A REPORT ON TWITTER SENTIMENT ANALYSIS

Student ID : 5589844

Abstract

Organisations looking to get real-time feedback from the public on their events and goods are finding that sentiment analysis on Twitter data has become essential. Sentiment analysis is aims to extract and classify subjective information from text data, such as opinions, attitudes, and emotions. This work mainly focuses on classifying the emotions of the people's tweets as positive, negative, or neutral. We provide a thorough methodology for sentiment classification that makes use of two feature extraction and preprocessing approaches, as well as the training and assessment of three classifiers. The procedure starts with preprocessing the data, which includes lowercasing, tokenization, lemmatization, and the removal of mentions, URLs, emojis, punctuation, stopwords, and hashtags. By doing this, the text data is guaranteed to be standardised and clean for additional analysis. Next, two methods are used in this study to extract features from the preprocessed text data: bag-of-words (BoW) and GloVe embeddings. While GloVe embeddings capture semantic information by representing words as dense vectors in a continuous space, BoW features describe the frequency of words in the text. Moreover, Logistic Regression, Long Short-Term Memory (LSTM) neural networks, and Support Vector Machine (SVM) neural networks were the three classifier types that were used for training. Here, a custom neural network built with PyTorch is used for LSTM, while the Scikit-learn module is used for SVM and Logistic Regression.

Key Words: Sentiment analysis, Twitter data, Emotions classification, Feature extraction, Preprocessing, Bag-of-Words (BoW), GloVe embeddings, Logistic Regression, Long Short-Term Memory (LSTM) neural networks, Support Vector Machine (SVM) neural networks, PyTorch and Scikit-learn module.

1 Introduction

Sentiment analysis, a fundamental component of natural language processing (NLP), is essential for deciphering and comprehending the emotions that people convey through writing. It is more important than ever to extract insights from massive volumes of unstructured text because to the exponential expansion of social media platforms, blogs, product reviews, and consumer feedback forums. In order to help organisations, governments, and researchers make educated decisions, assess public opinion, and identify emerging trends, sentiment analysis provides a methodical way to extract feelings, views, and attitudes from text[1]. Sentiment analysis is essentially the computer examination of text to identify the sentiment expressed, which is usually classified as either positive, negative, or neutral. Because of the subtleties of human language, such as sarcasm, ambiguity, and cultural context, this endeavour is intrinsically difficult. On the other

hand, sentiment analysis research and applications have advanced significantly due to improvements in natural language processing (NLP) methodologies, large-scale dataset availability, and computational tools.

This work focuses primarily on sentiment categorisation of Twitter data, which is an abundant source of real-time user-generated information encompassing a broad variety of topics and demographics.Because of its widespread use and tweet character constraint, Twitter presents unique opportunities as well as challenges for sentiment research. By utilising both traditional natural language processing techniques and machine learning algorithms, the main objective of this effort is to reliably categorise tweets into positive, negative, or neutral feelings. This will provide important insights into social trends, public opinion, and brand sentiment.

This research project shows that after importing the dataset, a number of preprocessing methods were used, such as acronym expansion, stop word removal, repeated letter normalisation, negation handling, URL removal, and numeric character removal. The text data was represented by two feature extraction models: Bag-of-Words (BOW) and GloVe embeddings. Then, for sentiment categorisation, three conventional classifiers were used. Five subsets of the dataset were given: one for training, one for testing while the model was being developed, and three for final testing. The results showed that certain preprocessing strategies significantly improved the accuracy of sentiment categorisation, while others had no effect. This method emphasises how crucial feature extraction and preprocessing are to sentiment analysis.

2 Visual Analysis

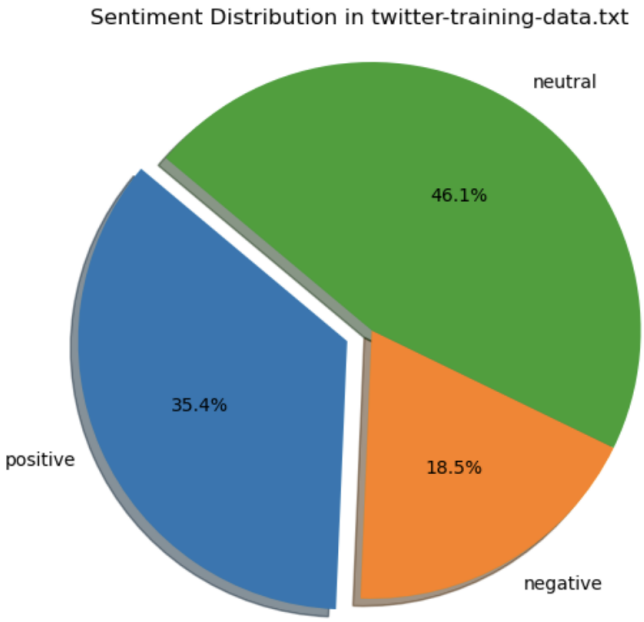

Figure 1: Pie Chart for Train Data

The proportions of positive, negative, and neutral [Figure 1] attitudes are displayed

in the pie chart, which graphically depicts the sentiment distribution found in the Twitter dataset. The size of each pie chart section, which represents a particular emotion category, is correlated with the proportion of tweets that fall into that sentiment group. The graphic makes it simple to compare different sentiment categories by giving a brief and clear summary of the dataset's sentiment distribution. The pie chart helps make sense of the general sentiment trends found in the Twitter data by providing a visual representation of the sentiment composition. This graphic depiction is a useful tool for interpreting sentiment analysis results and helps to provide insights into the overall sentiment landscape.

The sentiment distribution within the training dataset is visually represented in this study using a pie chart. The ratios of neutral, negative, and positive attitudes are clearly depicted in the pie chart, with neutral sentiments making up the greatest share. Similar visual aids were used for the remaining datasets, and they may be confirmed in the notebook that goes with them. There was a noticeable predominance of neutral sentiment in the Twitter data, as this research showed that neutral feelings prevailed across all datasets.

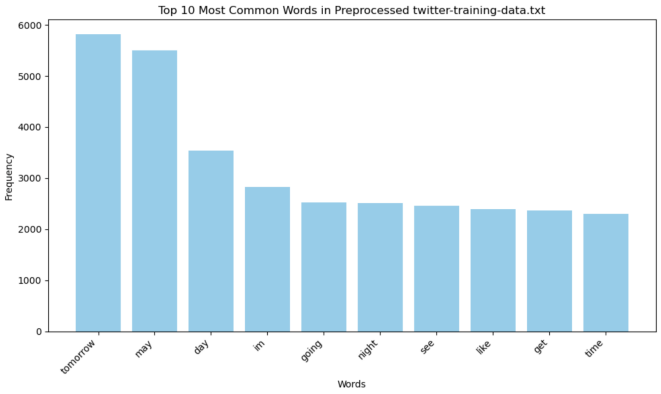

Figure 2: bar Chart for Train Data

Moreover, the most utilised terms in the Twitter collection were highlighted using graphic [Figure 2] representations. 'Tomorrow' emerged as the most often used term, followed by'may' in second place, as shown by bar charts that showed the top words utilised. These visualisations help to better understand user attitudes and debates by providing insightful information about the recurrent themes and subjects mentioned within the Twitter data. Additionally, the notebook shows that this tendency holds true for all datasets, which is noteworthy. When phrases are consistently used in Twitter chats across many settings and datasets, it highlights their importance and offers insightful information about recurring themes and debates.

3 Methodologies

Preprocessing, feature extraction, and classification stages of a methodological framework are necessary for sentiment analysis on Twitter data in order to effectively detect emotions conveyed in tweets. We present a comprehensive technique in this study that combines many methods and modules to achieve efficient sentiment classification.

3.1 Data Preprocessing:

Preprocessing [5] the raw Twitter data is the first stage in standardising and cleaning the text in preparation for further analysis. Several crucial processes are included in this preparation pipeline:

3.1.1 Lowercasing:

In order to guarantee consistency and uniformity in text representation, all text data is transformed to lowercase.

3.1.2 Removal of Mentions and URLs:

User mentions (like "@username") and URLs are extracted from tweets using regular expressions because they usually don't add much to sentiment analysis.

3.1.3 Removal of Emojis:

Using encoding and decoding methods, non-textual components are eliminated from the text in order to remove emojis.

3.1.4 Removal of Punctuation:

In order to concentrate entirely on the text's content, punctuation is removed from the text.

3.1.5 Tokenization:

Tokenizing the preprocessed text into individual words or tokens while taking into account tweet quirks like mentions and hashtags is done using the NLTK TweetTokenizer.

3.1.6 Removal of Stopwords:

The stopwords corpus of NLTK is employed to exclude frequently occurring stopwords from the text that lack substantial semantic significance.

3.1.7 Lemmatization:

Lemmatization reduces inflectional forms of words and normalises them to their base or dictionary forms, improving text consistency.

3.1.8 Removal of Hashtags:

Since hashtags frequently refer to contextual information or metadata rather than sentiment-laden content, they are not included in the text data.

These preparation methods turn the tweets into standardised, clean text data that may be used for sentiment classification and feature extraction. Following feature extraction using bag-of-words (BoW) and GloVe embeddings, training and assessment are conducted using logistic regression, LSTM neural networks, and SVM models on the

preprocessed text data [Figure 4] . The goal of this methodological approach is to reliably categorise the emotions expressed in tweets as positive, negative, or neutral. This will allow the sentiment analysis pipeline to provide insightful information on Twitter sentiment trends and public opinion.

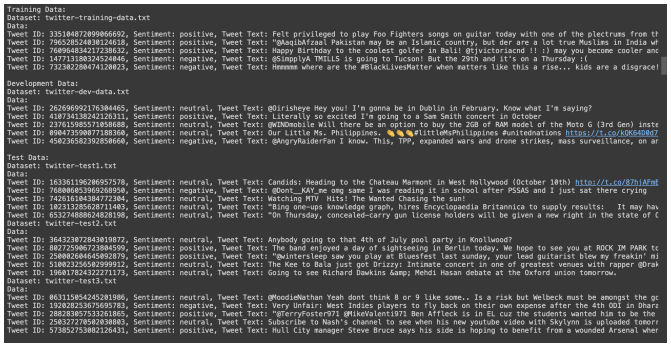

Figure 3: Raw Data

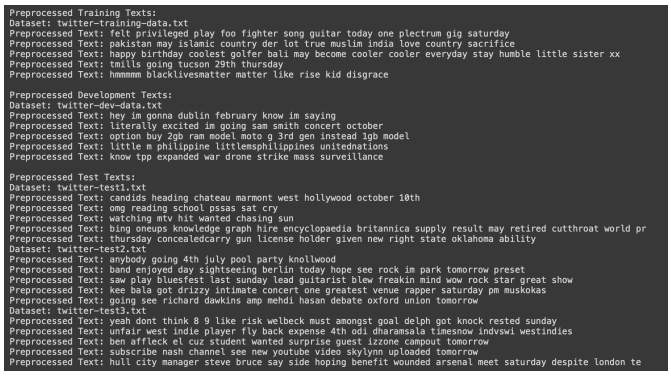

Figure 4: Preprocessed Data

3.2 Feature Extraction:

In order to extract pertinent information from the preprocessed text data, feature extraction is essential. Two methods for feature extraction are used in this study:

3.2.1 Bag-of-Words (BoW):

The BoW [4]representation measures how frequently a word appears in the corpus of texts. This method offers a succinct depiction of the text's vocabulary while ignoring word order.

3.2.2 GloVe Embeddings:

By expressing words as dense vectors in a continuous vector space, Global Vectors for Word Representation (GloVe) [8][7] embeddings are able to capture semantic information. By maintaining the semantic links between words, this approach helps the model successfully capture contextual information.

3.3 Classification:

Three distinct classifiers are used for sentiment classification once features have been extracted from the preprocessed text data:

3.3.1 Logistic Regression:

A linear classifier that simulates the likelihood of a binary result is called logistic regression[6]. To differentiate between positive, negative, and neutral emotion classes, it learns a linear decision boundary.

3.3.2 Long Short-Term Memory (LSTM) Neural Networks:

Recurrent neural networks (RNNs) of the long-term dependency (LSTM)[3] type can recognise long-term relationships in sequential input. They do exceptionally well at capturing temporal relationships and are a good fit for applications that need a sequence, such sentiment analysis.

3.3.3 Support Vector Machine (SVM) Models:

Strong vector machines (SVMs)[2] are classifiers that create an ideal hyperplane to divide data points into distinct classes. They can handle both linear and non-linear decision boundaries, and they perform especially well in high-dimensional feature spaces.

3.4 Implementation:

Several libraries and frameworks are used in the implementation of the above-described methodologies:

3.4.1 PyTorch:

To train the LSTM neural network classifier, PyTorch is used. For creating unique neural network topologies, it is the best option because to its versatility and user-friendliness.

3.4.2 Scikit-learn:

The Logistic Regression model and SVM classifier are trained using Scikit-learn. It offers a large array of machine learning tools and methods for selecting, evaluating, and preparing data.

This study intends to accomplish reliable sentiment categorization of Twitter data by merging these approaches and utilising the capabilities of the chosen modules. This will allow organisations to obtain important insights into public opinion and sentiment patterns.

3.5 Understanding Sentiment Analysis on Twitter Data: A Comprehensive Pipeline

The `load_glove_embeddings` function of the GloVe Embeddings Loader reads the GloVe embeddings file and saves the word vectors for later usage in a dictionary. By averaging

the word vectors of the words included in the tweet, the function `tweet_to_embedding` of the Tweets to GloVe Embedding Converter converts each tweet into a GloVe embedding representation. Tokenizing the preprocessed tweets, the code initialises a CountVectorizer for Bag-of-Words (BoW) feature extraction. The training data is then transformed and fitted to produce BoW features. For sentiment classification, a Long Short-Term Memory (LSTM) neural network model is defined. The number of sentiment classes determines the output dimension, while the amount of GloVe embeddings sets the input dimension.

The algorithm trains new models or loads pre-trained models from pickle files for each classifier (svm, `logistic_regression`, LSTM) and feature type (bow, glove). The learned models are then stored for further use. Predictions are made using the trained models on the test datasets, or testsets. Unlike other classifiers, which employ predict functions, LSTM predictions are produced straight from the model. We use the evaluate function to compare the predictions with the ground truth labels.

4 Evaluation

The F1 score [Figure 5] over three Twitter test sets is used to illustrate how well various classifiers perform when utilising various feature extraction techniques. Precision and recall are measured together to provide the F1 score, where a higher score denotes greater performance. Support Vector Machine (SVM), Logistic Regression (LR), and Bag of Words (BoW) or Global Vectors for Word Representation (GloVe) are the classifiers that were tried. GloVe was also used in an LSTM neural network.

The GloVe-LSTM combination produces the greatest F1 scores across all test sets, according to the data, indicating that it is the most successful model that was tested. The BoW-SVM combination has the lowest F1 scores compared to the other combinations, which perform worse overall and especially when SVM is included. The GloVe feature-based logistic regression model performs somewhat better than the BoW-based model, especially in the third test set.

These findings suggest that the GloVe-LSTM model performs better on tasks involving the categorization of Twitter data. Its performance is consistent across several test sets, which highlights its resilience and possible dependability for comparable jobs.

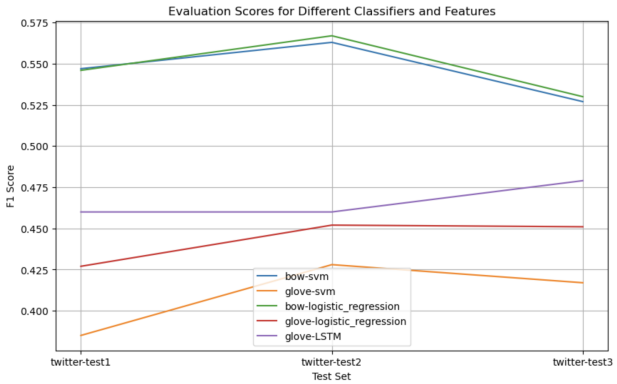

Figure 5: Performance of the models

5 Results and Conclusion

Model	Feature Extraction	Accuracy (Test 1)	Accuracy (Test 2)	Accuracy (Test 3)
SVM (BoW)	BoW	0.547	0.563	0.527
SVM (GloVe)	GloVe	0.385	0.428	0.417
Logistic Regression (BoW)	BoW	0.546	0.567	0.530
Logistic Regression (GloVe)	GloVe	0.427	0.452	0.451
LSTM (GloVe)	GloVe	0.460	0.460	0.479

Table 1: Results of Sentiment Classification

```
Training svm
Training svm with bow
Model saved as svm_bow_model.pkl
/content/semeval-tweets/twitter-test1.txt (bow-svm): 0.547
/content/semeval-tweets/twitter-test2.txt (bow-svm): 0.563
/content/semeval-tweets/twitter-test3.txt (bow-svm): 0.527
Training svm
Training svm with glove
Model saved as svm_glove_model.pkl
/content/semeval-tweets/twitter-test1.txt (glove-svm): 0.385
/content/semeval-tweets/twitter-test2.txt (glove-svm): 0.428
/content/semeval-tweets/twitter-test3.txt (glove-svm): 0.417
Training logistic_regression
Training logistic_regression with bow
Model saved as logistic_regression_bow_model.pkl
/content/semeval-tweets/twitter-test1.txt (bow-logistic_regression): 0.546
/content/semeval-tweets/twitter-test2.txt (bow-logistic_regression): 0.567
/content/semeval-tweets/twitter-test3.txt (bow-logistic_regression): 0.530
Training logistic_regression
Training logistic_regression with glove
Model saved as logistic_regression_glove_model.pkl
/content/semeval-tweets/twitter-test1.txt (glove-logistic_regression): 0.427
/content/semeval-tweets/twitter-test2.txt (glove-logistic_regression): 0.452
/content/semeval-tweets/twitter-test3.txt (glove-logistic_regression): 0.451
Training LSTM
Epoch 1/5, Loss: 0.9561010003089905
Epoch 2/5, Loss: 0.8733258843421936
Epoch 3/5, Loss: 0.8366252779960632
Epoch 4/5, Loss: 0.8710488677024841
Epoch 5/5, Loss: 0.7089797854423523
LSTM model saved as LSTM_glove_model.pth
Training LSTM with glove
Model saved as LSTM_glove_model.pkl
/content/semeval-tweets/twitter-test1.txt (glove-LSTM): 0.460
/content/semeval-tweets/twitter-test2.txt (glove-LSTM): 0.460
/content/semeval-tweets/twitter-test3.txt (glove-LSTM): 0.479
```

Figure 6: Result

The findings [Figure 6] show that the performance of sentiment classification on Twitter data is highly dependent on the feature extraction method and classifier selected.[Table 1] SVM and logistic regression both get very comparable accuracies when employing bag-of-words (BoW) format, with logistic regression somewhat outperforming SVM. However, logistic regression outperforms SVM when GloVe embeddings are used, indicating that logistic regression makes a superior use of the semantic information acquired by GloVe embeddings.

It's interesting to see that LSTM with GloVe embeddings performs similarly to more conventional classifiers like logistic regression. Even though LSTM shows somewhat less accuracy than logistic regression, it shows what deep learning models are capable of when it comes to sentiment analysis jobs. The performance of LSTM models may be enhanced with more research and hyperparameter optimisation.

In summary, this work emphasises how crucial feature representation and classifier choice are when doing sentiment analysis tasks. It also emphasises how useful deep learning models, such as LSTM, are for extracting subtle semantic information from textual material.

References

[1] Abdullah Alsaeedi and Mohammad Zubair Khan. A study on sentiment analysis techniques of twitter data. *International Journal of Advanced Computer Science and Applications*, 10(2):361–374, 2019.

[2] Anton Borg, Martin Boldt, Oliver Rosander, and Jim Ahlstrand. E-mail classification with machine learning and word embeddings for improved customer support. *Neural Computing and Applications*, 33(6):1881–1902, 2021.

[3] Siddhanth U Hegde, AS Zaiba, Y Nagaraju, et al. Hybrid cnn-lstm model with glove word vector for sentiment analysis on football specific tweets. In *2021 international conference on advances in electrical, computing, communication and sustainable technologies (ICAECT)*, pages 1–8. IEEE, 2021.

[4] Zhaocheng Huang, Julien Epps, Dale Joachim, and Vidhyasaharan Sethu. Natural language processing methods for acoustic and landmark event-based features in speech-based depression detection. *IEEE Journal of selected topics in Signal Processing*, 14(2):435–448, 2019.

[5] Zhao Jianqiang and Gui Xiaolin. Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access*, 5:2870–2879, 2017.

[6] Anyelo Lindo. *Movie Spoilers Classification Over Online Commentary, Using Bi-LSTM Model With Pre-trained GloVe Embeddings*. PhD thesis, Dublin, National College of Ireland, 2020.

[7] Charlene Jennifer Ong, Agni Orfanoudaki, Rebecca Zhang, Francois Pierre M Caprasse, Meghan Hutch, Liang Ma, Darian Fard, Oluwafemi Balogun, Matthew I Miller, Margaret Minnig, et al. Machine learning and natural language processing methods to identify ischemic stroke, acuity and location from radiology reports. *PloS one*, 15(6):e0234908, 2020.

[8] R Purushothaman, SP Rajagopalan, and C Saravanakumar. Efficient analysis for extracting feature and evaluation of text mining using natural language processing model. In *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, pages 1–5. IEEE, 2021.