Introduction to Natural Language Processing (NLP)
23CSAI05H

ID	Name
211392	**Ahmed Sameh**
211896	**Abdulrahman Ayman**
212071	**Sara Anabtawi**
206879	**Ezz Yasser**

Literature Review and Comparative Analysis for Book genre prediction

Ahmed Sameh
Artificial Intellegince major. Senior Student
The British University in Egypt
Cairo, Egypt
ahmed211392@bue.edu.eg

Ahmed Sameh
Artificial Intellegince major. Senior Student
The British University in Egypt
Cairo, Egypt
ahmed211392@bue.edu.eg

Abdulrahman Ayman
Artificial Intellegince major. Senior Student
The British University in Egypt
Cairo, Egypt
abdulrahman211896@bue.edu.eg

Ezz Yasser
Artificial Intellegince major. Senior Student
The British University in Egypt
Cairo, Egypt
Ezzeldein206879@bue.edu.eg

Abstract: This research investigates book genre prediction with deep learning approaches. We use oversampling and under sampling to address class imbalance in a dataset of 4658 records and 10 genres. The text data goes through a pretreatment pipeline that includes cleaning, stop word removal, stemming, label encoding, and tokenization with vectorization. Three different models are tested: CNN with CBOW Word2Vec embeddings, CNN with Skip-Gram Word2Vec embeddings, and LSTM with GloVe embeddings. All models produce promising results, with the CNN-CBOW model having the highest overall accuracy (76.7%) and weighted average F1-score (0.76). This demonstrates the efficacy of deep learning in book genre classification while emphasizing the significance of proper data balancing and text pretreatment procedures.

Keywords: LSTM, CBOW, SKIP-GRAM, KNN, CNN, RNN

I.Introduction

In today's world, due to ever-increasing demand to make computers perform tasks of humans, machine learning is used. It is a tedious task to manually read the entire book and classify it based on its genre. Novice writers find it troublesome to figure out the genre of their book, which can affect its reach to the right audience [1]. Big data technologies have an unbreakable relation to the science of artificial intelligence. Natural language analysis is one of their fields of expertise. Computers are able to identify patterns in texts and classify them based on specified groups. Open-source instruments are capable of recognizing text based on machine learning and preset input data, making this project simple to configure. NLP machine learning systems often use a statistical model to make probabilistic decisions[8]. Deep learning algorithms have also been applied recently, with great success. Input data consists of text fragments, including word sequences, phrases, and full papers. The current study is focused on using the programming language known as Python to predict the genre of a book. This study explores how books are classified based on their summaries. The proposed idea implies that books can be classed based on their written summaries' word content. After training on a dataset, the model by default will categorize new books into certain categories. One of the primary reasons for this is that books are significantly longer than other forms of text media. As a result, we'll be working with book summaries rather than the entire book [2].

Skip gram and CBOW are two common bigram NLP techniques for generating word embeddings. In the past, N-Gram models were substantially slower to train and resulted in less accurate embeddings that used more space. These models addressed issues and paved the path for more advanced models, such as 'GloVe' and 'Infersent', which offer scalability and specialization above traditional models. Skip gram and CBOW are important foundational structures for word embedding in NLP. Studying them can reveal patterns that can be applied generally [3]. GloVe is a popular word embedding technique in Natural Language Processing (NLP). These embeddings are numerical representations of words that convey their semantic meaning and relationships with other words. This enables NLP models to comprehend the complexity of language and execute tasks like:

Machine translation improves the accuracy with which material is translated from one language to another.

Text classification: Sort text into categories

Information retrieval: Locate appropriate papers or sections based on the user's inquiry. [4].

Long short-term memory architectures are complicated and advanced variants of RNN. Long-term neural networks are effective in constructing optimized layers by remembering input over time. Instead of a single gate in a regular RNN, this model features four interconnected gates with distinct functions. The network uses the prior hidden state and current input as input values. A Recurrent Neural Network (RNN) is a machine learning model widely used in NLP jobs. RNNs may grasp sequential information by using the output of the preceding layer as input, unlike typical neural networks that have distinct inputs. This enables them to remember previous calculations, making them effective for tasks like word prediction that rely on prior knowledge. Recurrent Neural Networks (RNNs) consist of interconnected RNN cells. Convolutional Neural Networks (CNN) are effective at identifying local patterns in data. The convolutional neural network architecture is carried out quite well among the experimental architectures, just as certain word pairs or groups of 3, 4 or 5 relate to the genre of the book. CNN discovers local patterns by building feature maps by element-wise multiplication with the kernel and input value slide region. The feature map is generated by summing all values [5].

II. Analysis

1. Models

I. Model 1 (LSTM)

LSTM relies on the concept of cell state, represented by the horizontal line at the top of Figure 1. The line stores the concealed state and is used for certain operations via gated cells. σ symbolizes the sigmoid function. The X in a circle indicates pointwise multiplication, whereas the + denotes point-wise addition. The first sigma gate is the forget gate, which determines whether to retain information or not.[5]

Model:

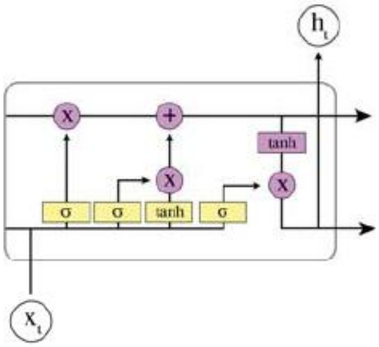


Figure 1.

II. Model 2 (CNN)

Three convolutional layers and three max pooling layers were used after the convolution. A dropout layer was introduced after the SoftMax function to prevent overfitting. Dropout is a regularization strategy that prevents the network from memorizing a certain dataset and allows it to adapt to different inputs. The concept refers to leaving hidden units. Our network was optimized using the Adam optimizer, which allows for faster convergence than other optimizers. We used the categorical cross entropy function as our loss function.[5]

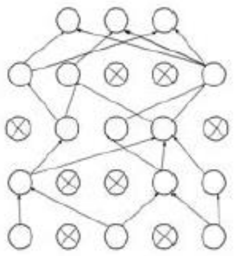


Figure 2.

III. Model 3 (RNN)

In Figure 3, an unfolded RNN is represented by xt, the input at time step t, and st, the hidden layers at time step t, which are calculated based on the previous output and the current step's input. The function st = f(Uxt+Wst−1) gathers information about prior timesteps, with ot representing the output for step t. U, V, and W are the parameters that are shared by all steps.[5]

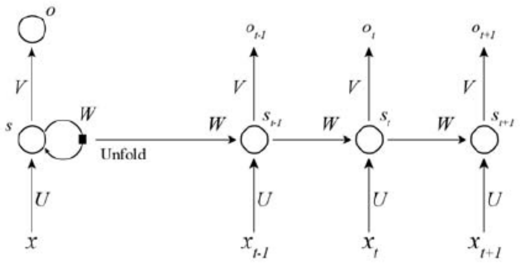


Figure 3.

While RNNs have the potential to detect long-term dependencies, they often fall short in practice. RNNs are limited in their ability to gaze back a few steps due to the vanishing gradient problem. Common activation functions like hyperbolic tangent (tanh) and sigmoid map the real number line to [0, 1], causing this issue. This leads to many inputs being assigned little values. In certain zones, even major input changes result in minor output changes. Stacking multiple layers worsens the issue, leading to an increasingly decreasing gradient.[5]

IV. Model 4 (KNN)

KNN is used to detect patterns in the given dataset, for a particular point belongs to this pattern or category. The algorithm presumes the similarity between the new point and old cases on which machine is already trained and accordingly keep the new point to the nearest most similar category can be used for regression as well. Depending on the K value, distance will be calculated for new datapoint, for example if k=2, then distance will be calculated from nearest two point and new point and then it will be classified into category. [6]

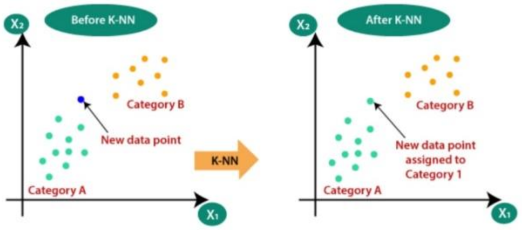


Figure 3.

V. Model 5 (SVM)

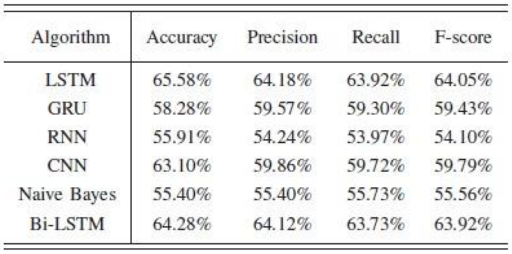
SVM refers to the Support Vector Machine, and it can be used for regression as well as classification. However, it is more used in classification goals. Support vectors are data points near to the hyperplane which affect its direction and position. SVM is useful for controlling both continuous and classified variables.[7]

VI. Model 6 (Logistic Regression)

Logistic regression is both a supervised classification algorithm and a binary classifier. This regression is typically used for splitting data into two classes. polynomial logistic regression can classify data into up to three classes or more. Logistic regression (LR) is a model created using a logistic function. This Logistic function is also known by the name Sigmoid Function.[7]

VII. Experimental Results

According to E. Ozsarfati, C. J. Saul, E. Sahin, and A. Yilmaz Paper [5] The efficiency of different machine learning techniques for classification book genres based solely on titles are investigated. titles can be ambiguous and not always descriptive of genre. So, this paper aims to determine which is the most accurate method for this task. Its approach includes data preprocessing steps such as word embedding creation to allow computer processing of titles. Then evaluating five different machine learning models that have been modified for optimal performance. Particularly, the dataset is unchanged throughout the experiment. As shown below in **Table 1** the results states that Long Short-Term Memory (LSTM) with dropout has the highest accuracy of 65.58% among the tested algorithms.



Algorithm	Accuracy	Precision	Recall	F-score
LSTM	65.58%	64.18%	63.92%	64.05%
GRU	58.28%	59.57%	59.30%	59.43%
RNN	55.91%	54.24%	53.97%	54.10%
CNN	63.10%	59.86%	59.72%	59.79%
Naive Bayes	55.40%	55.40%	55.73%	55.56%
Bi-LSTM	64.28%	64.12%	63.73%	63.92%

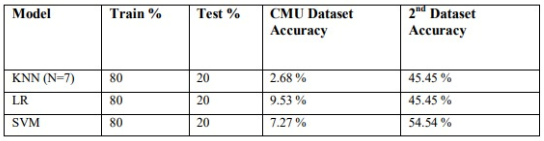
Table 1.

However, According to Verma, Nishtha in Paper [6] after Applying a 40-60 ratio of test and train. The Logistic regression model yields a f1 score of 0.337 and an accuracy score of 0.080, slightly lower than that of a logistic regression model. Similarly, dividing the dataset in a 20-80 ratio of test and train yields a f1 score of 0.57 and an accuracy score of 0.0942 as shown below in **Table 2.**

Model	Train%	Test%	F1-score	Accuracy score
KNN	60	40	0.329	0.0632
	80	20	0.489	0.0932
Logistic Regression	60	40	0.337	0.080
	80	20	0.57	0.0947
LSTM	80	20	0.49	0.09290

Table 2

Moreover, According to P. Shiroya, Darshan Vaghasiya, M. Soni, V. Patel, and B. Y. Panchal in Paper [7] The results indicate that the chosen model performs differently on the two datasets. Where Logistic Regression (LR) achieved the highest accuracy on the first dataset, Support Vector Machine (SVM) outperformed both LR and K-Nearest Neighbors (KNN) on the second dataset as shown below in **Table 3**, stating that it could be due to the Feature Weighting, Data Cleaning, and the Dataset Complexity.



Model	Train %	Test %	CMU Dataset Accuracy	2nd Dataset Accuracy
KNN (N=7)	80	20	2.68 %	45.45 %
LR	80	20	9.53 %	45.45 %
SVM	80	20	7.27 %	54.54 %

Table 3.

VIII. Our Approach

Dataset:

Our dataset is called book genre prediction it contains 4658 records records where it includes 10 classes and 4 features, there was an issue with this datasets and our models as there was an imbalance between the classes. So, this code uses text data from a book dataset to classify. First, the text summaries are converted to numerical vectors using CountVectorizer. The genre information is split as a target variable. To address the issue of underrepresentation, genres such as 'romance', 'psychology','sports', and 'travel' are oversampled, with at least 500 samples each. Genres with more than 500 samples are undersampled to match this figure. To eliminate bias, the oversampled and undersampled data are combined into a final dataset before being shuffled. Now the dataset has 5000 records balancing the classes together.

Preprocessing:

The preparation pipeline cleans and vectorizes text data for genre prediction in books. It includes numerous steps:

1. Text Cleaning: Removes unnecessary characters including backslashes, apostrophes, non-alphabetic characters, and extra whitespace. It also changes the text to lowercase.

2. Stopword Removal: This function uses the NLTK library to eliminate common English stopwords from the text. Additional custom stopwords, including numerals and popular words, are eliminated.

3. Stemming is the practice of reducing words to their base form in order to standardise variants. This stage applies the Porter Stemmer algorithm to each word in the text.

4. Label Encoding: Scikit-learn is used to encode genres by converting categorical labels into numerical representations.

5. Tokenization and Vectorization: Keras is used for tokenization, which separates the text into distinct words or tokens and assigns each token a unique integer index.

1- CBOW(Word2Vec)

We utilised Word2Vec, which uses the CBOW approach to learn distributed representations of words based on their context in the book summary column, to classify books by genre.CBOW attempts to anticipate a target word based on the context words around it. Word2Vec generates dense vector representations for each word by training on tokenized word sequences extracted from book summaries. The word embeddings created by the trained Word2Vec model capture the semantic links between the words in the book summary dataset. The CNN technique may efficiently handle textual input for genre classification since these embeddings represent words in a continuous vector space. Our model architecture classifies the genre using a Convolutional Neural Network (CNN). The architecture's Embedding layer uses pre-trained word embeddings to turn integer-encoded input sequences into dense vectors. Following that, a convolutional layer slides across the input to find genre categorization patterns before extracting local features from a set of word embeddings. In the following dense layer, the collected features are translated to the output classes using activation functions and matrix multiplication. This layer's neurons are mapped to the output classes, and SoftMax activation is used for multi-class classification. A dropout layer is used to reduce overfitting and increase generalization by killing neurons randomly during training. The model's recall, F1-score, and precision all varied. Sports, psychology, and travel genres have good recall and precision scores, indicating that the model can effectively classify instances in these areas. On the other side, genres such as horror and thriller had lower recall and precision ratings, indicating difficulty in correctly classifying these works. Overall, the model has an accuracy of 76.7% and a weighted average F1-score of 0.76.

2- Skip-Gram(Word2vec)

We use pre-trained Word2Vec embeddings to capture word relationships and assist the model in comprehending the text.The model contains a convolutional layer that extracts features from summaries.It employs pooling and dropout techniques to determine the most important attributes while minimising overfitting.Finally, it applies a Softmax layer to create genre predictions.The model is trained with a loss function and an optimizer.We assess its performance on a separate test dataset using accuracy metrics.Using the trained model, a function predicts genres based on summary data.We analyse performance using a random selection from the test set.We achieve a test accuracy of about 70.73% and validate with a bigger sample, resulting in an 83% correct prediction rate.

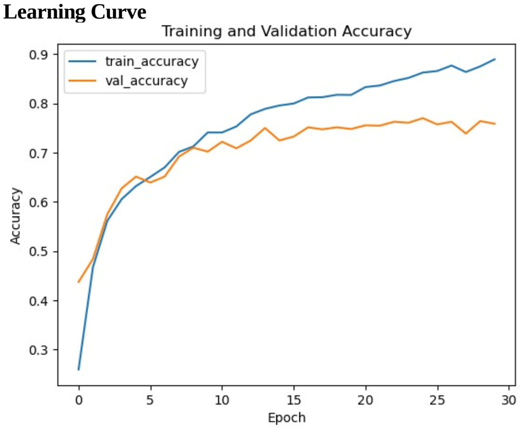
3- Glove

This model is built around pre-trained GloVe word embeddings. It initially loads GloVe embeddings and creates an embedding matrix using the tokenizer's word index. The model architecture consists of three layers: an embedding layer, a dropout layer, and an LSTM layer for sequence processing. The output layer is dense and uses softmax activation. The model is created with categorical cross-entropy loss and the Adam optimizer. Early halting is utilised in training to prevent overfitting. The code is trained on training data before being tested on test data, yielding 77% test accuracy and 88% train accuracy. It generates 431 correct predictions and 69 wrong ones. The classification report shows the precision, recall, and F1-score for each class, with a total accuracy of 77%.

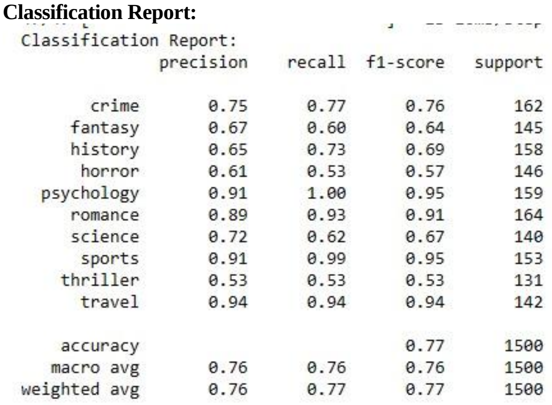
Performance:

Glove:

Learning Curve

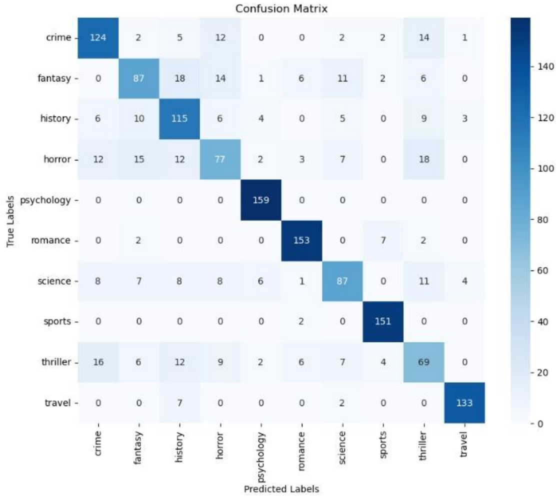


Classification Report:



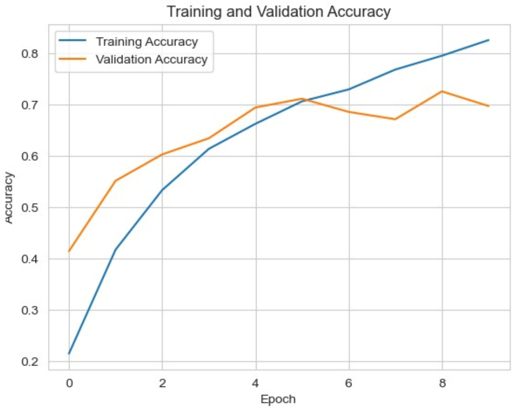
```
Classification Report:
              precision    recall  f1-score   support

       crime       0.75      0.77      0.76       162
     fantasy       0.67      0.60      0.64       145
     history       0.65      0.73      0.69       158
      horror       0.61      0.53      0.57       146
  psychology       0.91      1.00      0.95       159
     romance       0.89      0.93      0.91       164
     science       0.72      0.62      0.67       140
      sports       0.91      0.99      0.95       153
    thriller       0.53      0.53      0.53       131
      travel       0.94      0.94      0.94       142

    accuracy                           0.77      1500
   macro avg       0.76      0.76      0.76      1500
weighted avg       0.76      0.77      0.77      1500
```

Confusion Matrix

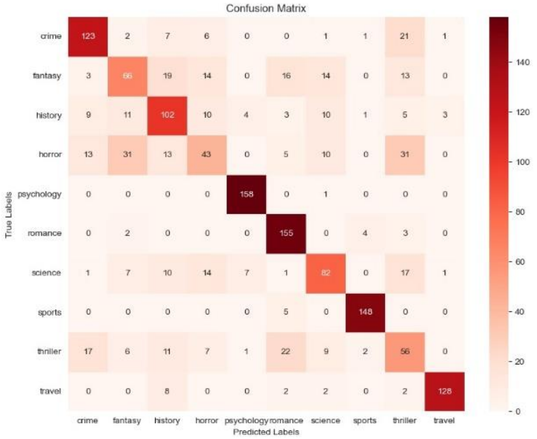


Skipgram:

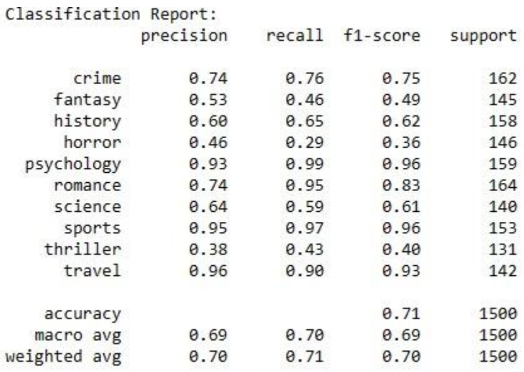
Learning Curve



Confusion Matrix



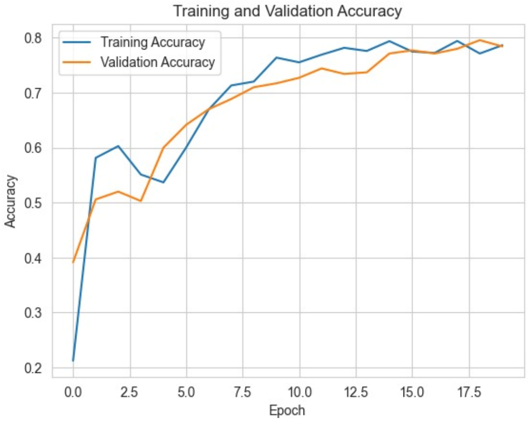
Classification Report:



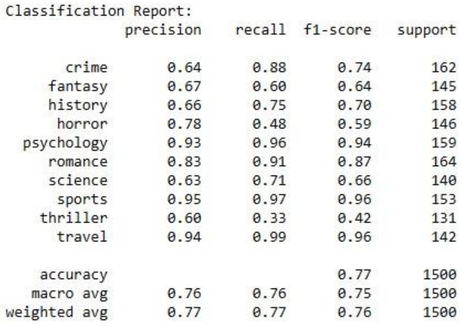
```
Classification Report:
              precision    recall  f1-score   support

       crime       0.74      0.76      0.75       162
     fantasy       0.53      0.46      0.49       145
     history       0.60      0.65      0.62       158
      horror       0.46      0.29      0.36       146
  psychology       0.93      0.99      0.96       159
     romance       0.74      0.95      0.83       164
     science       0.64      0.59      0.61       140
      sports       0.95      0.97      0.96       153
    thriller       0.38      0.43      0.40       131
      travel       0.96      0.90      0.93       142

    accuracy                           0.71      1500
   macro avg       0.69      0.70      0.69      1500
weighted avg       0.70      0.71      0.70      1500
```

CBOW:

Learning Curve

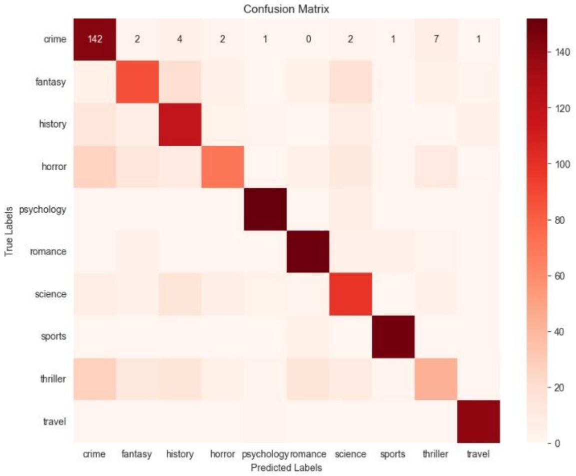


Classification Report:

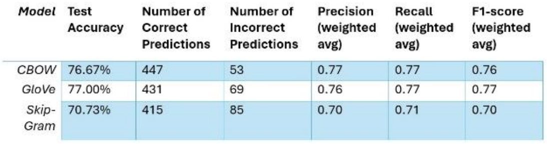


```
Classification Report:
              precision    recall  f1-score   support

       crime       0.64      0.88      0.74       162
     fantasy       0.67      0.60      0.64       145
     history       0.66      0.75      0.70       158
      horror       0.78      0.48      0.59       146
  psychology       0.93      0.96      0.94       159
     romance       0.83      0.91      0.87       164
     science       0.63      0.71      0.66       140
      sports       0.95      0.97      0.96       153
    thriller       0.60      0.33      0.42       131
      travel       0.94      0.99      0.96       142

    accuracy                           0.77      1500
   macro avg       0.76      0.76      0.75      1500
weighted avg       0.77      0.77      0.76      1500
```

Confusion Matrix



Model Comparison Table



Model	Test Accuracy	Number of Correct Predictions	Number of Incorrect Predictions	Precision (weighted avg)	Recall (weighted avg)	F1-score (weighted avg)
CBOW	76.67%	447	53	0.77	0.77	0.76
GloVe	77.00%	431	69	0.76	0.77	0.77
Skip-Gram	70.73%	415	85	0.70	0.71	0.70

III. Conclusion

Our research into book genre prediction using deep learning produced great results. We achieved 76.7% accuracy with a CNN-CBOW model after correcting class imbalance and implementing a complete text preprocessing pipeline. Furthermore, other models using Skip-Gram Word2Vec and GloVe embeddings performed similarly. These studies demonstrate the power of deep learning for automated book genre classification.

References

[1] S. Gupta, M. Agarwal and S. Jain, "Automated Genre Classification of Books Using Machine Learning and Natural Language Processing," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2019, pp. 269-272, doi: 10.1109/CONFLUENCE.2019.8776935.

[2] Desai, P., & Saraiya, M. N. G. (2021). "Book Genre Prediction." International Journal for Research in Applied Science & Engineering Technology (IJRASET), 9(X), pp. 1-10. ISSN: 2321-9653.

[3] P Menon, T.M. (2020). "Empirical Analysis of CBOW and Skip Gram NLP Models." IEEE.

[4] Pennington, J., Socher, R., & Manning, C. (2014). "GloVe: Global Vectors for Word Representation." In A. Moschitti, B. Pang, & W. Daelemans (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1532-1543). Association for Computational Linguistics. Doha, Qatar. doi: 10.3115/v1/D14-1162. [Online]. Available: https://aclanthology.org/D14-1162

[5] E. Ozsarfati, E. Sahin, C. J. Saul and A. Yilmaz, "Book Genre Classification Based on Titles with Comparative Machine Learning Algorithms," 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, 2019, pp. 14-20, doi: 10.1109/CCOMS.2019.8821643.

[6] Verma, N., "Book Genre Prediction Using LSTM," Doctoral dissertation, 2022.

[7] Miljković, D., 2017, May. Brief review of self-organizing maps. In 2017 40th international convention on information and communication technology, electronics and microelectronics (MIPRO) (pp. 1061-1066). IEEE.

[8] Wang, Danrui & Tan, Bowen & Wei, Muchen & Cui, Xuhao & Huang, Xingru. (2023). Using natural language processing and machine learning algorithm for book categorization. Applied and Computational Engineering. 2. 856-867. 10.54254/2755-2721/2/20220551.