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

Asadullah Al Galib, Maisha Mostofa Prima, Satabdi Rani Debi, MD Muntasir Mahadi,
Nayema Ahmed, Ehsanur Rahman Rhythm, Adib Muhammad Amit, Annajiat Alim Rasel
Department of Computer Science and Engineering (CSE)
School of Data and Sciences (SDS)
Brac University
66 Mohakhali, Dhaka - 1212, Bangladesh
{asadullah.al.galib, maisha.mostofa.prima, satabdi.rani.debi, md.muntasir.mahadi,
nayema.ahmed, ehsanur.rahman.rhythm, adib.muhammad.amit}@g.bracu.ac.bd, annajiat@gmail.com

***Abstract*—Bengali classical literature, a rich collection of works from poetry to plays, novels, and essays, is of significant cultural and historical importance. However, there is limited research on using Natural Language Processing (NLP) techniques for analyzing these texts compared to other languages. The first objective of this research is to compile a comprehensive dataset of Bengali classical texts, providing valuable resources for developers and researchers interested in exploring different aspects of Bengali classical literature using NLP techniques. The second objective is to apply NLP methods to classify genres, which enables researchers to access specific genres of interest easily.The research will use NLP methods and machine learning algorithms to develop a genre classification system specifically designed for Bengali classical literature, with implications for its preservation and accessibility.**

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

I. Introduction

Bengal's literary history shows how remarkable it has been in terms of traditional Bengali literature, in particular. During that time, prominent writers like Kazi Nazrul Islam, Rabindranath Tagore, Bankimchandra Chattopadhyay, and Sarat Chandra Chattopadhyay had a lasting impact on the cultural environment. To understand the depth of Bengali literature, it requires a comprehensive study on different genres and its characteristics.

Genre classification is important to research because it assists readers to discover and learn about the several characteristics, themes and different components. By analyzing the genres of ancient literature, we can gather knowledge of literary idioms and styles that were popular. By classifying the genres that literary works belong to, we can learn more about the cultural, social, and intellectual climate of the times by examining the traits and subjects of various genres in classical Bengali literature. Each genre presents a distinct viewpoint and highlights the originality, inventiveness, and literary skill of the writers who helped Bengali literature grow.

In this project, we will apply 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 genre of literary works given a short snippet from any of the categories such as essay, novel, poem or story.

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 [15].

The work of Parilkumar Shiroya et al. compare 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 [10].

The work of Joseph Worsham et al. develop 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 [14].

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 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 [7].

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 [12]. 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. It evaluates the chance that a document fits one or more predetermined categories. In [9], 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 [11].

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. Textual data is inherently sequential, where the arrangement of words within a sentence can dramatically alter its semantic significance. 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 [6]. 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.

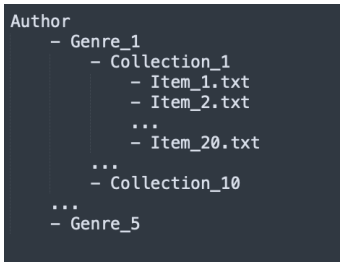


Fig. 1: Content hierarchy

The BERT (Bidirectional Encoder Representations from Transformers) model is one demonstration of these transformers [4]. 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.

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 [13]. We have compiled and published all four datasets on Kaggle, namely Rabindranath, Nazrul, Sarat, and Bankim [5]. 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 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 look 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 text was tokenized using the Natural Language Toolkit (NLTK) tokenizer, and stopwords and digits were removed to retain meaningful words. The preprocessed content was joined to form coherent documents, and each genre label was encoded into numerical values using the Label Encoder. The Term Frequency-Inverse Document Frequency (TF-IDF) vectorization technique was used to convert the textual data into numerical feature vectors. Hyperparameter tuning was performed using Grid Search with cross-validation, considering 'alpha' and 'fit prior' for the MNB classifier. Stratified sampling was used during the train test split to address data imbalance. The MNB classifier was trained on the training data using the best hyperparameters obtained from the grid search.

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 [4]. *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 [1]. *Bangla BERT Base*: This is a pre-trained language model using mask language modeling and trained on the same dataset as Bangla-Electra [2]. *BERT Multilingual Base Model (Uncased)*: This model also used masked language modeling, but was trained on Wikipedia for the top 102 languages [3].

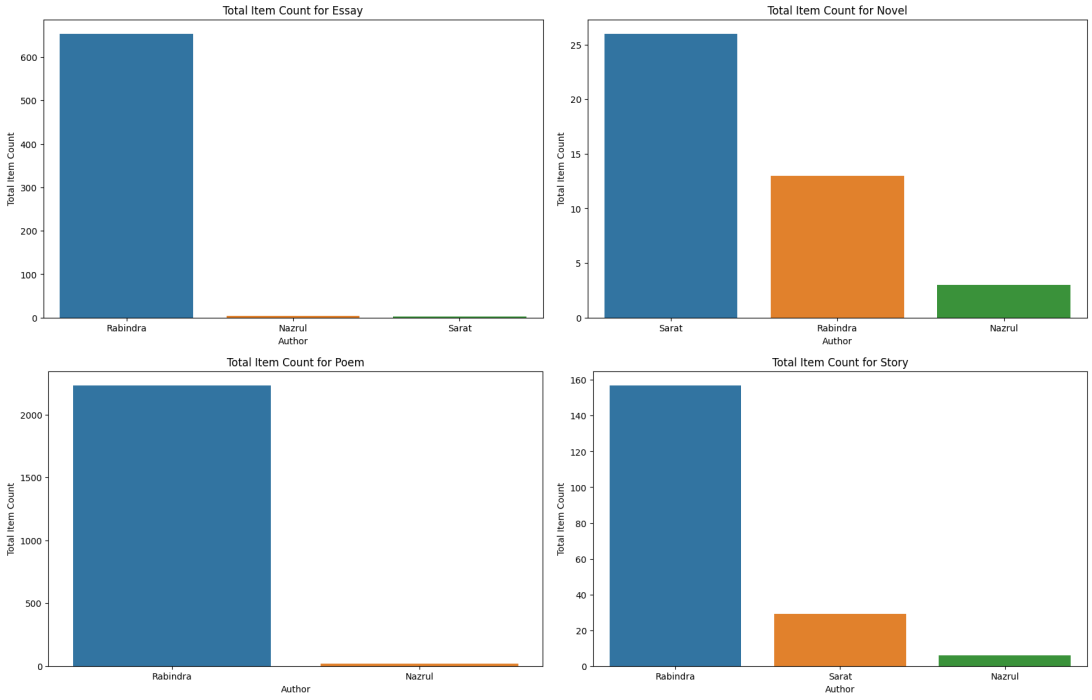


Fig. 2: Total item count by genre

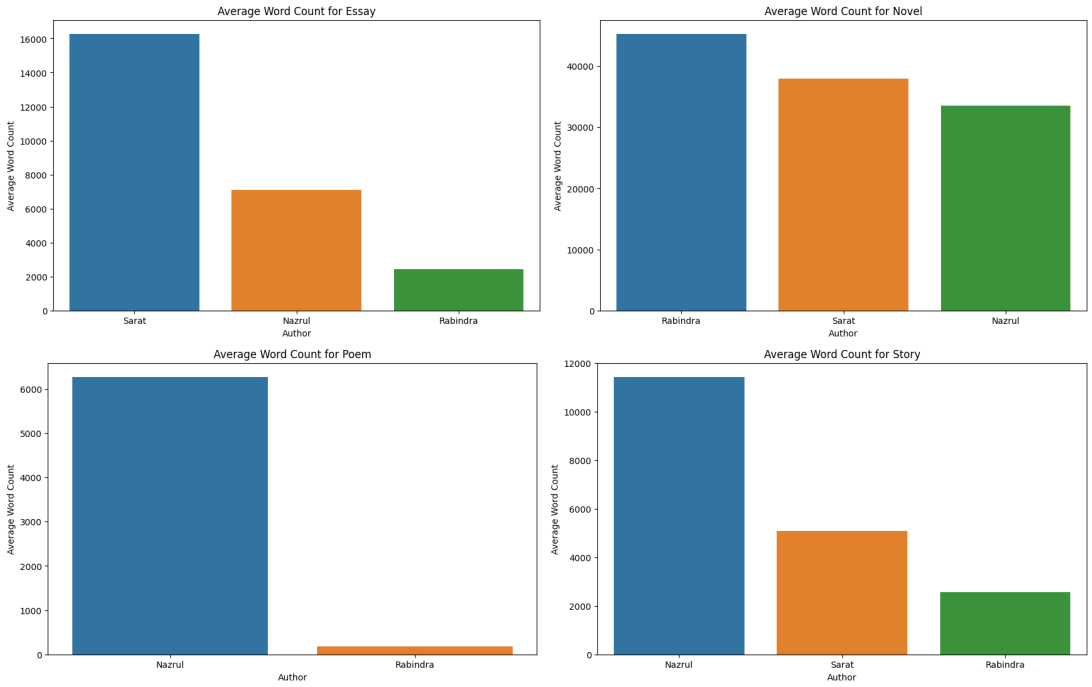
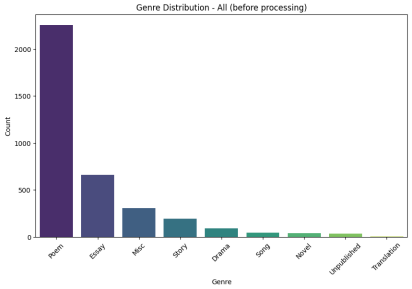
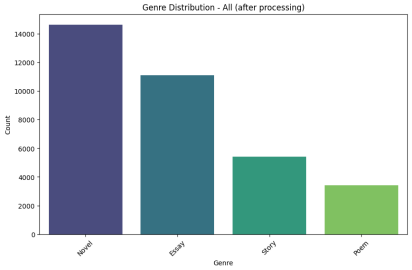


Fig. 3: Average word count by genre



(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

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, class-wise accuracy was computed, and the classification report provided precision, recall, F1-score, and support for each genre. We observed that applying 42 as the random state consistently improved accuracy when using multiple random state values. Notably, our analysis yielded the following insights for each author: For Nazrul, optimal hyperparameters were observed as 'alpha' = 0.01 and 'fit prior' = True, yielding an accuracy of 64.71%. Similarly, for Rabindranath, the best hyperparameters were 'alpha' = 0.01 and 'fit prior' = True, resulting in an accuracy of 87.5%. Correspondingly, Saratchandra demonstrated optimal settings at 'alpha' = 1 and 'fit prior' = True, with an accuracy of 43.48%. Aggregating across authors, the best performing hyperparameters were 'alpha' = 0.01 and 'fit prior' = True, resulting in an accuracy of 84.44%.

B. LSTM & Fine-tuned BERT

We generated performance measures for all the variants of LSTM models under different hyperparameter values. From Figure 5, 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 Model 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 6.

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 were 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.

References

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

[2] Bangla BERT Base. Bangla bert base model homepage, 2023. Accessed: August 23, 2023.

[3] BERT Multilingual Base. Bert multilingual base model (uncased) homepage, 2023. Accessed: August 23, 2023.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Asadullah Al Galib. Kaggle datasets, 2023. Accessed: August 23, 2023.

[6] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

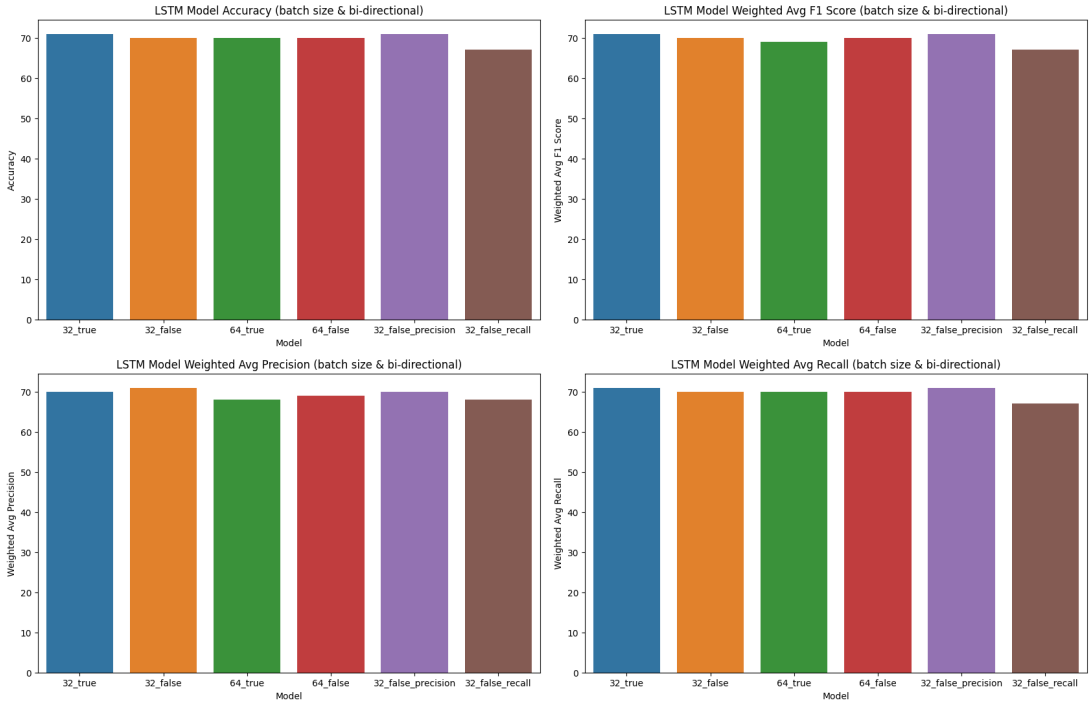


Fig. 5: Accuracy of LSTM model variants

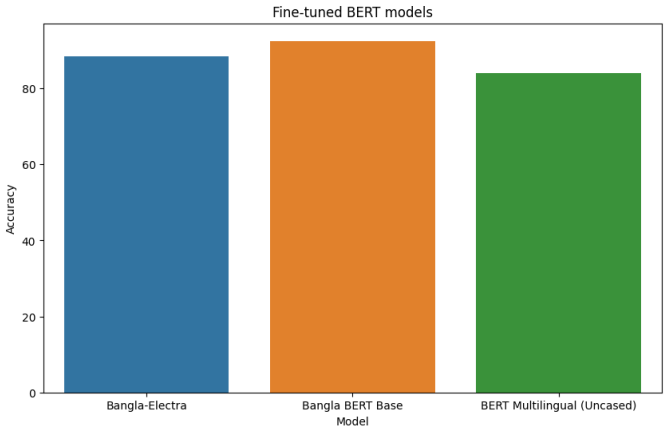


Fig. 6: Transformer model accuracy

[7] Anshaj 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*, pages 297–305. Springer, 2022.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] Kaushika Pal and Biraj 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)*, pages 83–87. IEEE, 2020.

[10] Brijeshkumar 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.

[11] Imran Rasheed, Vivek Gupta, Haider Banka, and Chiranjeev Kumar. Urdu text classification: a comparative study using machine learning techniques. In *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, pages 274–278. IEEE, 2018.

[12] Kamal Sarkar and Mandira Bhowmick. Sentiment polarity detection in bengali tweets using multinomial naïve bayes and support vector machines. In *2017 IEEE Calcutta Conference (CALCON)*, pages 31–36. IEEE, 2017.

[13] Rabindranath Tagore. Rabindra rachanabali - introduction, 2023. Accessed: August 23, 2023.

[14] Joseph Worsham and Jugal Kalita. Genre identification and the compositional effect of genre in literature. In *Proceedings of the 27th international conference on computational linguistics*, pages 1963–1973, 2018.

[15] Waleed A Yousef, Omar M Ibrahime, Taha M Madbouly, and Moustafa 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.