Applying Sentiment Analysis on Twitter

Aditya Matiwala (asm14@illinois.edu)
Srijan Kunta (skunta2@illinois.edu)

1 Summary

Twitter provides a real-time stream of public sentiment, making sentiment analysis crucial in order to understand different opinions. This project aims to develop a Tweet sentiment classifier that categorizes tweets as positive, negative, or neutral. This project will use Naive Bayes and logistic regression for text classification. To improve classification accuracy, we will use Word2Vec embeddings to capture semantic relationships between words in order to offer a more meaningful representation compared to TF-IDF approaches. We will also train and evaluate our models on publicly available datasets or by training our own data by scraping and manually labeling data that is populated from a public Twitter API or TweePy. Performance of the models will be measured and compared using accuracy, precision, recall, and F1-score. By comparing these different text representation techniques, this project will provide an insight into effective methods for sentiment classification while also providing an avenue for social media monitoring.

2 Software and Implementation

With regards to software, we will be using Python for its extensive support for machine learning and NLP libraries. This project will follow a structured pipeline which involves data preprocessing, feature extraction using Word2Vec, subsequent model training, and final evaluation. Before training the model, we will preprocess tweets to clean and normalize the text, which includes tokenization, or splitting text into individual words or subwords, stopword removal, or eliminating common words that do not contribute to a sentiment, and finally stemming and lemmatization, or converting words to their base form (e.g. talking -> talk). After preprocessing the data, we will leverage Word2Vec embeddings which will convert the text into numerical vectors. This is ideal since Word2Vec allows us to understand semantic relationships between the words and also improves generalization to better understand word similarities. Regarding model training and classification, we will train and compare two classification models which are Naïve Bayes Classifier and Logistic Regression. Both models will be trained on Word2Vec word embeddings. Finally, we will evaluate the models' results in order to further tune hyperparameters like regularization strength for logistic regression.

3 Proposed Datasets

We will first try to use existing public datasets that include sentiment labels. If we can't find an appropriate dataset, we will build our own dataset using the Twitter API, using tools such as Tweepy, an open source Python library to access and scrape TwitterAPI. We will use a query-based approach with Tweepy, using hashtags, keywords, and possibly user interactions, to get various posts that display emotion. We will try to tune our dataset to be diverse and balanced. We will try to find datasets that are already labeled, but if we can't find adequate datasets, then we will have to scrape and manually label our own data.

4 Evaluation and Testing

We will evaluate our model with standard metrics such as accuracy, precision, recall, and F1 score, using a standard 80/20 training and test split. We can use confusion matrices to look for possible patterns and model bias. We will also compare our model against good pre-trained sentiment analysis models such as VADER or models from Hugging Face.

5 Timeline

Week 1

- Collect data using TwitterAPI with Tweepy (2-4 hours)
- Pre-processing and labeling dataset (2-4 hours)

Week 2

- Word2Vec embeddings(2-4 hours)
- Implement and train Naive Bayes Classifier (5-7 hours)

Week 3

- Implement and train Logistic Regression (5-7 hours)

Week 4 (Milestone)

- Train Both models using word2vec embeddings (5-7 hours)

Week 5

- Hyperparameter tuning (2-3 hours)
- Computing Evaluation Metrics (2-3 hours)

Week 6

- Final Report (3-5 hours)
- Final Video/Presentation (3-5 hours)

Listed above are high-level tasks that need to be completed for this project. Each task has a corresponding estimated time for completion. Both of us will be working on the tasks concurrently as these tasks are dependent on each other. We will be meeting on Mondays and Wednesdays at 4 PM to work on this project. The timeline is more aggressive/front-loaded in order to allow for more flexibility in time between the milestone and final presentation.