Book Genre Classification using NLP Techniques

Rishi Kiran Reddy Nareddy nareddy@usc.edu
Koushik Reddy Konda kkonda@usc.edu
Rohith Kumar Kandula vkandula@usc.edu
Peddholla Sai Kumar Reddy peddholl@usc.edu
Kasarla Sai Kumar Reddy kasarla@usc.edu

1 Abstract

In this paper, we will provide an update on the progress we've made in the project thus far.

2 Tasks Performed

2.1 Creating and preprocessing Data

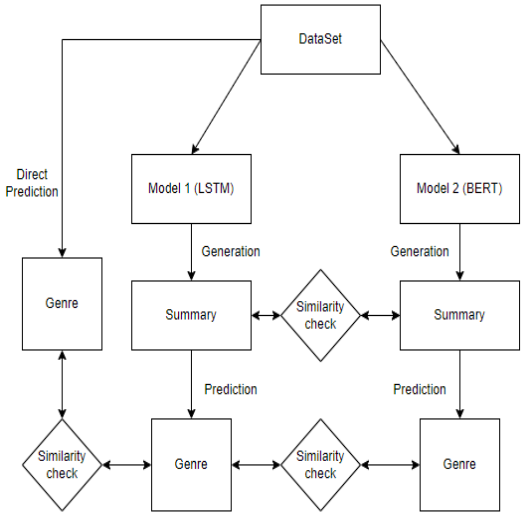
Creating and preprocessing data are pivotal components in the "Book genre classification using NLP techniques" project. Data acquisition involves the collection of a book corpus, encompassing all genres of interest, coupled with their corresponding labels. Subsequently, data preprocessing involves a series of text normalization techniques that transform raw textual data into a format that is suitable for machine learning algorithms. Specifically, the preprocessing phase entails the conversion of all text to lowercase, the elimination of extraneous whitespaces, and the application of lemmatization to reduce morphological variations. These techniques standardize the text data and reduce its complexity, enabling machine learning algorithms to learn patterns and classify books accurately. By performing rigorous data creation and preprocessing, we can ensure that the machine learning models are trained on a high-quality, representative corpus that closely approximates the real-world data. We are further exploring other techniques which probably can improve the data quality and yield better learning.

2.2 Book Genre Prediction using Original Description

In book genre prediction using NLP techniques, the summary is pre-processed by removing stop words, special characters, and converting to lowercase. The TF-IDF vectorizer is then used to transform the summary into numerical vectors for input to a logistic regression classifier. Multi-label binarization converts genre labels to binary format for classification, and model training and validation prevent overfitting. The trained classifier predicts the genre of new books by converting descriptions to numerical vectors, with performance measured by F1-score and accuracy. As of now, our model has achieved an accuracy of 36.53% in predicting book genres based on textual descriptions. We are currently exploring additional methods to enhance the accuracy of our model.

2.3 Summarizing Book description

The book description summarization process employs an extractive summarization technique using Python and the nltk package. The input text is segmented into individual sentences and preprocessed. The remaining words' frequency distribution is then computed, and a score is assigned to each sentence based on the sum of its constituent word frequencies. The top sentences are selected for summary generation based on their scores. This method efficiently generates a brief summary that aids in book selection. Our current summarization model is capable of generating a summary of 200 characters. Nonetheless, we are continuing to investigate methods for enhancing the quality of the generated summaries.



3 Risks and Challenges

3.1 Summarization efficacy.

The main challenge of our project is to decide the optimal length of a summary. The efficacy of a summary we generate will strongly dictate how accurately we predict the genre of the book. Ideally, a summary must contain all the important parts of a book, identifying how the story progresses. The challenge here is the ambiguity in choosing an extractive model versus an abstractive model. While an abstractive model understands the underlying semantics of the story and generates a summary, an extractive model will identify important sentences from a description and will output a summary that is a subset of the story. Right now our approach is using multiple extractive models and determining which one would lead to a better result.

3.2 Pre-processing the summarizing data.

One of the major challenges is removing extraneous information, such as background details and character descriptions, from book summaries. Selecting relevant features, and identifying genre-indicative topics. Imbalanced data can bias the model, requiring oversampling, undersampling, or weighted loss functions. Interpretability is crucial, and we are exploring techniques that can provide interpretable explanations to ensure the model is not reliant on spurious correlations or biases.

3.3 Optimal summary length

Summary length can also be a factor in capturing the essence of a book from its story. For instance, a 200-word plot may be enough but the flip side may be that sometimes it can take a 300 or 400-word summary to understand the plot. Determining a medium ground is the challenge here. Although the length of the summary might seem minuscule in the big picture, sometimes summarization models might add data not present in the plot and skew the results, or smaller summarizations might omit main plot points. Also, shorter lengths might make training our model easy.

3.4 Accuracy of Summarised text

Evaluating the accuracy of summarization is challenging due to the subjective nature of summaries and the complexity of natural language. There are various metrics like Cosine similarity, Average Overlap, and, ROUGE, each with its strengths and weaknesses, and no clear consensus on the best one. Also, human evaluators may differ on the quality of a summary, and designing algorithms that capture the nuances of a text is difficult. Accuracy can vary based on factors such as source text length and complexity, the quality of the language model, and the evaluation metric used.

4 Mitigating Risk

For each of the risks that we mentioned in the previous section, we have a plan to address them.

4.1 Summarization efficacy

Using multiple summarization models such as LSTM and BERT can help mitigate the risk of relying too much on one model. Using two models (extractive and abstractive) can ensure important features are not overlooked. Differences in generated summaries can be analyzed to refine our understanding of the text. Multiple models can identify biases or limitations in a specific model and increase accuracy and robustness.

4.2 Pre-processing the summarization data

There are two ways to go about this, first would be to clean the plot and then feed it to the summarization task, so that those words will not be included in the summary. Another way would be to preprocess the data after summarization. Right now, we think the latter would be the better choice as we think pre-processing after summarization would not only be easy but also would not leave some important essence. Experimentation on both and finding the best result is also an avenue to be explored.

4.3 Accuracy of Summarised text

We are planning to use cosine similarity and average overlap metrics to compare summary accuracy between two texts. Cosine similarity measures the similarity between two summary vectors in high-dimensional space. Using bag-of-words or TF-IDF models, we calculate cosine similarity to determine if the summaries contain similar words and phrases. Average overlap measures the degree of word overlap between two summaries, indicating the content and meaning similarity. Comparing these metrics can help evaluate the accuracy of summaries.

Division of Labor

Rishi Nareddy & Sai Kasarla: Tasks Performed
Sai Peddholla: Risks and Challenges
Koushik Konda & Rohith Kandula: Mitigating Risk.

References

[1] Automated Genre Classification of Books Using Machine Learning and Natural Language Processing
[2] Book Genre Classification Based on Titles with Comparative Machine Learning Algorithms
[3] Book Genre Categorization Using Machine Learning Algorithms (K-Nearest Neighbor, Support Vector Machine and Logistic Regression) using Customized Dataset
[4] Kaggle CMU Book Summary Dataset
[5] Four Minute Book