CSE440: Natural Language Processing II

Project Report
Online Sexism Detection (EDOS)

by

A.S.M Zawadul Karim - 23241072

Ayesha Bintee Rob - 23241079

Qurratul Ayen Elma - 20201121

Rahnuma Rued - 21301264

Section - 01

Supervised by Dr. Farig Yousuf Sadeque

Submitted May 9, 2024

1 Introduction

In today's tech-driven era, combating misogyny across all spheres, including online platforms and professional environments, is paramount for fostering inclusivity and tackling discrimination. To address this challenge, innovative solutions are imperative, leading to the adoption of advanced machine learning techniques for automated identification and classification of misogynistic language. By harnessing the power of Recurrent Neural Networks (RNNs) with Bidirectional Long Short-Term Memory (Bi-LSTM) architecture, this project introduces a pioneering approach to detect misogyny. RNNs, particularly LSTM networks, excel in processing sequential data like text, owing to their ability to capture temporal dependencies. Bidirectional processing enhances contextual understanding by analyzing text in both forward and reverse directions, enabling the precise identification of linguistic patterns and nuances. The project encompasses two main tasks: Task A involves identifying sexism instances in text, while Task B entails classifying detected sexism into distinct categories. By leveraging machine learning methods and NLP models, the aim is to develop a system adept at detecting and categorizing misogyny occurrences in textual data.

2 Dataset

Sourced from SemEval2023 Task 10, the dataset utilized in this project contains text alongside labels denoting sexism and the sexism category. In Task A, the initial dataset is entirely maintained, containing both sexism and non-sexism data. In contrast, non-sexist data were excluded from Task B to concentrate exclusively on detecting sexism categories. Furthermore, to mitigate shape mismatch concerns, dummy 0 and 1 values were appended to the labels in consideration of the four multi-class categories. This methodology guarantees consistency in the way labels are represented across all classes. Using dummy values facilitates the model's training process, allowing it to accurately classify occurrences of sexism into their corresponding categories while avoiding dimensionality mismatches.

3 Pre-Processing

To generate word embeddings, we utilised a Keras tokeniser with a vocabulary size of 40,000 and GloVe embeddings (100D, 6B tokens) for preprocessing. These embeddings capture the semantic relationships between words by encoding textual data into numerical vectors. To guarantee consistent sequence lengths, the pad sequences function provided by the Keras library was employed, with a maximum size of 256 tokens. This preprocessing pipeline's standardization of input dimensions enables the sexism detection model's training and inference procedures to be executed more efficiently. By tokenizing text and embedding it into a continuous vector space, the model gains the ability to understand and process linguistic patterns, which is crucial for accurately identifying instances of sexism.

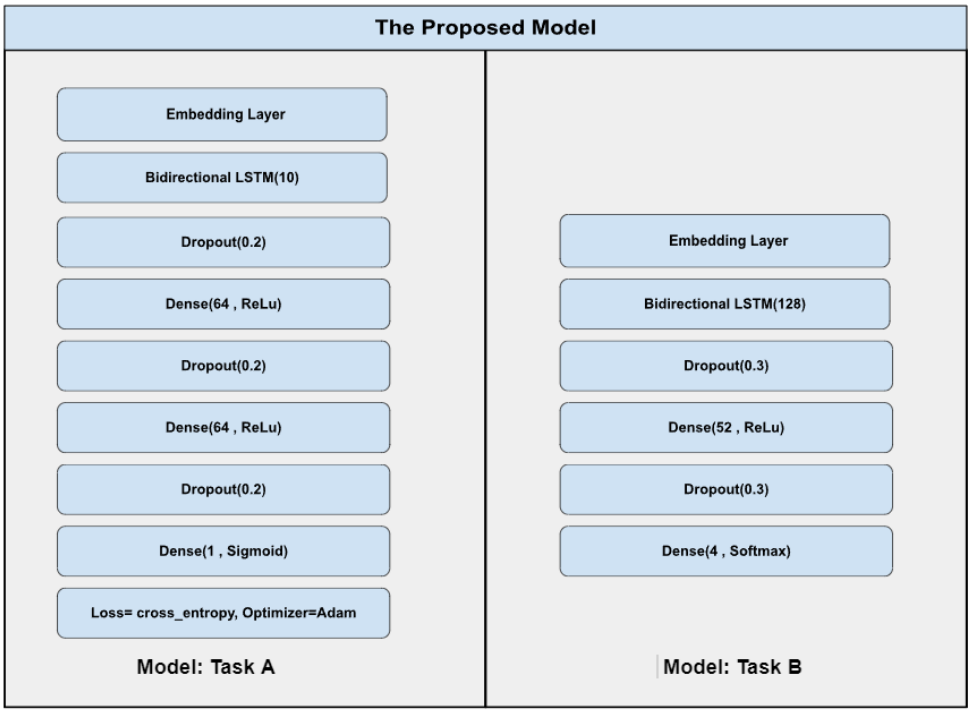
4 Splitting Train-Test Data

For both tasks, 80% of the dataset was allocated for training and 20% for testing, employing a random state 42 for consistency. This split enables the model to learn from diverse examples during training while evaluating its performance on unseen data. The random state aids in result reproducibility across runs. Through this methodology, the machine learns discriminatory patterns from training data, enhancing its ability to accurately detect and categorize instances of sexism in diverse textual contexts.

5 Model

5.1 Task A

Our Bi-LSTM model comprises an Embedding layer followed by a Bidirectional LSTM layer with ten neurons in each direction, totaling twenty neurons. Two Dropout layers with a dropout rate of 0.2 are incorporated after the LSTM layer and each Dense layer. Dropout mitigates overfitting by randomly deactivating 20% of neurons during training, promoting model generalization. The model then includes two Dense layers with 64 neurons each, employing ReLU activation functions to introduce non-linearity and facilitate feature extraction. The final layer comprises a single neuron with a sigmoid activation function, outputting binary classification probabilities. Overall, the model architecture is designed to effectively capture semantic information from input sequences while preventing overfitting through dropout regularisation, ultimately enhancing its ability to generalize to unseen data.



5.2 Task B

For Task B, the model initiates with an Embedding layer, where each word in the input sequence is transformed into a dense vector of dimensionality 100, initialised with pre-trained word embeddings for enriched semantic representation. Subsequently, a Bidirectional LSTM layer with 128 units is employed, allowing the model to capture both forward and backward dependencies within the text, thereby comprehensively understanding contextual nuances. Following the LSTM layer, a Dense layer with 52 neurons and ReLU activation further refine the extracted features, facilitating the model's ability to discern intricate patterns in the data. To prevent overfitting and enhance generalization, a Dropout layer with a 30% dropout rate is applied, randomly deactivating a fraction of neurons during training. This regularisation technique encourages the model to learn robust representations that generalize well to unseen data. The final layer employs softmax activation, producing a probability distribution over the classes and enabling multi-class classification using categorical cross-entropy loss.

6 Evaluation

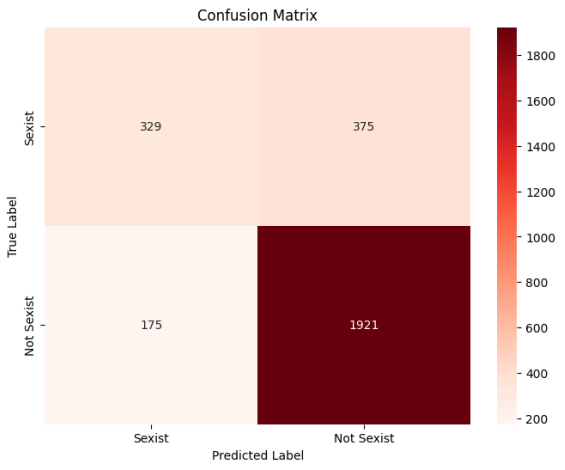
6.1 Task A

For Task A, the model demonstrated strong performance across various evaluation metrics:

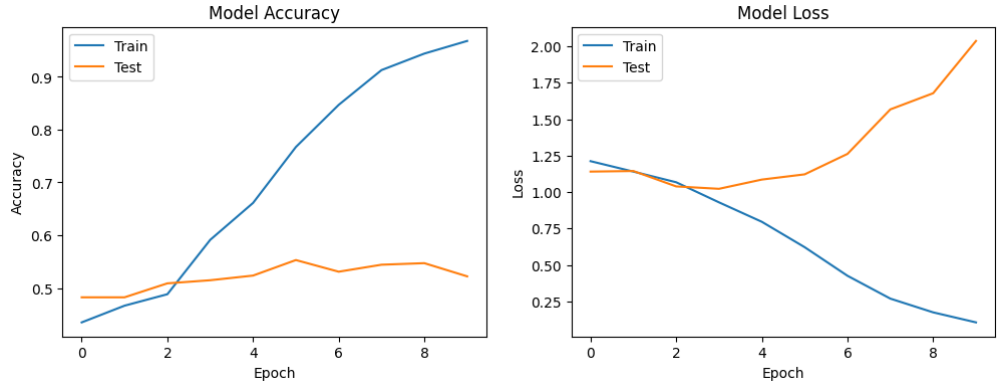
- Training
Accuracy: 90.21% Precision:90.42% Recall: 97.44%

- Testing
Accuracy: 80.36% Precision:82.67% Recall: 91.65%

The progressive reduction in loss across epochs during training indicates that the model effectively learns to minimize errors and improve its predictive capabilities over time. This iterative refinement process indicates the model's ability to adapt and optimize its parameters to fit the training data better.
As the loss decreases, the model becomes increasingly adept at distinguishing between sexist and non-sexist instances, enhancing its overall performance and reliability for sexism detection.
Visualized in a confusion matrix, the model accurately classified 1,921 out of 2,096

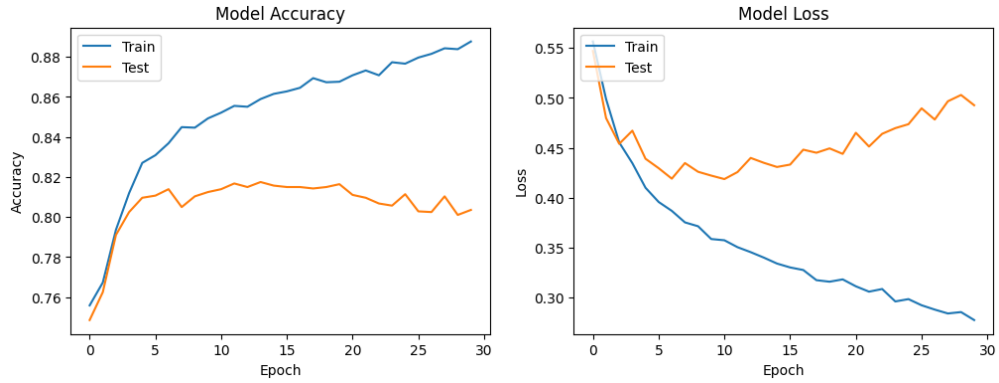


non-sexist instances and 375 out of 704 sexist instances. Moreover, with an F1 score of 87.48%, the model achieves a balanced measure of precision and recall, indicating robust performance in identifying instances of sexism while minimizing false positives and negatives. This metric encapsulates the model's effectiveness in binary classification tasks and underscores its utility as a reliable tool for detecting sexism in textual data.



6.2 Task B

The multi-class sexism detection model exhibited exceptional performance during training, boasting a training accuracy of 99.04%, a precision of 99.11%, and a shallow training loss of 5.1%. However, such stellar metrics starkly contrast with the lackluster performance observed during testing, where the model struggled significantly, achieving only a 52.21% accuracy and a precision of 53.42%. This stark drop in performance between the training and testing phases indicates overfitting, a common challenge when models are trained on insufficient data or when there's a significant class imbalance.



The overfitting issue in this scenario can be primarily attributed to the drastic reduction in the dataset size due to the removal of non-sexist data to address null values. Dropping approximately 10,000 instances out of the original 13,000 undoubtedly led to a loss of crucial diversity in the dataset, exacerbating the overfitting problem. With such a small dataset, the model might have memorized the training examples instead of learning generalizable

patterns, performing poorly on unseen data. Several strategies can be employed to mitigate overfitting and improve model generalization:

1. Augmenting the existing dataset through techniques like data synthesis or oversampling the minority class can help rebalance the dataset and provide the model with more varied examples to learn from.
2. Employing more sophisticated model architectures or fine-tuning hyperparameters through cross-validation can further enhance the model's ability to generalize to unseen data.
3. Collecting more diverse and representative data, especially non-sexist instances, is crucial for improving the model's robustness and ensuring its effectiveness in real-world scenarios.

7 Conclusion

In conclusion, this project introduces an advanced model for automated sexism detection in textual data, leveraging Recurrent Neural Networks with Bidirectional Long Short-Term Memory architecture. With its nuanced linguistic analysis and categorization capabilities, the model shows promising accuracy in identifying sexism across digital platforms. Its refinement holds the potential for scalable solutions to foster inclusivity and equality. Continued research will further enhance these models, advancing efforts to create fairer digital environments.