CS918 Natural Language Processing Assignment 2 : Sentiment Classification for Social Media - Report

2222991

March 21, 2024

This is a report for the CS918 Natural Language Processing Assignment 2 : Sentiment Classification for Social Media. In this assignment, we have developed machine learning models including Support Vector Machines (SVM), Naïve Bayes (NB) and Maximum Entropy (MaxEnt), as well as a Long Short-Term Memory (LSTM) network with attention mechanisms. This report summarises the structure and techniques used in the classification and the performance results.

1 Pre-processing

The first step before building the models is to preprocess the data, which involves data cleaning, tokenizing, and lemmatizing.
For data cleaning, the following steps are performed:

- Lowercasing all words in tweets to ensure uniformity.
- Replacing emojis with their descriptions by using 'demoji', as the sentiments in emojis play significant role in the sentiment classification.
- Expanding contractions to standardise text.

Additionally, the following steps are performed by regex patterns: Removing URLs, user mentions, non-alphanumeric characters, single-character words, numbers that are solely made of digits and combinations of digits and alphabets, which are all irrelevant for sentiment analysis. However, hashtags in tweets were kept as they contribute on conveying sentiment, especially on social media platform.
Afterward, words in tweets are tokenized and lemmatized to their base forms with the help of NLTK's part-of-speech tagging, which helps to select the correct lemma for words, thereby improves accuracy during lemmatization.

2 Traditional machine learning methods

2.1 Feature Extraction

Two feature extraction methods are employed for traditional machine learning models:

- Bag of Words (BoW): This approach counts word occurrences of words. Function 'CountVectorizer()' from sklearn is used.
- TF-IDF (Term Frequency-Inverse Document Frequency): This approach quantifies the importance of a word to a document within a collection of documents, relative to frequency. Function 'TfidfVectorizer()' from sklearn is used, with parameters `max_features = 5000` and `min_df = 3`. These parameters were found by using GridSearchCV.

2.2 Machine Learning Models

Support Vector Machine (SVM): 'LinearSVC' implementation is chosen on this sentiment classification problem. For feature BoW, parameters `'C = 1' and 'max_iter = 5000'` are used. For feature TFIDF, parameters `'C = 1'` is used. All theses parameters were determined by using GridSearchCV.
Naïve Bayes (NB): 'MultinomialNB' implementation is chosen on this sentiment classification problem. Parameter `'alpha = 1'` is used and was found by using GridSearchCV.
Maximum Entropy (MaxEnt): MaxEnt, aka Logistic Regression in sklearn has also been implemented in this sentiment classification problem. Parameters `'C = 1', 'max_iter = 1000'` are used. All theses parameters were determined by using GridSearchCV.

3 Neural Network Machine Learning Methods

3.1 Feature Extraction

Word embeddings using pre-trained GloVe vectors were employed by following the instruction. This feature helps to represent the words in vector space and capture semantic similarities between words based on distribution. The embedding matrix, used as embedding layer of the LSTM, was constructed from a reference word index. To follow the instruction of 5,000 maximum number of tokens limit, we iterated over words extracted from training data which sorted by frequency. We then retrieved and assigned the corresponding GloVe vector only for those words found within GloVe embedding index. This approach ensures that as much information as possible about the words in both training data and GloVe embedding is captured, forming the embedding layer of the LSTM model.

3.2 Model

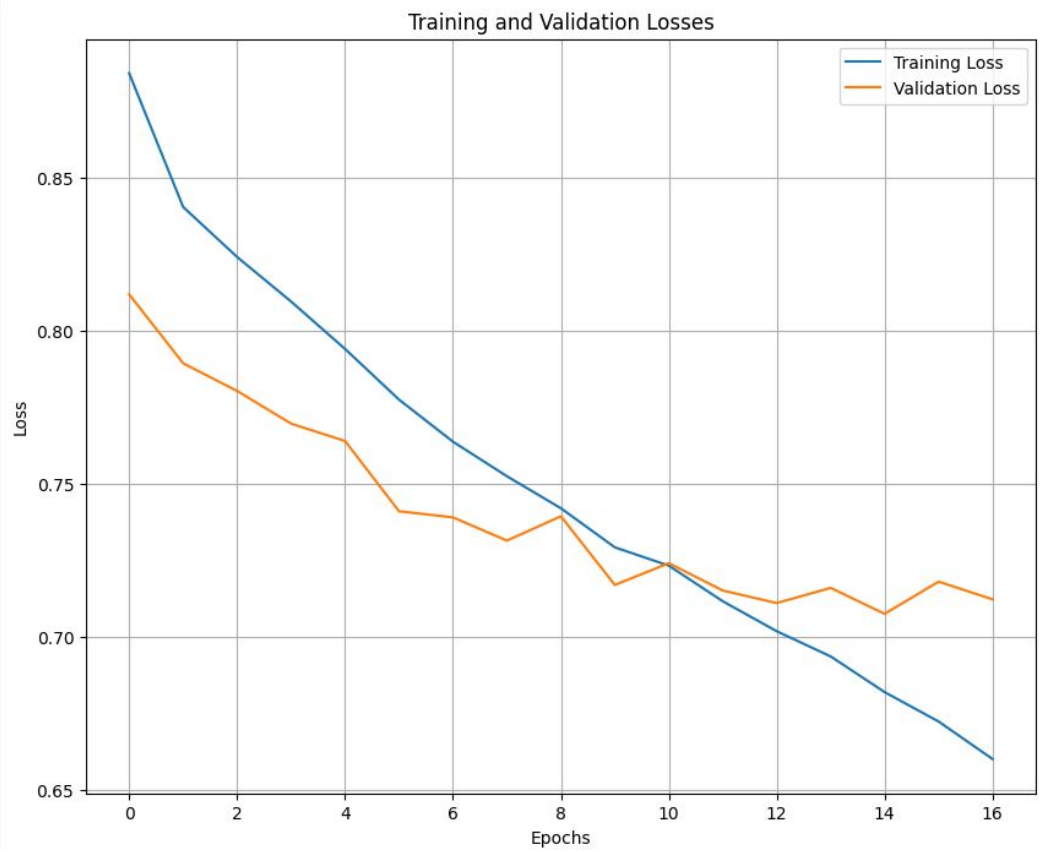
3.2.1 Bidirectional LSTM model with attention mechanism

In addition to the original architecture specified in the instructions of assignment, we have developed a LSTM model with enhanced bidirectional LSTM and an additional layer of attention mechanism.
The bidirectional LSTM improves performance by processing data in both forward and backward directions. It captures both the preceding and following words of each tweet,

enabling the model to better understand the sentiment in each tweet.
The attention mechanism, on the other hand, helps to weigh the importance of different words within a tweet, allowing the model to focus on the most informative parts of sentiments.
The following hyperparameters and configurations were fully tested and found to perform best for the model:

- Loss function used: CrossEntropyLoss(); to effectively measure the difference between the predicted and actual distributions for such multi-class classification.
- Size of the LSTM's hidden layers: 1024
- Learning rate: 0.001
- Batch size: 64
- Dropout rate: 0.4
- L2 regularization factor: 0.00001

The model implement early stopping mechanism to prevent overfitting. The early stopping patience parameter was set to 2, meaning the model will cease training if the validation loss does not reach its lowest value for two consecutive epochs. Below is a plot of the training and validation losses during training:



3.2.2 Zero-Shot Classification with BART

To evaluate the performance of state-of-the-art language models on sentiment analysis, we have also implemented a BART model pre-trained on the MNLI dataset for testing and reference. This test employs Zero-Shot Classification, which means that raw tweets are input directly into the model without any fine-tuning to assess whether it can correctly classify the target sentiment label of the tweets.

4 Evaluation

The macroaveraged F1 score output of all classifier:

Model	**test1**	**test2**	**test3**
bow-svm	0.535	0.564	0.503
TFIDF-svm	0.581	0.602	0.542
bow-nb	0.496	0.461	0.496
TFIDF-nb	0.400	0.431	0.401
bow-MaxEnt	0.574	0.573	0.536
TFIDF-MaxEnt	0.571	0.581	0.532
GloVe Embeddings-LSTM	**0.636**	**0.642**	**0.574**
BART(Reference)	0.620	0.635	0.600

5 Error Analysis and Insights

During the testing of models, It was found that excluding of using stop words during tokenization and lemmatization actually improved performance. This indicates that stop words carry sentiment information which helps the models, especially LSTM, learn to classifier better.
Also, the dataseet is imbalanced, with 'neutral' sentiments outnumbering 'positive' and 'negative', and the number of tweets with positive sentiments also is double of the number of tweets with negative sentiments. This could bias the model's predictions towards the positive and negative classes.

6 Reference

`https://huggingface.co/facebook/bart-large-mnli`