Project Deliverable 2

Twitter Sentiment Analyzer

Cindy Wang
https://github.com/cindywang3299/twitter-sentiment-analyzer

1 Problem statement

Twitter Sentiment Analyzer will determine the sentiment (positive, negative or neutral) present in a piece of writing. It analyzes topics by parsing the tweets fetched from Twitter using Python.

2 Data Preprocessing

The dataset chosen for training the sentiment analysis model is from "Sentiment140"(`http://help.sentiment140.com/for-students/`). The specified dataset contains 1.6 million samples and is labelled in the following format:

0 - the polarity of the tweet (0 = negative, 2 = neutral, 4 = positive)
1 - the id of the tweet (2087)
2 - the date of the tweet (Sat May 16 23:58:44 UTC 2009)
3 - the query (lyx). If there is no query, then this value is `NO_QUERY`
4 - the user that tweeted (robotickilldozr)
5 - the text of the tweet (Lyx is cool)

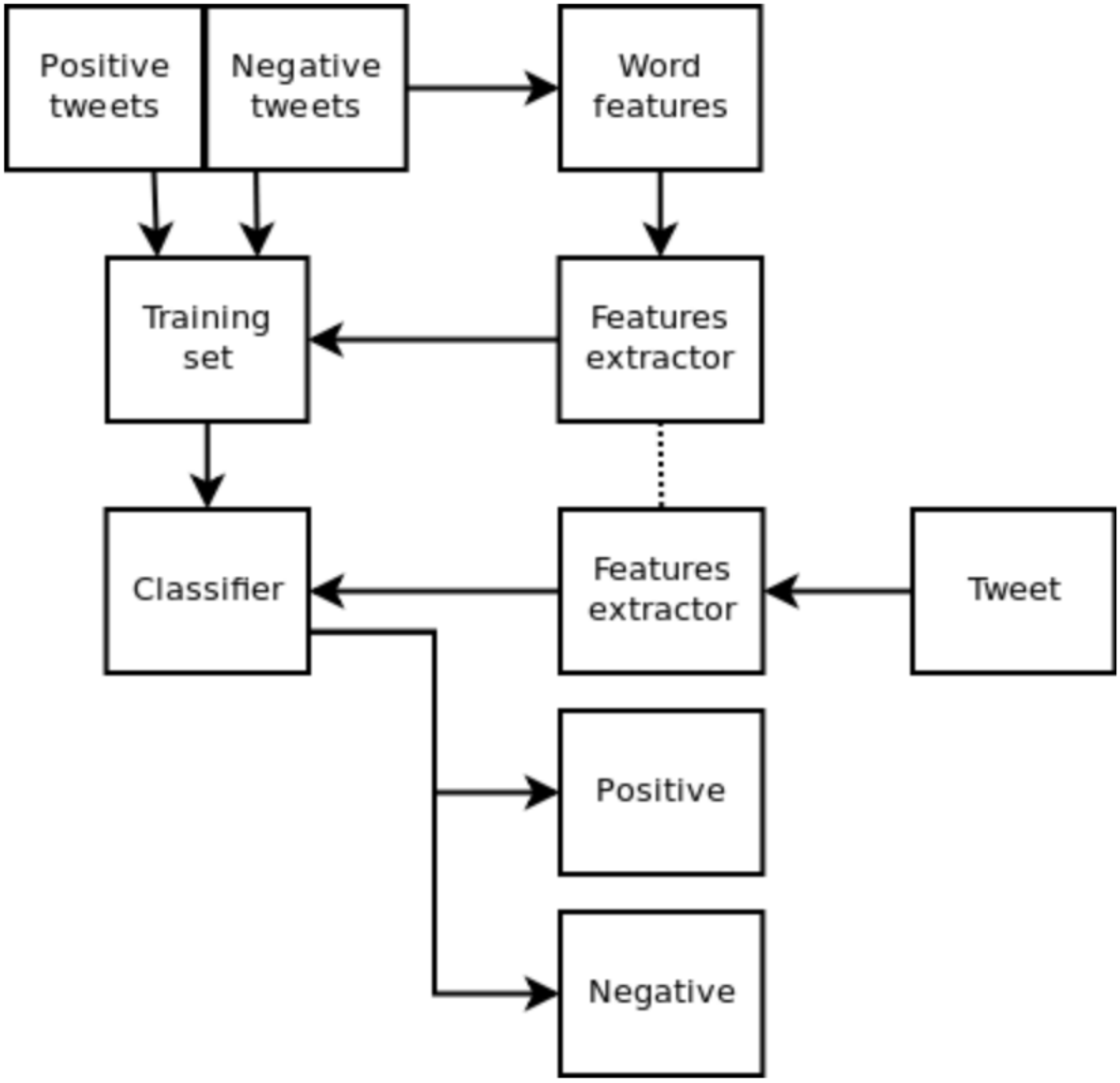
The objective of this step is to clean noise those are less relevant to find the sentiment of tweets such as punctuation, special characters, numbers, and terms which don't carry much weight in context to the text. The data will be preprocessed in the following ways:

- Removing Twitter handles (@user)
- Removing punctuations, numbers, and special characters
- Removing short words
- Tokenization
- Stemming

3 Machine learning model

The model has been adapted from a simple linear regression model. It predicts the probability of occurrence of an event by fitting the data to a logit function. The bag-of-words approach was used to convert textual data into numerical representation.

After preprocessing, the dataset now contains clean, textual information ready for analysis. Before training the model, I have classified the data by modifying features from the original dataset to filter tweets by negative sentiment (target = 0) and positive sentiment (target = 1). The classification process is shown in the following graph:



I have split the data into three sets for training, development, and testing purposes. The ratio I chose to split my data is 98/1/1 for training, development, and testing separately. Since the size of chosen dataset is relatively large, it is more than enough to evaluate the model and refine the parameters. Here, the training set refers to the sample of data used for learning by the model; the development set is the sample of data used to tune the parameters of a classifier and provide an unbiased evaluation of a model (cross validation set); the testing set will be used only to assess the performance of the final model.
The data has been split two times for cross-validation, the first to split train and the development test, and the second to split dev and test. I referenced *Andrew Ng's "deeplearning.ai" course* (`https://www.coursera.org/learn/machine-learning-projects`) on how to split the data.

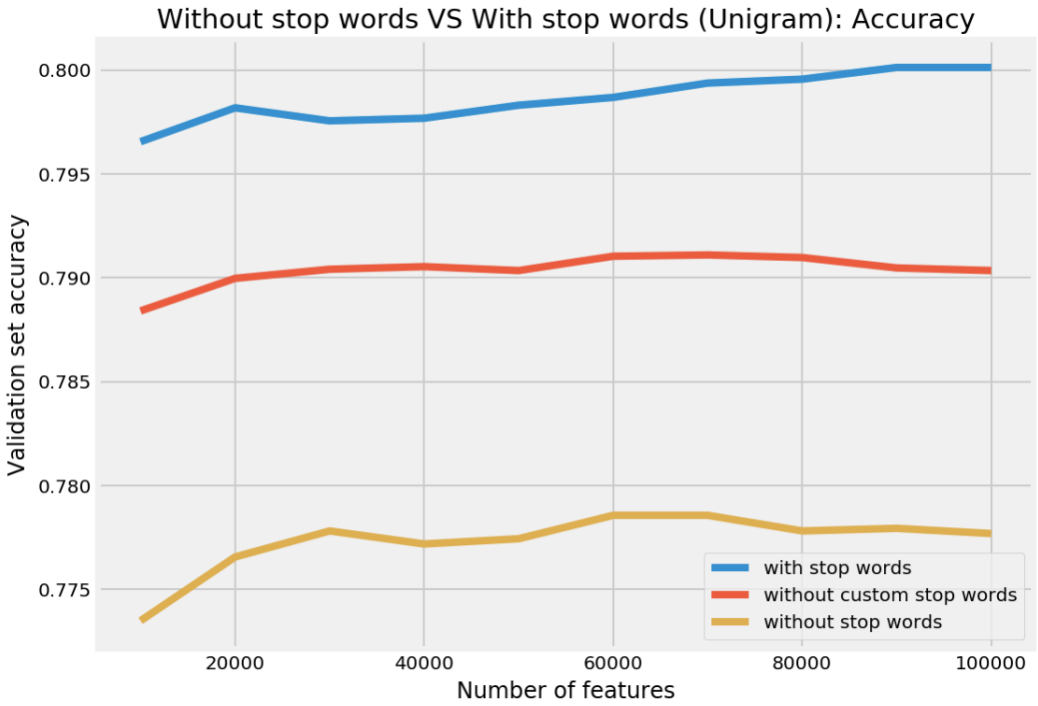
Baseline provides a point of reference to compare when comparing various machine learning algorithms. I imported the ZeroR classifier, which is useful for determining a baseline performance as a benchmark for other classification methods. To avoid overfitting or underfitting the model, I imported a library called *TextBlob* (`https://textblob.readthedocs.io/en/dev/`) as the other baseline I used for comparison. With these two tools, a more reliable evaluation can be performed upon analyzing the performance of the model.

4 Preliminary results

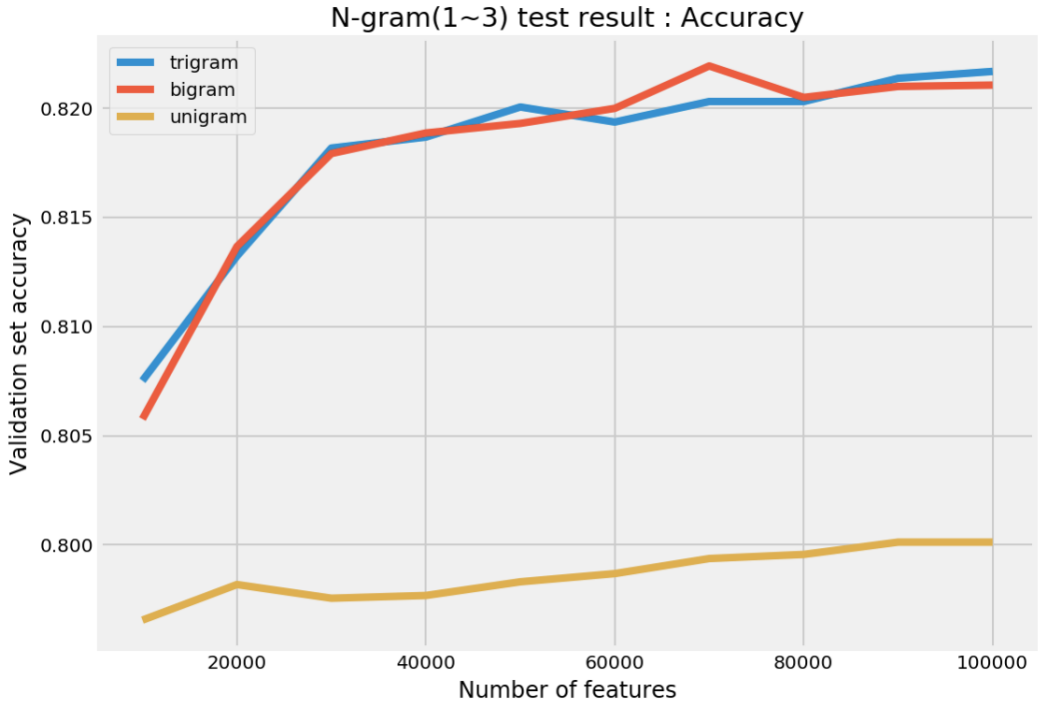
Using the Zero-Rule (ZeroR) classifier, the model reported 49.45% negative in the validation set, which means that if a classifier predicts negative for every validation data, it will get 49.45% accuracy.

```
Train set has total 1564120 entries with 50.02% negative, 49.98% positive
Validation set has total 15960 entries with 49.45% negative, 50.55% positive
Test set has total 15961 entries with 49.68% negative, 50.32% positive
```

Moreover, stop words were introduced during feature extraction. It is often assumed that removing stop words is a necessary step, and will improve the model performance. A custom list of ten common stop words was defined. It was found that By looking at the evaluation result, removing stop words did not improve the model performance, but keeping the stop words yielded better performance in this particular case. Results are shown in the following chart:



Then the accuracy on validation set for the different number of features was checked and was compared under three different conditions (unigram, bigram, and trigram). Results are shown in the following chart:



Importing TextBlob library yielded 60.65% accuracy, which was 11.2% more accurate compared to the null accuracy. F1 score is the harmonic mean of precision and recall. The harmonic mean is a specific type of average, which is used when dealing with averages of units, like rates and ratios. By calculating the harmonic mean of the two metrics, it will give us a good idea of how the model is performing both in terms of precision and recall.

```
Accuracy Score: 60.65%
--------------------------------------------------------------------------------
Confusion Matrix

          predicted_positive  predicted_negative
positive                7094                 822
negative                5458                2586
--------------------------------------------------------------------------------
Classification Report

             precision    recall  f1-score   support

          0       0.76      0.32      0.45      8044
          1       0.57      0.90      0.69      7916

avg / total       0.66      0.61      0.57     15960
```

5 Next steps

So far, the logistic regression model has been proven free of underfitting or overfitting the data, and thus successful in terms of training. However, it was quite time-consuming applying to the dataset. After careful experimenting, convolutional neural network shows a better accuracy and precision compared to logistic regression for the aim of my project. For Deliverable 3, I plan to also train the CNN model for a parallel comparison with the current logistic regression model, hoping to obtain better accuracy. In the

meantime, I will try to visualize the data I've been working with and look into applying my model to more up-to-date datasets.