Genre Classification: A Machine Learning Based Comparative Study of Classical Bengali Literature

***Abstract*—Bengali literature, specifically classical Bengali literature has been a source of inspiration, a spark for paradigm-shifting revolutions, and the sole sustaining source of cultural thirst for hundreds of millions of people over many generations. Unfortunately, very few attempts have been made to analyze this never-ending collection of literary works from the luminary figures of Bengali literature. The availability of high-quality research-ready datasets comprising all the authenticated literary works has been a key obstacle in conducting NLP research, utilizing the most recent advancements in deep learning and large language models. Identifying the genre of a given text snippet is a key step in analyzing a vast collection of works comprising different styles, themes, and motivations from classical authors. From classifying previously unexplored archival documents to identifying and suggesting similar literary works for modern recommender engines, genre classification opens the door for many downstream and specialized use cases. In this project, we initiate an ambitious goal of compiling a comprehensive dataset of literary works from classical authors and eventually extending the collection to contemporary writers as well. We explore both classical methods such as Naive Bayes as well as LSTM and recent transformer-based models to classify genre from short text snippets. We concluded that fine-tuning pre-trained BERT models produced much higher accuracy than both classical and LSTM models.**

***Index Terms*—Genre Classification, Bengali, Classical Literature, Naive Bayes, LSTM, Transformers, BERT.**

I. Introduction

A handful of classical literary giants have provided the cultural and historical fuel for generations of Bengalis in times of hope and despair alike. Bengali is one of the world's most spoken languages. Yet the depth of literary analyses of this beautiful language is scant. The main ingredient of such analyses, annotated and pre-processed authentic datasets has been but a few and consequently restraining the researchers to create their own datasets using OCR and other complicated methods. A huge amount of time is spent on dataset preparation rather than actual analyses. We aimed to bridge this gap and enable future researchers to spend the bulk of their efforts on applying emerging techniques to Bengali content. Toward that goal, we have compiled four datasets so far from the works of four classical writers, namely Rabindranath Tagore, Kazi Nazrul Islam, Sarat Chandra Chattopadhyay, and Bankimchandra Chattopadhyay.

Accurately classifying genres of classical works will enable us to easily and efficiently parse through a vast collection of archival documents and categorize them or find similar works in terms of styles, themes, narratives, and other literary factors. It can be used for further downstream tasks of identifying and recommending similar works and comparing them with contemporary works to map the evolution of modern works from the classical ancestors. In this project, we applied both classical machine learning techniques such as Naive Bayes as well as modern deep learning algorithms such as LSTM and Transformer-based BERT models to classify genres of literary works given a short snippet from any of the categories such as essay, novel, poem or story. Given the vast diversity and unique literary styles of these authors, spanning a multitude of themes and topics, we attempted to provide a benchmark for future research works regarding the classification task.

II. Related Work

Waleed A. Yousef et al. trained recurrent neural networks (RNN) at the character level on English and Arabic written poems in order to learn and identify the meters that give the poems their phonetic pronunciation. Datasets crawled from non-technical sources were cleaned, formatted and published to a publicly accessible repository for scientific research [1].

The work of Parilkumar Shiroya et al. compares three machine learning algorithms: K-NN, SVM and LR for classifying books by their genres based on their titles and abstracts. The paper uses a customized dataset that consists of books that are translated to English from Gujarati or Hindi origin books. The paper reports that SVM achieved the highest accuracy and was fast in processing and predicting output among the three algorithms [2].

The work of Joseph Worsham et al. develops approaches for genre identification that can be used with complex and exceedingly large literary works. The paper uses the Gutenberg Dataset and compares current models to traditional methods and evaluates various machine learning approaches, including CNN, LSTM, HAN, Naive Bayes, k-Nearest Neighbors, Random Forests, and XGBoost. XGBOOST outperforms all models, achieving the highest accuracy of 84% among deep learning models [3].

Anshaj Goyal et al. compared machine learning and deep learning approaches for literary genre classification using the same Gutenberg dataset as the previous paper for the experiment. The experiment used three statistical machine learning algorithms (Random Forest, Naïve Bayes, and SVM) and four deep learning approaches (LSTM, CNN, RNN, and BERT) on the entire texts for classification. The SVM classifier achieved 85% accuracy in genre prediction among the machine learning models, while the RNN approach performed better among the deep learning models. The RNN approach outperformed all other models in the evaluation [4].

III. Methodology

The methodology for this paper involves:

- Compiling a comprehensive collection of Bengali classical literary texts.
- Building baseline model Naive Bayes for genre classification as performance benchmarks.
- Training deep learning models (e.g., LSTM, Transformer) tailored to the characteristics of Bengali classical literature.
- Analyzing results by evaluating metrics like accuracy, precision, recall, and using confusion matrices and classification reports.

A. Naive Bayes:

Naive Bayes computes classification probabilities based on prior information while assuming feature independence. Utilizing the Bayes theorem, naive Bayes functions by calculating the conditional probabilities of features given classes [5]. It is frequently used for a variety of text categorization applications across several languages. For simplicity, it assumes feature independence and calculates conditional probabilities of class labels based on document features. In [6], similar probabilities are determined by Naive Bayes for the classification of texts in Turkish and Urdu. Naive Bayes frequently performs well in text classification scenarios. The algorithm plays a crucial role in categorization, sentiment analysis, and document labeling across several languages for text classification tasks [7].

B. LSTM

Long Short-Term Memory (LSTM) networks were conceived with the express purpose of capturing and retaining long-range dependencies within sequential data, rendering them highly suitable for a diverse array of Natural Language Processing (NLP) tasks, prominently including text classification. At its core, LSTM sought to address the problem of vanishing and exploding gradients [8]. What sets LSTM apart is its ingenious incorporation of memory cells and gating mechanisms. These mechanisms encompass input gates, forget gates, and output gates. In the realm of text classification, LSTM-based models have left an indelible mark. Applications like sentiment analysis, spam detection, and document categorization have all reaped the benefits of LSTM's prowess.

C. Transformers

Transformers and their related pre-trained language models have become powerful tools in the most recent era of text classification [9]. These models take advantage of the enormous amount of unlabeled data to construct sophisticated representations. The ability to adapt makes their abilities stand out; they excel at both understanding language nuance and adjusting to various downstream duties.

The BERT (Bidirectional Encoder Representations from Transformers) model is one demonstration of these transformers [10]. BERT, with its extensive pre-training on text corpora, has demonstrated exceptional results in text categorization tasks, outperforming traditional methods due to its strong contextual and semantic knowledge.

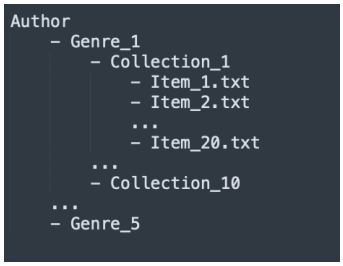


Fig. 1: Content hierarchy

IV. Dataset Compilation

Conducting any significant literary analyses in any language requires verified and authentic datasets in a readily usable format. To compile the datasets from an authentic source in Bengali, we chose the digitized literary works published by the Department of Information Technology & Electronics, Government of West Bengal, India [11]. We have compiled and published all four datasets on Kaggle, namely Rabindranath, Nazrul, Sarat, and Bankim [12]. The overall content hierarchy for each author is described in Figure 1.

There are two parts to dataset compilation, crawling & parsing content and aggregating & formatting the content to usable formats. Both of these segments are described below:

A. Crawling & parsing content

The crawler is created using *requests* and *beautifulsoup* libraries. Starting from the landing page of a genre, the crawler can independently crawl through all the navigation links of that content and create separate text files for each content for further downstream processing.

B. Aggregating & formatting content

Given an author, the aggregator starts from the root directory of that author where the text files were stored during the crawling step. It then traverses the nested directories for different genres and collections and reads the contents of individual items from the text files. Finally, separate CSV and TXT files are created for each genre comprising all individual items found in that genre.

C. Challenges

While crawling the content from the mentioned site, we faced some challenges due to the HTML structure and layouts used in the site. Some of these challenges are described here:

- Each author repository had its own content layout and hierarchy structure for different genres and no uniform pattern in the starting pages.
- Pagination links were broken for some contents which made it difficult for the crawler to crawl the content of a single item in a sequential order.

To overcome these challenges, we used a fallback mechanism to deal with broken pagination links so that if the crawler failed to parse which page to visit next using the default resolver, it used the secondary resolver to identify next-page links.

V. Dataset Analysis & Preprocessing

After conducting an initial analysis of all four datasets, we excluded Bankim from the training data due to its lack of diverse genre distribution. We aggregated the remaining three datasets into a single CSV file. We decided to consider the genres that have a good distribution across all three authors. After analysis, we discarded works from all other genres except these four categories - novel, essay, story, and poem. In our classification tasks, we aim to predict the genre class for a given text from these four categories.

A. Analysis

First, we ran some basic exploratory analysis on the aggregated dataset. From Figure 2, we can see that Rabindranath has written more stories than both Saratchandra and Nazrul. The sheer number of poems by Rabindranath compared to Nazrul is very high. Saratchandra, however, has written more novels compared to the other two. Out of the three, only Rabindranath has a gigantic number of essays, whereas both Saratchandra and Nazrul have written only a handful of essays.

When we looked into the average word count across genres for each author, we noticed something interesting. Even though Rabindranath has written more essays, out of a handful of essays written by Saratchandra, the average word count in the essay genre is much higher than both Nazrul and Rabindranath. All three authors have a similar word count for the novel. For both poem and story genres, Nazrul has a much higher average word count compared to the other two. The average word count in the poem genre for Nazrul is higher than Rabindranath. See Figure 3.

Individually, all three authors have a higher average word count in novels compared to other genres. This is expected since the novel genre has a longer format than the rest.

B. Preprocessing

After aggregating all three datasets, we carried out further preprocessing to remove unnecessary symbols and punctuations as well as splitting the longer items into shorter segments retaining the genre and author. Using a predefined mapping for each genre, the individual sentences are merged into shorter segments. For example, if we have a novel of 1000 lines, and we split it into segments of 10 lines each, then after the split-and-merge step, we will have 100 novel segments.

This splitting step allows us to create a uniform sequence length for each genre since training an LSTM requires inputs of certain lengths. From Figure 4 (a, b), we can see the effect of preprocessing the dataset by splitting and merging. Figure 4 (c) shows the impact of splitting by creating items in all genres with a much closer average word count.

VI. Model Building & Training:

A. Naive Bayes

We used some machine learning techniques, including Multinomial Naive Bayes (MNB) classification, to classify genres in classical Bengali literature. The Term Frequency-Inverse Document Frequency (TF-IDF) vectorization technique was used to convert the textual data into numerical feature vectors. Stratified sampling was used during the train test split to address data imbalance.

B. Deep Learning

For the genre classification task, we used two deep learning models, namely LSTM and transformer-based fine-tuned BERT. For LSTM, we considered both the regular and bi-directional LSTM. We also explored the model performance for single-layer and stacked-LSTM layers for different hyperparameter values.

1) LSTM: In order to build and train an LSTM model, a dynamic custom model builder is used to create a model architecture based on different hyperparameter values. We used Adam Optimizer with a small learning rate. For the loss function, we use 'categorical cross-entropy and 'accuracy' as the performance metric.

Initially, we trained both the regular and bi-directional models for different combinations of various hyperparameters. After extensive experimentation with different values, we finalized the training architecture of the models. We ran both the regular and bi-directional stacked LSTM layers for different batch sizes. To measure the effectiveness of stacked layers, we ran another version of the model by keeping the batch size at 32 but varying the number of layers. We also carried out hyperparameter tuning by tweaking the max vocabulary size and max sequence length.

2) Transformers: For the transformer-based classification, we considered three pre-trained language models trained on the Bengali language and then fine-tuned the models for the genre classification task. BERT and similar encoder architectures consider each token of a text in the context of all the tokens left to it as well as all the tokens right to it [10]. *Bangla-Electra*: It is a language model trained on the Bengali language using ELECTRA from Google. The model was trained on OSCAR crawled dataset and Bengali Wikipedia dump [13]. *Bangla BERT Base*: This is a pre-trained language model using mask language modeling and trained on the same dataset as Bangla-Electra [14]. *BERT Multilingual Base Model (Uncased)*: This model also used masked language modeling, but was trained on Wikipedia for the top 102 languages [15].

VII. Result Analysis

After training the genre classification model, we evaluated its performance using various metrics to gauge its effectiveness in distinguishing between different literary genres.

A. Naive Bayes

The model's accuracy was assessed using testing data. Then confusion matrix was computed, and the classification report provided precision, recall, and f1-score for each genre. We ran Multinomial Naive Bayes separately on each dataset as well as on the aggregated dataset. Our analysis yielded the following insights for each author: accuracy of 64.71%, 87.5%,

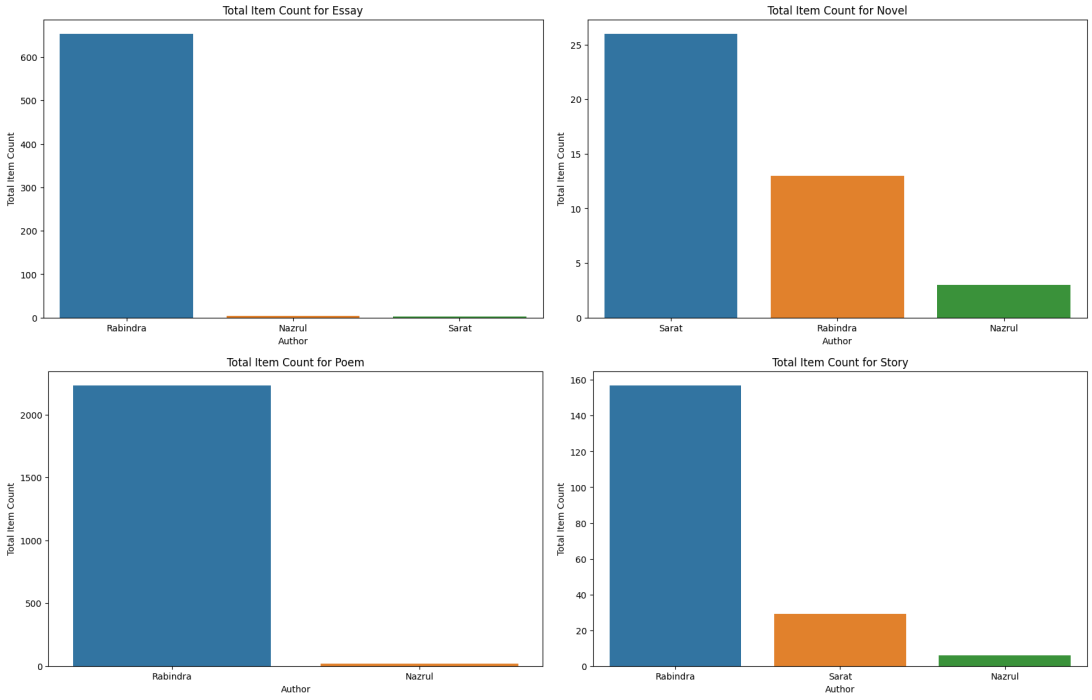


Fig. 2: Total item count by genre

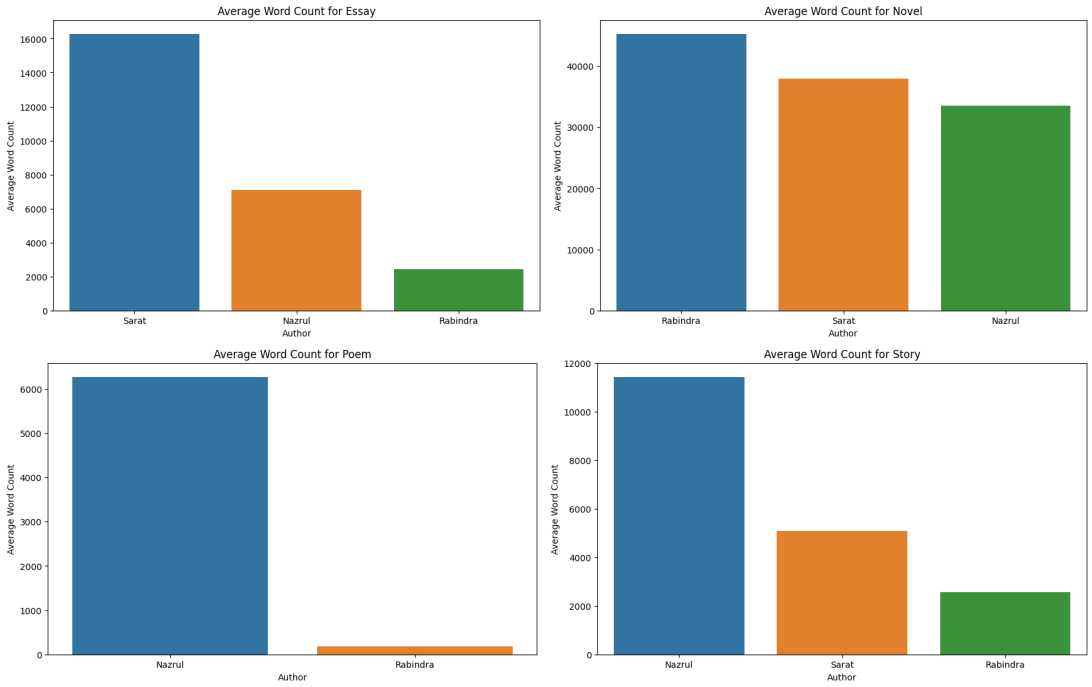
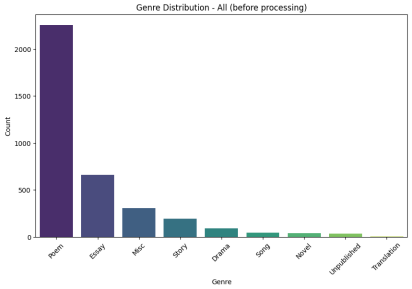
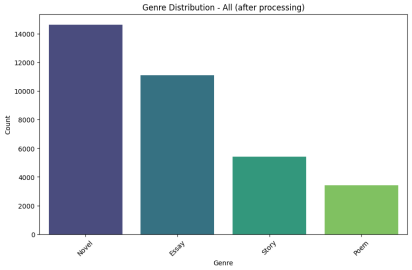


Fig. 3: Average word count by genre



(a) Essay



(b) Novel

```
Dataset loaded...

                max    min          mean
genre
drama         34384     57   6451.528090
essay         48111      4   2525.189107
misc          25071     19   1467.508143
novel        134035  11661  39926.023810
poem          15884      1    242.982705
song          33601     87   4785.543478
story         24427     31   3237.755208
translation    6476    385   3430.000000
unpublished   17025    139   1881.114286
Grouped word count generated...

Dataset preprocessed...

Dataset filtered...

        max  min        mean
genre
essay  1435   10  150.158553
novel   357   15  114.525403
poem    829   10  159.508173
story   399   10  114.558454
Grouped word count generated...
```

(c) Novel

Fig. 4: Effect of content splitting & merging

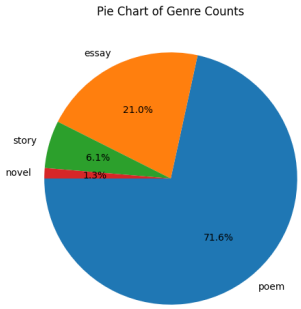


Fig. 5: Genre distribution pie chart

and 43.48% for Nazrul, Rabindranath, and Saratchandra, respectively. Aggregating across authors, resulted in an overall accuracy of 79%. For both separate and aggregated datasets, we used prior probabilities for each class during training and applied the additive smoothing parameter to tackle zero probabilities for words not present in some classes.

Due to dataset imbalance favoring the genres such as poems and essays which have higher numbers of individual items and longer contents respectively see Figure 5, the recall and precision on the aggregated dataset are very high compared to stories and novels which amount to 7.4% of the entire dataset. We applied both undersampling as well as different variants of the oversampling technique to add entries for minority classes. While undersampling produced very low accuracy and f1 scores, SMOTEN oversampling produced slightly better results (overall accuracy of 67%, f1 score of 74%) than Random oversampling (overall accuracy of 65%, f1 score of 72%). Without oversampling the accuracy was 79% and f1 score was 74%. For f1 score, we considered the weighted average score rather than the macro average.

B. LSTM & Fine-tuned BERT

We generated performance measures for all the variants of LSTM models under different hyperparameter values. From Figure 6, we can see that the performance of all variants on the aggregated dataset is quite similar, measured by accuracy, precision, recall, and f1-score. We trained two variants of the models for regular LSTM layers by using precision and recall as performance metrics instead of accuracy. However, the overall accuracy across all the variants is around 70%.

After training the Bangla-Electra model for 15 epochs, we gained an accuracy of 88.4%. After training both Bangla BERT Base and BERT Multilingual Base models for only 5 epochs, we gained accuracy of 92.5% and 84% respectively. For all three models, we noticed a significant drop in loss just after 3 epochs. See Figure 7.

We can see that different variants of LSTM and Bi-directional LSTM models produced similar classification scores across different metrics. However, Transformer-based BERT models produced much higher accuracy than simple LSTM models trained from scratch. Pre-training language models played a significant role in the performance boost of the downstream task, genre classification.

VIII. Future Works

In the future, we plan to try more recent pre-trained models such as the Multilingual Representations for Indian Languages from Google. We also want to conduct a more comprehensive hyperparameter tuning to exhaust all variables and possibilities to measure the model performance under different values.

IX. Conclusion

In this paper, we presented a comparative analysis of genre classification of traditional Bengali literature based on machine learning. A comprehensive dataset of Bengali classical texts from various genres and authors was compiled and analyzed using three classification models: Multinomial Naive Bayes, LSTM, and Transformer. The performance of the models was compared using metrics like accuracy, precision, and recall. This research emphasized the importance of the automated genre classification system for the preservation and accessibility of Bengali classical literature, enabling researchers to efficiently navigate vast archival materials.

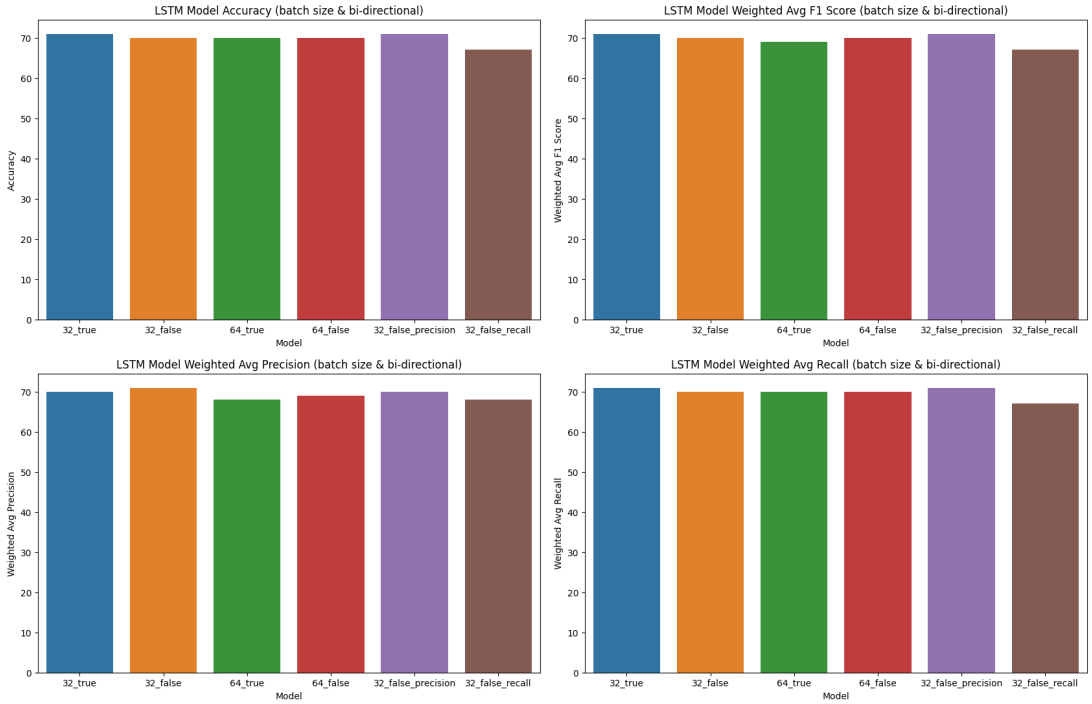


Fig. 6: Accuracy of LSTM model variants

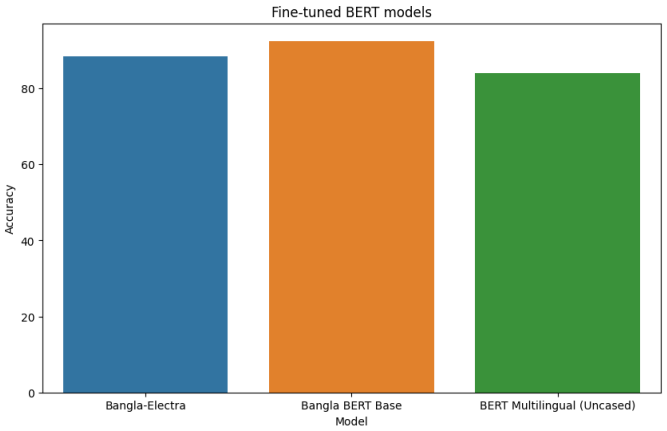


Fig. 7: Transformer model accuracy

REFERENCES

[1] W. A. Yousef, O. M. Ibrahime, T. M. Madbouly, and M. A. Mahmoud, "Learning meters of arabic and english poems with recurrent neural networks: a step forward for language understanding and synthesis," *arXiv preprint arXiv:1905.05700*, 2019.

[2] B. Y. Panchal, "Book genre categorization using machine learning algorithms (k-nearest neighbor, support vector machine and logistic regression) using customized dataset," *Book Genre Categorization Using Machine Learning Algorithms (K-Nearest Neighbor, Support Vector Machine and Logistic Regression) using Customized Dataset*, 2021.

[3] J. Worsham and J. Kalita, "Genre identification and the compositional effect of genre in literature," in *Proceedings of the 27th international conference on computational linguistics*, pp. 1963–1973, 2018.

[4] A. Goyal and V. Prem Prakash, "Statistical and deep learning approaches for literary genre classification," in *Advances in Data and Information Sciences: Proceedings of ICDIS 2021*, pp. 297–305, Springer, 2022.

[5] K. Sarkar and M. Bhowmick, "Sentiment polarity detection in bengali tweets using multinomial naïve bayes and support vector machines," in *2017 IEEE Calcutta Conference (CALCON)*, pp. 31–36, IEEE, 2017.

[6] K. Pal and B. V. Patel, "Data classification with k-fold cross validation and holdout accuracy estimation methods with 5 different machine learning techniques," in *2020 fourth international conference on computing methodologies and communication (ICCMC)*, pp. 83–87, IEEE, 2020.

[7] I. Rasheed, V. Gupta, H. Banka, and C. Kumar, "Urdu text classification: a comparative study using machine learning techniques," in *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, pp. 274–278, IEEE, 2018.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[11] R. Tagore, "Rabindra rachanabali - introduction," 2023. Accessed: August 23, 2023.

[12] A. A. Galib, "Kaggle datasets," 2023. Accessed: August 23, 2023.

[13] Bangla-Electra, "Bangla-electra model homepage," 2023. Accessed: August 23, 2023.

[14] B. B. Base, "Bangla bert base model homepage," 2023. Accessed: August 23, 2023.

[15] B. M. Base, "Bert multilingual base model (uncased) homepage," 2023. Accessed: August 23, 2023.