnns and beyond, in: Y. Goldberg, S. Riezler (Eds.), Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016, ACL, 2016, pp. 280–290.

[7] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.

[8] T. Shi, Y. Keneshloo, N. Ramakrishnan, C. K. Reddy, Neural abstractive text summarization with sequence-to-sequence models, Trans. Data Sci. 2 (2021) 1:1–1:37.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems

30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 5998–6008.

[10] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186.

[11] Q. Wang, P. Liu, Z. Zhu, H. Yin, Q. Zhang, L. Zhang, A text abstraction summary model based on bert word embedding and reinforcement learning, Applied Sciences 9 (2019) 4701.

[12] 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, ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, 2020, pp. 7871–7880.

[13] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving language understanding by generative pre-training (2018).

[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 2020, 13-18 July 2020, Virtual Event, volume 119 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 11328–11339.

[15] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020) 140:1–140:67.

[16] T. Hasan, A. Bhattacharjee, M. S. Islam, K. S. Mubasshir, Y. Li, Y. 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, Online Event, August 1-6, 2021, volume ACL/IJCNLP 2021 of *Findings of ACL*, Association for Computational Linguistics, 2021, pp. 4693–4703.

[17] 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, NAACL-HLT 2021, Online, June 6-11, 2021, Association for Computational Linguistics, 2021, pp. 483–498.