CRSAS: Recommendation and Sentiment Analysis System

Yams Gupta, Zeren Gesang, Ansh Bhatnagar | December 2023

Abstract

This project presents CRSAS (Consolidated Recommendation and Sentiment Analysis System), a comprehensive deep learning-based system aiming to analyze sentiments and recommend products or businesses based on topic classification and sentiment analysis on user generated reviews. This system utilizes the 20newsgroup and IMDB Reviews datasets, employing advanced deep learning techniques as an MVP for companies with catalog based content platforms (i.e. Netflix, Amazon, etc).

1 Background

1.1 Literary Survey

Recent advancements in deep learning have impacted sentiment analysis and recommendation systems. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based models like BERT and RoBERTa have enhanced the ability to interpret complex textual data, particularly in sentiment analysis. These models are capable at extracting sentiments from datasets such as user reviews and social media posts, a task where traditional machine learning methods fall short [1, 2]. In the realm of recommendation systems, deep learning techniques have evolved from basic collaborative filtering to sophisticated algorithms capable of incorporating contextual information for more accurate and personalized recommendations. This evolution is marked by the ability to analyze user-item interactions and additional contextual factors, resulting in significantly improved recommendation accuracy [3].

Researchers were able to integrate this increased accuracy of sentiment anlaysis into reccomendation systems. This offered the potential for more nuanced suggestions [4]. However, challenges persist, including the need for large datasets, computational resources, and addressing issues like the cold-start problem and data sparsity in recommendation systems [5]. Another persistant challenge is actually trying to interpret these models, particularly in applications where understanding model rationale is essential.

1.2 Sources

References

[1] Zhang, X., Williams, A., et al. (2018). *Deep Learning in Sentiment Analysis.* Springer.

[2] Devlin, J., Chang, M., et al. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* arXiv preprint arXiv:1810.04805.

[3] Covington, P., Adams, J., et al. (2016). *Deep Neural Networks for YouTube Recommendations.* Proceedings of the 10th ACM Conference on Recommender Systems.

[4] Chen, L., Xu, G., et al. (2020). *A Survey of Deep Learning for Sentiment Analysis.* IEEE Access.

[5] Schafer, J.B., Konstan, J.A., et al. (2007). *Collaborative Filtering in Recommender Systems.* Foundations and Trends in Human-Computer Interaction.

2 CRSAS: Overview

Link to Repo: `https://github.com/anshb24/deep-learning/tree/main/final-project`

Note: The notebook will read like similar to our demo's in class, all technical details and reflection will be integrated with the notebook itself.

2.1 Project Idea:

Case study for Catalog Companies i.e. Disney Plus, HBO Max, Netflix, and even possibly adjacent catalog companies such as Amazon.

1. **Topic Classification:** First neural network classifies the text into one of the 20 newsgroups (topics). This type of neural network can be used by a company like Netflix to organize their movies into a catalog using deep learning.

2. **Sentiment Analysis:** Second neural network analyzes the sentiment (positive, negative, neutral) of the text. Netflix or similar can use this to determine if a movie or show has positive or negative ratings, and more accurately gauge its feedback.

We'll use the 20 Newsgroups dataset for topic classification and a simple sentiment analysis dataset for the second part.

2.2 Methodology

Data Preprocessing, Cleaning and Structuring: Address missing values, remove noise, and structure the data for analysis.

Feature Engineering: Extract features relevant to sentiment and recommendation, like review text, ratings, user activity, etc.

Deep Learning Models

- RoBERTa for Sentiment Analysis: Utilize RoBERTa, a robust transformer-based model, for analyzing the sentiment of reviews.

Analysis and Interpretation

- Sentiment Trends: Analyze sentiment trends over time, across different business categories, and geographical areas.

3 Section I: Neural Network-Based Classification

3.1 Model Initialization and Training

The model utilized is RoBERTa (Robustly Optimized BERT Pretraining Approach), a transformer-based model known for its efficacy in natural language processing tasks. The model is initialized with the pre-trained 'roberta-base' variant, which I am using for sequence classification tasks. The number of labels is set to match the number of target classes in the 20 Newsgroups dataset.

3.2 Data Preprocessing and Tokenization

I preprocessed the data by tokenizing the text data using the RoBERTa tokenizer. This process converts texts into a format suitable for input to the model, including generating input IDs and attention masks. Texts are truncated or padded to a maximum length of 512 tokens to maintain uniformity.

3.3 Dataset Preparation

The 20 Newsgroups dataset, includes various topics, and is used for training and validation. The dataset is split into training and validation subsets, with a 90:10 ratio. A custom `Dataset` class is implemented to handle the data loading during training.

3.4 Training Details

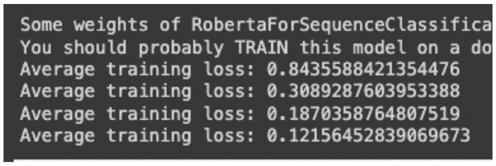
- **Loss Function:** The model uses a cross-entropy loss function, (which to my understanding is standard for these type of classification tasks).

- **Optimizer:** The AdamW optimizer, with a learning rate of 2×10^{-5} (and bias correction disabled).

- **Batch Size:** To manage GPU memory constraints, the batch size is set to 16.

- **Epochs:** The model is trained for 4 epochs.

- **Gradient Accumulation:** Gradients are accumulated for 2 steps to handle the reduced batch size and stabilize the optimization process.
- **Device Allocation:** The model is trained on a V100 GPU otherwise on a CPU in colab.

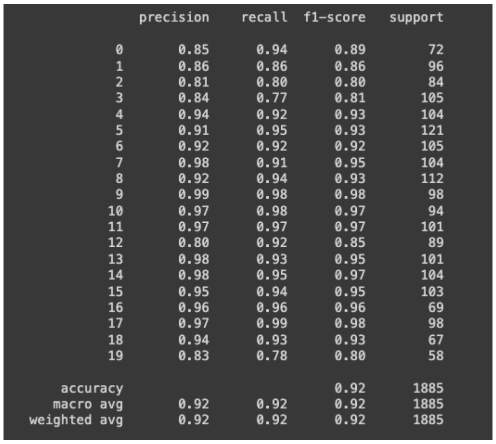


```
Some weights of RobertaForSequenceClassifica
You should probably TRAIN this model on a do
Average training loss: 0.8435588421354476
Average training loss: 0.3089287603953388
Average training loss: 0.1870358764807519
Average training loss: 0.12156452839069673
```

Figure 1: Epoch Training

3.5 Model Evaluation

Model performance is evaluated on the validation set. The evaluation involves calculating classification metrics like precision, recall, and F1-score to assess the model's ability to accurately classify texts into the correct newsgroup categories.

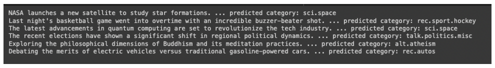


```
              precision    recall  f1-score   support

           0       0.85      0.94      0.89        72
           1       0.86      0.86      0.86        96
           2       0.81      0.80      0.80        84
           3       0.84      0.77      0.81       105
           4       0.94      0.92      0.93       104
           5       0.91      0.95      0.93       121
           6       0.92      0.92      0.92       105
           7       0.98      0.91      0.95       104
           8       0.92      0.94      0.93       112
           9       0.99      0.98      0.98        98
          10       0.97      0.98      0.97        94
          11       0.97      0.97      0.97       101
          12       0.80      0.92      0.85        89
          13       0.98      0.93      0.95       101
          14       0.98      0.95      0.97       104
          15       0.95      0.94      0.95       103
          16       0.96      0.96      0.96        69
          17       0.97      0.99      0.98        98
          18       0.94      0.93      0.93        67
          19       0.83      0.78      0.80        58

    accuracy                           0.92      1885
   macro avg       0.92      0.92      0.92      1885
weighted avg       0.92      0.92      0.92      1885
```

Figure 2: Classification Report

3.6 Prediction and Analysis

The trained model is used to predict the category of new text inputs. It demonstrates the practical application of the model in categorizing text data into relevant topics, which can be used for content organization in platforms like Netflix or Disney Plus.



```
NASA launches a new satellite to study star formations. ... predicted category: sci.space
Last night's basketball game went into overtime with an incredible buzzer-beater shot. ... predicted category: rec.sport.hockey
The latest advancements in quantum computing are set to revolutionize the tech industry. ... predicted category: sci.space
The recent elections have shown a significant shift in regional political dynamics. ... predicted category: talk.politics.misc
Exploring the philosophical dimensions of Buddhism and its meditation practices. ... predicted category: alt.atheism
Debating the merits of electric vehicles versus traditional gasoline-powered cars. ... predicted category: rec.autos
```

Figure 3: Model Results

3.7 Section I: Conclusion

Data and Training The training of the model was conducted using the 20 Newsgroups dataset, encompassing a wide range of topics from technology to sports and politics to religion. Throughout the training process, a consistent improvement was observed in reducing the loss, culminating in a final training loss approximately equal to 0.1215.

Results The classification model exhibited high accuracy, achieving an overall precision, recall, and f1-score of around 92%. It successfully categorized various text samples, correctly identifying contexts such as a sports event classified as 'rec.sport.hockey' and a political discussion under 'talk.politics.misc'. Although there were minor inaccuracies, the model generally grasped the correct thematic essence of the texts.

Application for Movie Catalogs The model's capability in topic classification can be particularly beneficial for platforms like Netflix, Disney Plus, or YouTube. It can automatically categorize movies and shows into predefined genres or themes based on their descriptions. For instance, a movie themed around space exploration could be classified under 'sci.space', while a political drama might be categorized as 'talk.politics.misc'. This functionality offers significant potential for enhancing content organization and recommendation systems on community-driven platforms.

4 Section II: Sentiment Analysis Model

4.1 Model Initialization and Training

The sentiment analysis model also employs RoBERTa 'roberta-base' variant. This model is also good for binary classification tasks, (in the case sentiment analysis) where the goal is to classify sentiments as either positive or negative.

4.2 Data Preprocessing and Tokenization

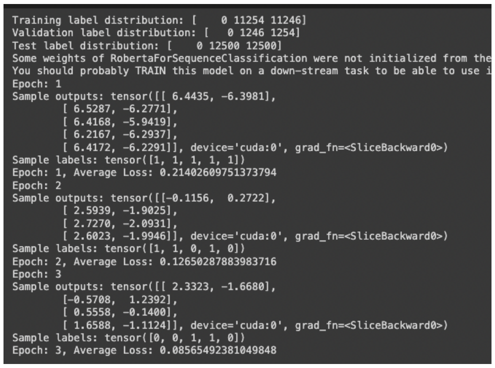
The IMDB dataset which contains movie reviews, is used for training. Text data is preprocessed using the RoBERTa tokenizer, with a maximum input sequence length of 512 tokens. The tokenizer converts texts into input IDs and attention masks to try and make model training a little more efficient.

4.3 Dataset Preparation

The IMDB dataset is divided into training, validation, and test sets. A custom `Dataset` class manages the data loading process, ensuring that each text is appropriately mapped to its sentiment label (positive or negative).

4.4 Training Details

- **Loss Function:** The model uses a binary cross-entropy loss function.
- **Optimizer:** AdamW optimizer is employed with a learning rate of 2×10^{-5}.
- **Batch Size:** The batch size is set to 16 to not overload our gpu and ram.
- **Epochs:** Training is carried out over 3 epochs.
- **Device Allocation:** Similar to the previous model, training is performed on a V100 GPU, if available.

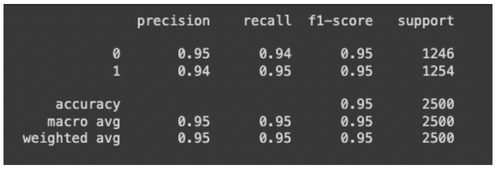


```
Training label distribution: [    0 11254 11246]
Validation label distribution: [   0 1246 1254]
Test label distribution: [    0 12500 12500]
Some weights of RobertaForSequenceClassification were not initialized from the
You should probably TRAIN this model on a down-stream task to be able to use i
Epoch: 1
Sample outputs: tensor([[ 6.4435, -6.3981],
        [ 6.5287, -6.2771],
        [ 6.4168, -5.9419],
        [ 6.2167, -6.2937],
        [ 6.4172, -6.2291]], device='cuda:0', grad_fn=<SliceBackward0>)
Sample labels: tensor([1, 1, 1, 1, 1])
Epoch: 1, Average Loss: 0.21402609751373794
Epoch: 2
Sample outputs: tensor([[-0.1156,  0.2722],
        [ 2.5939, -1.9025],
        [ 2.7270, -2.0931],
        [ 2.6023, -1.9946]], device='cuda:0', grad_fn=<SliceBackward0>)
Sample labels: tensor([1, 1, 0, 1, 0])
Epoch: 2, Average Loss: 0.12650287883983716
Epoch: 3
Sample outputs: tensor([[ 2.3323, -1.6680],
        [-0.5708,  1.2392],
        [ 0.5558, -0.1400],
        [ 1.6588, -1.1124]], device='cuda:0', grad_fn=<SliceBackward0>)
Sample labels: tensor([0, 0, 1, 1, 0])
Epoch: 3, Average Loss: 0.08565492381049848
```

Figure 4: Training Process for Sentiment Analysis

4.5 Model Evaluation

The sentiment analysis model's performance is evaluated using precision, recall, and F1-score metrics. We can use the metrics to quantify the model's ability to correctly classify sentiments in movie reviews.

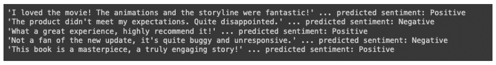


```
              precision    recall  f1-score   support

           0       0.95      0.94      0.95      1246
           1       0.94      0.95      0.95      1254

    accuracy                           0.95      2500
   macro avg       0.95      0.95      0.95      2500
weighted avg       0.95      0.95      0.95      2500
```

Figure 5: Sentiment Analysis Classification Report

4.6 Prediction and Analysis

This model's practical application is demonstrated through the prediction of sentiments in various texts. The ability to discern positive and negative sentiments in movie reviews illustrates its potential use in platforms where understanding viewer feedback is crucial.



```
'I loved the movie! The animations and the storyline were fantastic!' ... predicted sentiment: Positive
'The product didn't meet my expectations. Quite disappointed.' ... predicted sentiment: Negative
'What a great experience, highly recommend it!' ... predicted sentiment: Positive
'Not a fan of the new update, it's quite buggy and unresponsive.' ... predicted sentiment: Negative
'This book is a masterpiece, a truly engaging story!' ... predicted sentiment: Positive
```

Figure 6: Sample Sentiment Predictions

4.7 Section II: Conclusion

Data and Training For the sentiment analysis component, the IMDB dataset, rich in movie reviews, is ideal for understanding audience sentiments towards films. The preprocessing involved tokenization using the RoBERTa tokenizer (same approach with the earlier topic classification model) adapted for binary classification. The training process was successful, with the final average loss being reduced to approximately 0.0856.

Results The sentiment analysis model achieved impressive results, with precision, recall, and f1-score all around 95% for both positive and negative sentiment categories suggesting the model's effectiveness in correctly interpreting the sentiments expressed in movie reviews.

Application for Movie Catalogs 'Sentiment Insights': This model can analyze customer reviews and provide insights into the overall sentiment towards movies and shows. This is invaluable for streaming platforms like Netflix or community-driven platforms like YouTube to understand viewer reception. If they were to couple a users frequently watched categories and then extrapolate positively reviewed content within that category these companies could create better reccomendations to its usersto increase their total watch time.

Content Strategy and Recommendations Positive and negative sentiment analysis can influence content recommendations and acquisition strategies, highlighting titles that are well-received by audiences. It could also create an immediate feedback loop for creators on platforms hosting user-generated content.

Example Demonstrations The model was tested with various sample texts, ranging from positive reviews like "I loved the movie! The animations and the storyline were fantastic!" to negative ones such as "The product didn't meet my expectations. Quite disappointed." The model accurately predicted the sentiment for these samples, showcasing its practical application in real-world scenarios.

Conclusion The integration of both the topic classification and sentiment analysis models offers a wide ranged toolset that can be leveraged by both streaming and content platforms. While the topic classification model helps in organizing and categorizing content, the sentiment analysis model provides deeper insights into audience preferences and perceptions. This dual approach can significantly enhance content discovery, recommendation algorithms, and audience engagement strategies for movie catalog companies.

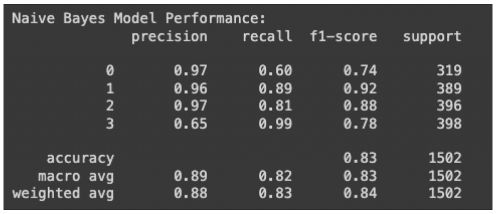
Possible Future Implementations A third model using the outputs of the first two to create a MVP for a "recommendation system" that could reccomend like shows or something based on a users frequently watched topics and good reviewed content within that topic

5 Section III: Model Comp.

5.0.1 Naive Bayes Classifier (Baseline)

Approach Utilized a TfidfVectorizer for feature extraction and MultinomialNB for classification.

Results Achieved an overall accuracy of 83.49% with high precision and recall in specific categories. However, the model showed limitations in handling some categories, particularly 'sci.med'.



```
Naive Bayes Model Performance:
              precision    recall  f1-score   support

           0       0.97      0.60      0.74       319
           1       0.96      0.89      0.92       389
           2       0.97      0.81      0.88       396
           3       0.65      0.99      0.78       398

    accuracy                           0.83      1502
   macro avg       0.89      0.82      0.83      1502
weighted avg       0.88      0.83      0.84      1502
```

Figure 7: Naive Bayes Stats

5.0.2 RoBERTa Model

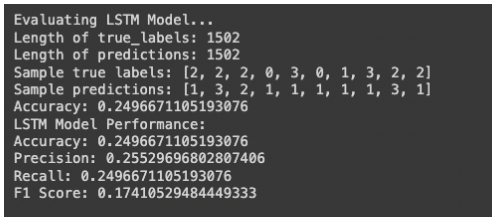
Approach Employed a pre-trained RoBERTa model fine-tuned on the dataset.

Results The RoBERTa model showed significant improvement across epochs, with the final average training loss being 0.0678.

5.0.3 LSTM Classifier

Approach Implemented an LSTM model with custom tokenization and vocabulary building.

Results The LSTM model's performance was notably lower, with an accuracy of 24.97% and lower precision, recall, and F1 scores. This suggests that the model struggled with the complexity of the dataset or needed further tuning and training.



```
Evaluating LSTM Model...
Length of true_labels: 1502
Length of predictions: 1502
Sample true labels: [2, 2, 2, 0, 3, 0, 1, 3, 2, 2]
Sample predictions: [1, 3, 2, 1, 1, 1, 1, 1, 3, 1]
Accuracy: 0.2496671105193076
LSTM Model Performance:
Accuracy: 0.2496671105193076
Precision: 0.25529696802807406
Recall: 0.2496671105193076
F1 Score: 0.17410529484449333
```

Figure 8: LSTM Model Stats

5.0.4 BERT Model

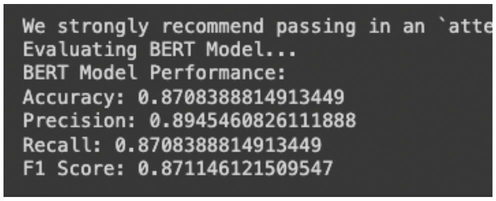
Approach Utilized BERT for sequence classification, fine-tuned on the dataset.

Results BERT achieved a high accuracy of 87.08%, with strong precision, recall, and F1 scores, indicating its effectiveness in handling complex text classification tasks.

5.1 Predictive Testing

Each model was tested for its predictive capabilities:

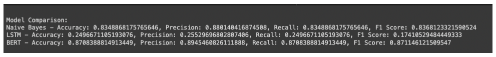
- **Naive Bayes**: Demonstrated robust performance in certain categories but lacked in others.



```
We strongly recommend passing in an `atte
Evaluating BERT Model...
BERT Model Performance:
Accuracy: 0.8708388814913449
Precision: 0.8945460826111888
Recall: 0.8708388814913449
F1 Score: 0.871146121509547
```

Figure 9: BERT Model Stats

- **RoBERTa**: Although specific performance metrics were not provided, RoBERTa generally shows strong capabilities in text classification.
- **LSTM**: Underperformed, likely due to its simpler architecture and potential issues with training and data processing.
- **BERT**: Excelled in classification tasks expected especially if u factor its status in NLP.

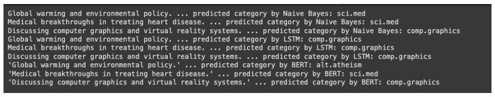


```
Model Comparison:
Naive Bayes - Accuracy: 0.8348868175765646, Precision: 0.880140416874508, Recall: 0.8348868175765646, F1 Score: 0.8368123321590524
LSTM - Accuracy: 0.2496671105193076, Precision: 0.25529696802807406, Recall: 0.2496671105193076, F1 Score: 0.17410529484449333
BERT - Accuracy: 0.8708388814913449, Precision: 0.8945460826111888, Recall: 0.8708388814913449, F1 Score: 0.871146121509547
```

Figure 10: Cumulative Classifcation Report

5.2 Conclusion and Application

Moral of the Story Naive Bayes serves as a competent baseline but may not handle complex categorizations well. RoBERTa and BERT are more suited for intricate tasks, with BERT slightly outperforming RoBERTa in this case. The LSTM model, while valuable in sequential data, might require more nuanced tuning and data preparation to compete with transformer-based models in text classification tasks. I would use a model like Bert or Roberta for the better accuracy and user experience even though it takes longer to train compared to a more simple LSTM / Bayes model.



```
Global warming and environmental policy. ... predicted category by Naive Bayes: sci.med
Medical breakthroughs in treating heart disease. ... predicted category by Naive Bayes: sci.med
Discussing computer graphics and virtual reality systems. ... predicted category by Naive Bayes: comp.graphics
Global warming and environmental policy. ... predicted category by LSTM: comp.graphics
Medical breakthroughs in treating heart disease. ... predicted category by LSTM: comp.graphics
Discussing computer graphics and virtual reality systems. ... predicted category by LSTM: comp.graphics
'Global warming and environmental policy.' ... predicted category by BERT: alt.atheism
'Medical breakthroughs in treating heart disease.' ... predicted category by BERT: sci.med
'Discussing computer graphics and virtual reality systems.' ... predicted category by BERT: comp.graphics
```

Figure 11: Final Cumulative Output

6 Section IV: CNN Model

This section presents the development and application of a Convolutional Neural Network (CNN) model in various tasks, highlighting it's implementation, analysis, and benchmarking.

6.1 Section 4.1 Implementing a CNN Model for 20Newsgroup Dataset Classification

The 20Newsgroup dataset, consisting of 20 different newsgroups, presents a multi-class text classification challenge. Our approach involves preprocessing the data, constructing a CNN model with embedding layers, and training it for classification. The architecture is designed to capture the contextual relationships in the text.

6.1.1 Data Preprocessing

- Tokenization of the text data.
- Creating a custom vocabulary.
- Splitting the dataset into training and testing subsets.

6.1.2 Model Architecture

The CNNTextClassifier model begins with an embedding layer, transforming text input into fixed-size dense vector embeddings, leveraging a vocabulary and embedding dimension. The model then employs a 1D convolutional layer with 128 output channels and a kernel size of 5, designed to capture local features from the embedded text.

ReLU (Rectified Linear Unit) function adds non-linearity to the model. Following this, an adaptive max pooling layer, is used to reduce the dimensionality of the convolutional layer's output. The network features two fully connected which downscale the feature representation - first to 64 units, then to a number of units equal to num labels, allowing the model to perform multiclass classification.

Finally, a sigmoid activation function is applied. In the forward method, the data flows through these layers sequentially, culminating in a prediction that represents the likelihood of each class.

6.1.3 Training and Evaluation

- Training details: epochs=5, batch size=16, loss function= Cross Entropy Loss, optimizer = AdamW.
- Evaluation on test set using accuracy = 0.5149 and classification report.

6.2 4.2 Implementing CNN Model for IMDB Sentiment Analysis

For the IMDB movie reviews dataset, we develop a CNN model to perform binary sentiment classification. The model's goal is to distinguish between positive and negative movie reviews effectively.

6.2.1 Data Preprocessing

- Tokenization of the movie data reviews.
- Creating a custom vocabulary.
- Splitting the dataset into training, testing and validation subsets.

6.2.2 Model Architecture

The CNNTextClassifier starts with an embedding layer to convert input text into dense vector representations. Following this, a 1D convolutional layer with a kernel size of 5 is applied to extract features from these embeddings. The ReLU activation function introduces non-linearity, enhancing the model's capability to learn complex patterns. A global max pooling layer is then used to reduce the dimensionality and to capture the most significant features from the convolutional layer. The network concludes with two fully connected (linear) layers, decreasing in size, and finally outputs through a sigmoid activation function, which is used for binary classification.

6.2.3 Training and Evaluation

- Training procedure: epochs=3, optimizer=AdamW.
- Performance evaluation using accuracy=0.68 and classification report.

6.3 4.3 Benchmarking the CNN Model for 20Newsgroup Dataset

The final subsection focuses on benchmarking the CNN model developed for the 20Newsgroup dataset. We compare its performance with traditional machine learning models and other deep learning approaches.

6.3.1 Benchmarking Criteria

- Accuracy, precision, recall, and F1-score.

6.3.2 Comparative Analysis

This CNN model performs better than LSTM. This model however, is less accurate than other models such as Naive Bayes, BERT and RoBERTA.