Cyberbullying classification using RNN and BERT

Sathiyajith K S sxk6394

2 December 2023

1 Abstract

Cyberbullying has become a prevalent issue in online communication platforms, posing significant threats to the well-being of individuals. This paper discusses the approach of classifying cyberbullying tweets with the help of neural network models. Particularly, this paper uses RNN neural network model and BERT Large Language model to classify the tweets among several labelled classes. The reason for choosing RNN for this task is because they excel at learning complex patterns and are well-suited for tasks involving sequential and temporal dependencies. On the other hand, BERT model is used as pretrained model for transfer learning which is powerful for tasks involving limited task-specific data. This paper performs a comparison study on the performances of RNN and BERT models on this multiclass classification task and discusses on how various hyperparameters affect the performance of these models.

2 About Dataset

This dataset contains more than 47,000 tweets labelled according to the class of cyberbullying:

1. Age
2. Ethnicity
3. Gender
4. Religion
5. Other type of cyberbullying
6. Not cyberbullying

The data has been balanced in order to contain 8000 of each class with two columns – text and target. The text column contains the tweet and the target column contains one of the 6 classes specified above.

3 Data Visualization

The dataset contains 5 classes with religion as the most populated class of 7997 rows and other cyberbullying class with least number of rows (7823). There are around 6271 duplicate tweets which when removed leaves a low populated other cyberbullying class of only 4903 tweets. There are also lot of shorter tweets with 10000 tweets less than 10 words. There are a few longer tweets of about 10 tweets that has more than 100 words. So, these duplicate and improper length tweets will be primarily handled in the data preprocessing.

4 Data Preprocessing

This task requires heavy data preprocessing since tweets are informal language with a lot of expressions that donot pertain to english grammar like emojis, hashtags, mentions, urls, abbreviations and symbols.

- Duplicate entries can introduce inaccuracies and inconsistencies in the dataset, leading to compromised data quality. It can also disproportionately influence the outcomes of data analysis or machine learning models, potentially introducing bias. So, the duplicate entries are removed from the dataset.
- Since the text column consists of tweets, there are lot of emojis which are just a group of characters that doesn't add any meaning to the text. Such characters can again introduce inaccuracies. So, the emojis are also removed from the tweets.
- The tweets also contain abbrevations and contractions which in itself doesn't add any meaning, but when expanded adds significant value to the text. Such contractions are expanded.
- The tweets are also from multiple different languages and hence other language tweets are filtered and only english tweets are considered for this problem statement.
- Apart from emojis, the tweets also contain mentions, links, hashtags, excessive spaces, special characters and other non-ASCII characters. Removing them simplifies the data and focuses the analysis on the essential information, improving the signal-to-noise ratio. These are also removed since they just add inaccuracies in the dataset.
- The resulting text is converted to lowercase because converting all text to a consistent case (e.g., lower case or upper case), you eliminate variations in letter casing that may exist in the original data. This helps in standardizing the representation of words and facilitates easier comparisons.
- The words are simplified to their root word with the help of WordNetLemmatizer and PorterStemmer because it ensures that tokens represent the

core meaning of words, preventing unnecessary fragmentation of information.

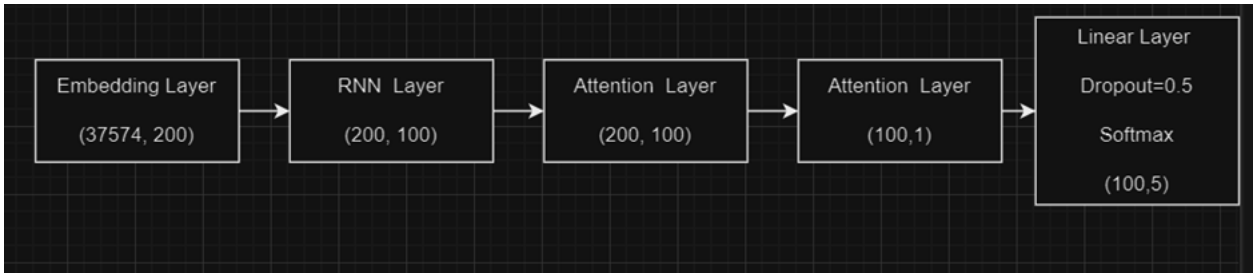
- The shorter tweets are removed since it is hard to find sentiment from shorter sentences. Even the words which are short are removed because they don't add value to the sentences.
- The word sequences are tokenized and padded with max padding to maintain uniformity in the input sequences that are fed to the input layer of the model.

5 Implementation

6 Feature Selection

- The text lengths are plotted and the outliers are removed based on the threshold (100) because Outliers can distort data visualizations, making it challenging to interpret patterns and trends.
- The target column is encoded as integers ranging from 0 to 5 for all the 6 target classes.
- The class imbalance is handled with the help of RandomOverSampler.
- I have used Word2Vec embedding model with 200 as dimension for converting the words to numeric vector inputs to the model.
- Then the embedding matrix is created of shape – vocab size * Embedding dimension.
- The dataset is split into 3 parts – 60% for training, 20% for validation and 20% for testing.
- These individual datasets are then converted to tensor datasets and initialized dataloaders to feed batch-wise data to the model.

7 Model Architecture – RNN



- First the embedding layer converts the word sequences into numerical vector.
- Word2Vec is used as the pretrained embedding model to create a vocabulary and embedding matrix of dimension 200.
- Then the RNN layer processes the word embeddings and passes the output as hidden states to attention layer.
- The 2 attention layers assigns weights to different parts of data based on the relevance. The weights are calculated by applying a tanh activation.
- The Attention layer helps in dealing with long sequences by selectively attending to relevant information. They provide context aware information, which help in capturing dependencies and relaationships between elements in input sequence.
- Then the output is passed to a fully connected layer in which the log Softmax is applied to obtain the probabilities. Finally dropout is applied for regularization and to prevent overfitting.

8 Model Architecture – BERT

BERT is basically an Encoder stack of transformer architecture. A transformer architecture is an encoder-decoder network that uses self-attention on the encoder side and attention on the decoder side. BERTBASE has 12 layers in the Encoder stack with larger feedforward networks (768 units). It contains 512 hidden units and 8 attention heads. BERTBASE contains 110M parameters in total.

- The Bert base model is loaded with input ids and attention masks.
- The input is encoded with the BERT Tokenizer with maximum length padding and special tokens.
- The output of the last hidden state from BERT is passed to the linear layer to apply ReLU activation function and then passed to another fully connected layer.
- The CrossEntropy loss function is used for calculating model's performance since this is a multiclass classification task. Adam Optimizer is used for accelerate the training process by giving optimized learning rates.

9 Challenges and Obstacles

- The task and dataset is heavily dependent on data preprocessing. So, I had to define custom functions with regular expressions and libraries like

demoji to remove emojis. The dataset had other languages too which made the preprocessing task harder. I had used langdetect to detect and filter other language tweets.

- Class imbalance was a big problem since it introduces model bias, poor generalization and difficulty in learning minor patterns. I used RandomSampler to duplicate the entries across the 5 classes and made it uniform.
- Both the RNN and BERT models performed well except for two classes – gender and non-bullying. This could be because the dataset is less populated among these 2 classes. So, more data could be collected in these 2 classes.
- The model gave poorer accuracy because the model parameters became too large or NaN. This problem was due to the large gradient calculated during the backpropagation process. There are many ways to solve this problem like Batch Normalization, Gradient clipping and adjusting model weights etc. Gradient clipping seemed to be the best solution since it controls the gradient value from expanding beyond a threshold.

10 Results and Observations

- The learning rate is set as 4e-4. The model is trained with Adam Optimizer with decay set to the default value of 1e-8.
- The dropout is set as 0.5 since it is the standard practise and also since the model gives maximum accuracy under this dropout, it is not changed further.
- Number of epoch is 10 with early stopping enabled so that whenever the validation accuracy increases the model is saved. And Once the accuracy starts stabilizing, the training is stopped.
- The RNN model gave validation accuracy of 91.402% and prediction accuracy of 93% .
- RNN gives highest F1-score for age and ethnicity classes, while comparatively poorer f1-score for gender and non-bullying classes of about 87% and 81% respectively.
- The BERT model gave validation accuracy of 94.42% and prediction accuracy of 94%.
- Similar to RNN, BERT also gives highest F1-score for age and ethnicity classes, while comparatively poorer scores for gender and non-bullying classes of about 92% and 84% respectively.

- The reason for this trend could be because a lot of gender bullied tweets are classified as non-bullying and a lot of non-bullying tweets are not classified correctly.
- From the results above, we can conclude that the BERT base model performs better than RNN model for this sentiment analysis task of classifying bullying tweets.