Utilizing Optical Character Recognition to Summarize Bengali Text

Abhijit Saha, Faiza Bushra, Ahmed Anwar, Rubaba Rashid, Md Sabbir Hossain, Ehsanur Rahman Rhythm and Annajiat Alim Rasel
Department of Computer Science and Engineering (CSE)
School of Data and Sciences (SDS)
Brac University
66 Mohakhali, Dhaka - 1212, Bangladesh
{abhijit.saha, faiza.bushra, ahmed.anwar, rubaba.rashid, md.sabbir.hossain1, ehsanur.rahman.rhythm}@g.bracu.ac.bd, annajiat@gmail.com

***Abstract*—Optical Character Recognition (OCR) has proven to be a valuable tool for extracting textual information from images, converting physical documents into digital formats that can be easily edited, searched, and shared. However, OCR technology is not without its limitations, particularly when it comes to recognizing and processing non-Latin scripts such as Bengali. Over 230 million people speak Bengali, making it one of the most widely used languages in the world. Summarizing Bengali material is one area where OCR could be especially helpful, saving time and effort for scholars and journalists who are working with a big corpus of text. In recent years, there have been several attempts to develop OCR-based text summarization techniques for Bengali. One challenge it struggles with is the complex nature of the language, its rich vocabulary, its highly inflected grammar and the inclusion of a number of ligatures in Bengali Script and diacritics that can significantly complicate the OCR process. Overall, the development of OCR-based text summarization could save time and effort, while also providing a more objective and systematic approach to analyzing large volumes of text. It is expected that in the future, Bengali text summarizing methods based on OCR will become more advanced and precise as technology advances.**

***Index Terms*—OCR, Bengali, Corpus, Summarization, Ligatures, Data Preprocessing, LDA, RNN, Accuracy, BLEU**

I. Introduction

OCR, or Optical Character Recognition, is a technology that scans printed or handwritten text and converts it into a digital format that may then be modified, searched, and saved digitally. Digitizing text required a lot of time and effort before the advent of OCR technology. This was inefficient because of the time it took and the potential for mistakes that may have negative effects on accuracy and output.

The goal of developing optical character recognition technology was to speed up and improve the accuracy of the method of digitizing text. In order to detect and digitize printed or handwritten text, OCR software use complex algorithms to evaluate the forms and patterns of the letters. This facilitates the rapid digitization of vast quantities of paper records, allowing users to save time and ensure precision in their work. By automatically turning scanned documents into text that is machine-readable, OCR technology does away with the necessity for human data input. This allows documents to be scanned and translated into other languages through machine translation algorithms, saving time and improving accuracy compared to manual data entry and making printed materials obtainable to people with visual disabilities by turning them into digital text that can be examined aloud by screen readers.

Despite the potential of the technology, it still has trouble with non-Latin scripts. The Bangla language has several significant historical and writings published in it. Enhancing education by facilitating the sharing of educational resources across regions and countries, the development of Bangla OCR technology can aid in the preservation of this cultural heritage by making it simpler to digitize and store historical and written works in digital format.

However, OCR technology has struggled to accurately detect and digitize Bengali text due to the complexity of Bengali script. The researchers that need to deal with Bengali texts have faced a significant challenge because of this: it is time-consuming and inefficient to search and analyze vast amounts of Bengali text. While Bangla OCR technology has come a long way since the 1980s, current OCR systems still fall short of ideal accuracy. As a consequence, there is a lot of focus on developing better Bangla OCR systems. However, with the tools at our disposal now, a full-fledged Bangla OCR is not only feasible, but its use can yield accurate results and reduce processing time.

II. Literature Review

One of most researchable topic in NLP is text summarization. OCR (Optical Character Recognition) technology can convert printed text or scanned images into machine-readable digital text. Because this digitization automates the extraction of relevant information from documents, it reduces the requirement of manual summarization efforts and it is mostly used in text summarization process.

Many studies have been conducted in this field for various languages.To avoid ambiguity or contradiction, the process of calculating similarity via a basic pattern matching algorithm

requires the possession of an exhaustive set of patterns for each meaning. The difficulty and time-consuming nature of manually compiling is tedious task. Recent natural language processing applications highlight the need for an efficient method to obtain the similarity between very textual data or sentences [1].The utilization of a lexical dictionary to evaluate the similarity of a set of words collected from several sentences being compared is one extended form of word co-occurrence methods. Sentence similarity can be calculated simply by adding the similarity values of all word pairs[[2]] .There is also a model that uses a reinforcement technique for abstractive text summarization and employs convolutional sequence learning. The central concept of this work is textual learning in context. The convolutional neural network consolidates the entire process [3].

III. OCR and Summarization Approaches

1) OCR: For the purpose of extracting Bangla text from images we will be using Bengali dataset that has Bengali images and their corresponding texts. Later we will use the pytesseract model using Bengali dataset that will be able to recognize Bengali texts from images fed to the model.

2) Summarisation: There are three major approaches for summarisation - extractive based, abstractive based and keyword based approach. We will be working on abstractive summarisation approach which has the following steps:

1) Data Collection
2) Data Preprocessing
3) Data Splitting
4) Model Selection
5) Fine-tuning
6) Summarization
7) Evaluation

IV. DATA COLLECTION AND PREPROCESSING

A. Data Collection

For OCR: To apply OCR technique we made our own dataset by taking around more than 300 pictures of text paragraphs written in Bengali language. Alongside we also collected the text data of those pictures and stored them to check the accuracy of OCR. .

For Summarization: To train a model for the purpose of summarization we collected around 23422 texts from the Prothom Alo Newspaper and their subsequent summarized texts. The texts were categorized into 7 types including-

1) Business News Data
2) Entertainment News Data
3) Opinion News Data
4) Politics News Data
5) Sports News Data
6) World News Data
7) Bangladesh News Data

Subsequent csv files were made for each type for data preprocessing. Each csv file had 6 columns- Title, Description, Meta-Summary, Summary, Syndicate-catagorys, Keywords-for-related-articles. An illustration of which is shown below:

Fig. 1. Dataset of Summarization

B. Data Preprocessing

For OCR: In our dataset, we discovered a wide variety of images that required preprocessing before the pytesseract model could be applied to extract texts. We used normalization, skew correction, image scaling, noise removal, thresholding, binarization, skeletonization, thinning, etc as preprocessing techniques for this purpose. To begin, we flipped the pictures into monochrome gray. The picture quality was then improved by eradicating any noise or distortions that might compromise OCR readability. The grayscale picture was then thresholded to create a binary one. The picture is then transformed into a monochrome version, with the words appearing in black on a white backdrop. Next, we rotate the image to straighten out the text lines if they are not already perfectly horizontal. After that, we got rid of the last of the image's imperfections and noise, such as lines and dots. To get the desired level of text thickness in the photos, we next used Thinning and Skeletonization. The next step we took to increase OCR precision was to standardize the size and orientation of the fragmented letters or phrases.

For Summarization: We applied multifarious preprocessing techniques on the collected dataset in csv format to ensure that it was in a suitable format for training our model. For each of the 7 csv files we checked for duplicate and missing/null values. A few duplicate texts were found which might have been mistakenly taken during data collection. We removed the rows with duplicate values. Meanwhile for the null values we used an imputation technique. We got null values for the 'syndicate-catagorys' column of the csv file only. As the dataset was already divided, it was easy to replace the null values with its subsequent category using .fillna() function of the pandas library. Finally after preprocessing, all the refined datasets were combined together into 1 csv file to train our model.

V. Methodology

The first phase is data cleaning, in which irrelevant information like HTML elements, punctuation, extra spaces and stop words is eliminated. Stop words are those that appear frequently in a language but contribute nothing to a summary; hence, it

is important to get rid of them. Since the machine cannot understand a text in its short form, we must define the word's full meaning before adding contractions. The dataset is made more manageable by this method so that it may be thoroughly evaluated.

Tokenizing sentences is the next phase, which entails parsing the text into its component sentences. This is a stage since it helps the summarization model comprehend the organization of the text and the connections between individual phrases. Tokenizing a Bangla phrase effectively requires knowledge of the language's morphology, making it more difficult than in English. Tokenization, the process of parsing phrases into their constituent words, is the third phase. This is a vital stage since it teaches the summarization model the significance of each word and their connections to one another. We will use 'bnlp toolkit' for the purpose of tokenization of our Bengali texts. [8]

Thirdly, the regular expression is used to remove the rare character or unwanted character to remove from texts. Removing spaces, English characters, punctuation form text, Bengali digit from text is the prime objective of using regular expression in our research.

Data normalization, the fourth phase, involves reformatting the text into an easily analyzed format. In the case of Bangla, this means adapting the text to a uniform script like Unicode. This is a vital stage since it guarantees that the summarization method has a proper grasp of the text.

Data representation is the fifth stage, and it entails expressing the text in a way that the summarization model is able to understand. Bag of Words, TF-IDF, and Word Embeddings are all useful tools for doing this phase, which entails transforming the text into numerical values.

Initially we divide our summarisation dataset into two parts-train and test data. We use 80% of the dataset for training our models and the rest of 20% data is used to test our models.

Now, selecting the best model is essential since it affects the final quality of the summary. To effectively summarize the most crucial points of the input text, a decent model should be able to extract the relevant semantic and contextual information. While there are a variety of models for text summarization, not all of them work well with the Bangla language.

VI. Result and Analysis

VII. Conclusion

Bengali text preprocessing is more challenging than that of other languages. This necessitates the development of a preprocessing library for Bengali text. We have completed our statistical analysis and sample comparisons. Moreover, no machine can guarantee a 100% reliable outcome. Every device has a restricted range of applications. The same holds true for our summarizer model. Our research led us to conclude that building our own Bangla WordNet and corpus would improve efficiency and accuracy. Then, and only then, will it have a chance at improved output. Our models may have underperformed because of the data source, especially the summary and article lengths. [9] The model might not grasp the data set's structure. Because our model was being run on Google Colab, it underperformed and frequently crashes whenever we inputted a large batch size due to the GPU memory limit being reached.Future work will hopefully include taking use of pre-trained models and fine-tuning them to carry out summarization. [10] In addition, we want to clean up our data by hand, article by article, to ensure the highest possible quality. This will aid our little model in its pattern-detection efforts, ultimately improving its overall performance.

References

[1] D. Michie, "Return of the Imitation Game,"

[2] Juan-Manuel Torres-Moreno, Automatic Text Summarization (Cognitive Science and Knowledge Management) 1st Edition , 2014.

[3] Gehring, Jonas, et al. "Convolutional sequence to sequence learning." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017

[4] Peter J. Liu et al. "Generating Wikipedia by Summarizing Long Sequences". International Conference on Learning Representation (ICLR), 2018

[5] P. Raundale and H. Shekhar, "Analytical study of Text Summarization Techniques," 2021 Asian Conference on Innovation in Technology (ASIANCON), PUNE, India, 2021, pp. 1-4, doi: 10.1109/ASIANCON51346.2021.9544804.

[6] J. Yan and S. Zhou, "A Text Structure-based Extractive And Abstractive Summarization Method," 2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP), Xi'an, China, 2022, pp. 678-681, doi: 10.1109/ICSP54964.2022.9778497.

[7] Liu, Y. and Lapata, M. (2019) "Text summarization with pretrained encoders," Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) [Preprint]. Available at: https://doi.org/10.18653/v1/d19-1387.

[8] T. Islam, M. Hossain and M. F. Arefin, "Comparative Analysis of Different Text Summarization Techniques Using Enhanced Tokenization," 2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 2021, pp. 1-6, doi: 10.1109/STI53101.2021.9732589.

[9] Gehrmann, S., Deng, Y. and Rush, A. (2018) "Bottom-up abstractive summarization," Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing [Preprint]. Available at: https://doi.org/10.18653/v1/d18-1443.

[10] S. Bal And E. ŞORA GÜNAL, "The Impact of Features and Preprocessing on Automatic Text Summarization," ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY , vol.25, no.2, pp.117-132, 2022