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

***Abstract*—Bengali classical literature holds immense cultural and historical significance, encompassing a rich collection of works ranging from poetry to plays, novels, and essays. However, compared to other languages, there has been limited research and development in leveraging NLP techniques for analyzing Bengali classical texts. By addressing this gap, we aim to unlock the potential of NLP in exploring and understanding this literary heritage. The first objective of this research is to compile a comprehensive dataset of Bengali classical literary texts. This dataset will serve as a valuable resource for researchers and developers interested in exploring various aspects of Bengali classical literature using NLP techniques. The second objective is to apply NLP techniques for genre classification of Bengali classical texts. Genre classification is crucial for organizing and categorizing literary works, enabling researchers to easily access specific genres of interest. By employing machine learning algorithms and NLP methods such as text preprocessing, feature extraction, and classification models, we aim to develop an automated genre classification system specifically tailored to Bengali classical literature. The outcomes of this research have broad implications for the preservation and accessibility of Bengali classical literature. By automating the analysis and categorization of these texts, researchers and enthusiasts can efficiently navigate through vast repositories of archival materials, leading to deeper insights into the historical, cultural, and linguistic aspects of Bengali classical literature.**

***Index Terms*—Bengali Literature, SVM, Naive Bayes**

I. Introduction

Bengal's literary history is seen to have reached its literary zenith during the 19th century, especially in terms of classical Bengali literature. It was a time when the region saw the rise of notable writers like Kazi Nazrul Islam, Rabindranath Tagore, Bankimchandra Chattopadhyay and Sarat Chandra Chattopadhyay who had a lasting impact on the cultural environment. Exploring the many genres of ancient Bengali literature and examining its traits and themes is an important part of knowing and appreciating its richness.

In order to recognize and comprehend the unique qualities, themes, and creative components present in a given body of work, genre classification is essential to literary study. We learn about the many literary styles and idioms that were in vogue at this time by exploring the genres found in classical Bengali literature.

Poetry is an important literary genre in traditional Bengali literature. Bengali poetry comes in a variety of formats, including lyrical pieces and narrative lines. It displays a wide range of style, structure, and content. One can explore the vivid world of rhymes, sonnets, epics, and other literary forms within the field of poetry, which highlight the mastery of language and imagery by well-known poets of the time.

Drama is an important sub-genre of traditional Bengali literature. The existence of dramatic works, such as plays and theatrical productions, gives the literary landscape a vibrant new dimension. Bengali playwrights created engrossing stories that they then brought to life on stage by examining human emotions, social problems, and philosophical ideas. Drama was a genre that offered a stage for social critique, amusement, and contemplation, making it a crucial component of traditional Bengali literature. 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 exploration, we will contrast and evaluate the main literary sub-genres of ancient Bengali literature, emphasizing their unique characteristics, theme investigations, and creative contributions. By doing this, we can fully comprehend the complex nature of traditional Bengali literature and recognize its lasting significance in Bengal's and other countries' literary traditions.

II. Related Work

Waleed A. Yousef et al. [1] 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.

The work of Parilkumar Shiroya et al. [2] 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.

The work of [3] 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.

Anshaj Goyal et al. [4] 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 than among the deep learning models. The RNN approach outperformed all other models in the evaluation.

III. Methodology

The methodology for this paper involves: 1. Dataset analysis to compile a comprehensive collection of Bengali classical literary texts. 2. Building baseline models (SVM and Naive Bayes) for genre classification as performance benchmarks. 3. Training deep learning models (e.g., RNN, Transformer) tailored to the characteristics of Bengali classical literature. 4. Analyzing results by evaluating metrics like accuracy, precision, recall, and using confusion matrices and classification reports. 5. Discussing future directions, including advanced pre-training techniques and exploring related tasks like sentiment analysis or authorship attribution, to advance NLP in Bengali classical literature.

In 1997, two German scientists made the initial LSTM proposal. This architecture solves the issue of exploding and disappearing gradients in vanilla RNNs. The originality of LSTM is found in its memory cells and gating mechanisms, which include input, forget, and output gates. Data retention over lengthy sequences aids NLP and time series prediction. Gating manages information flow by capturing dependencies in sequential data over time. By managing information flow, LSTMs excel at jobs with long-term dependencies, making them useful in dealing with sequential data with complicated patterns and linkages.

The architecture of the Transformer is encoder-decoder. Long-term dependencies are captured by self-attention in both components, which is excellent for NLP tasks. It is essential for machine translation, language modeling, sentiment analysis, and other sequential data applications due to its parallel processing and complicated linkage handling.

Both LSTM and Transformer can be used in text classification tasks, where the goal is to assign a label or category to a given text.

1. LSTM in Text Classification : When modeling long-term dependencies in the input text is necessary, LSTM can be employed successfully in text classification tasks since it is well suited for jobs involving sequential data. Each word or character in the text is turned into word embeddings or character embeddings using this method. These embeddings are then supplied into the LSTM layers, which sequentially process the input sequence, capturing contextual information at each step. The LSTM's final hidden state is utilized to represent the entire input text, which is then sent through a classifier (for example, a fully connected layer) to predict the class label.

2. Transformer in Text Classification: Transformer, on the other hand, excels at coping with long-distance dependencies and parallel processing. The input text is tokenized into subword or word-level tokens in text classification, and each token is turned into word embeddings. The Transformer encoder is then fed these embeddings. The Transformer's self-attention mechanism enables the model to effectively capture associations between different tokens in the input sequence, independent of their placements. The Transformer encoder's output representation is then processed via a classifier to determine the final class prediction.

In practical applications, Transformers have shown remarkable performance gains over LSTMs, especially in large-scale text classification tasks, where they can take advantage of parallel processing and effectively handle long texts. However, LSTMs can still be useful in scenarios where the text sequences are relatively short, and the sequential processing capability is beneficial.

In the paper " Personality Classification Based on Twitter Text Using Naive Bayes, KNN and SVM", two algorithms were used named "Naive Bayes" and "Support Vector Machine (SVM)" for text classification to predict personality traits based on text written by Twitter users.

Naive Bayes: This research paper uses the Naive Bayes Probabilistic Classification Method (Naive Bayes) based on the Bayes' theorem to predict personality traits in text data. This method involves calculating the probability of each personality trait based on the words in the text, and assigning the text to the most likely personality trait. The article uses the Multinomial Neveu (MNB) Naive Bayes variant for this purpose.

Naive Bayes Working: Naive Bayes is a pre-processing technique that prioritizes text data, such as tweets, by tokenizing the words, stalling them, and filtering out stop words. TF-IdF (Term Frequency-Inverse Document Frequency) is calculated for each word within the dataset, which captures the importance of the word in relation to the entire dataset. For each personality trait, a TF-IDF-weighted algorithm is trained on labeled data for each trait.

SVM: Additionally, the text data is classified using the support vector machine (SVM) algorithm, which is a supervised learning technique that works by finding the most suitable hyperplane to divide different classes of text into higher-dimensional spaces. This technique can be used to classify both non-linear and linear text data.

SVM Working: The goal of SVM is to identify a hyperplane that is most suitable for different personality traits, with a maximum margin between the data points. The SVM classifier then assigns the corresponding personality trait to the text based on the side it falls on.

Naive Bayes and Support Vector Machines (SVM) are used to rank the text data according to different personality traits, based on the relationships and patterns between words and traits found in training data.

A. Dataset Compilation

We have compiled datasets for all four authors from the digitized collection of literary works published online by the Department of Information Technology & Electronics, Government of West Bengal, India. The overall content hierarchy for each author is this:

- The landing page contains all the genres available.
- Inside each genre, works are organized under collections and sometimes individual items as well.
- Inside each collection, we have individual items that comprise the collection.
- Finally, each individual item is further broken down into pages.

In order to handle this complexity and dynamic content hierarchy, we designed the crawler in such a way that, given the landing URL for each genre, it can automatically crawl all the collections, items inside the collections, and all the pages for an item. For each item, after crawling and parsing all the pages that belong to that item, we store each item in a separate text file, maintaining the directory hierarchy shown here:

B. Data Pre-processing

After collecting all the contents, we ran each item content that is stored in a text file through a preprocessor that performed basic preprocessing steps such as removing special symbols, page numbers, and repetitive title names. Finally, we aggregated all text files of all collections inside all genres into a single JSON or CSV file to be used for various classification or text generation tasks.

C. Model Training

starts here

IV. Results

results goes here

V. Future Works

future works goes here

VI. Conclusion

conclusion goes here

References

[1] Joseph Worsham and Jugal Kalita. 2018. Genre Identification and the Compositional Effect of Genre in Literature. In Proceedings of the 27th International Conference on Computational Linguistics, pages 1963–1973, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

[2] Goyal, Anshaj, and V. Prem Prakash. Statistical and Deep Learning Approaches for Literary Genre Classification. Advances in Data and Information Sciences, Springer Singapore, 2022, pp. 297–305.

[3] Yousef, Ibrahime, Madbouly, Mahmoud. 2019. Learning meters of Arabic and English poems with Recurrent Neural Networks: a step forward for language understanding and synthesis, 7(6), e1218.

[4] Panchal, 2021. Book Genre Categorization Using Machine Learning Algorithms (K-Nearest Neighbor, Support Vector Machine and Logistic Regression) using Customized Dataset, Sardar Vallabhbhai Patel Institute of Technology.