Sentiment Analyzer

Tanmay Patil

For any technology platform trying to measure social opinion, a reliable sentiment analyzer is necessary for its completeness. In this project we are trying to build visualizations so that the end-user can deduce opinion about the topic he is interested in. Hence, to fulfill this need we have build an automatic sentiment extractor which classifies each tweet into either positive, negative or neutral sentiment.

One of the non-functional requirements of this project was to use Pharo, Moose & Roassal as the technology stack. After consulting with the product owner, Dr. Bergel, we reached to the conclusion that there was no NLP (natural language processing) library in these technologies. Building a natural language processor from scratch was out of the scope of this project and it would also lead to a sub-optimal processor as none of the team members had any expertise in NLP. Hence, we decided to switch to *python* to fulfill this necessity of a good NLP library. Python has a comprehensive library called NLTK (Natural Language Tool Kit) and we decided to make use of this library for our project.

There were multiple approaches by which the analyzer could have been implemented. Most common of them are as follows:

1. Naïve Bayes Classifier
2. Maximum Entropy Classifier
3. Support Vector Machines (SVM)

The sentiment analyzer for TweetViz implements the *Naïve Bayes approach*. This approach was primarily selected because it is easy to understand and implement. The Naïve Bayes algorithm requires *training* i.e. we need to train the algorithm on particular known inputs so that later on the algorithm may use the concept of *conditional probability* to test them on unknown inputs.

The data flow for the sentiment analyzer has been showed in the flowchart (Figure 1). It is explained in detail below:

- The analyzer takes a JSON file of tweets as input.
- Each tweet is pre-processed to a cleaner version so that a more accurate analysis can be deduced. Pre-processing of tweets involves the following actions:
    - ➢ Convert tweet to lower case.
    - ➢ Remove all URLs.
    - ➢ Remove all twitter handle mentions (eg. @super9user).
    - ➢ Remove all additional white spaces and line breaks.
    - ➢ Replace all hashtags with actual words. (eg ‘#ASU will be replaced by just ‘ASU’)

- The analyzer takes an additional input whether new training data is to be built or to use the existing training (if it exists).

- The analyzer uses a feature list (most common words used to describe sentiment) and known inputs (already classified tweets) to combine them and build training data. This training data is then written on file for future use.
- The analyzer then iterates over each tweet and uses the Naïve Bayes approach to classify them into positive, negative or neutral sentiment.
- The output is a JSON file consisting of all the tweets tagged with their sentiment.

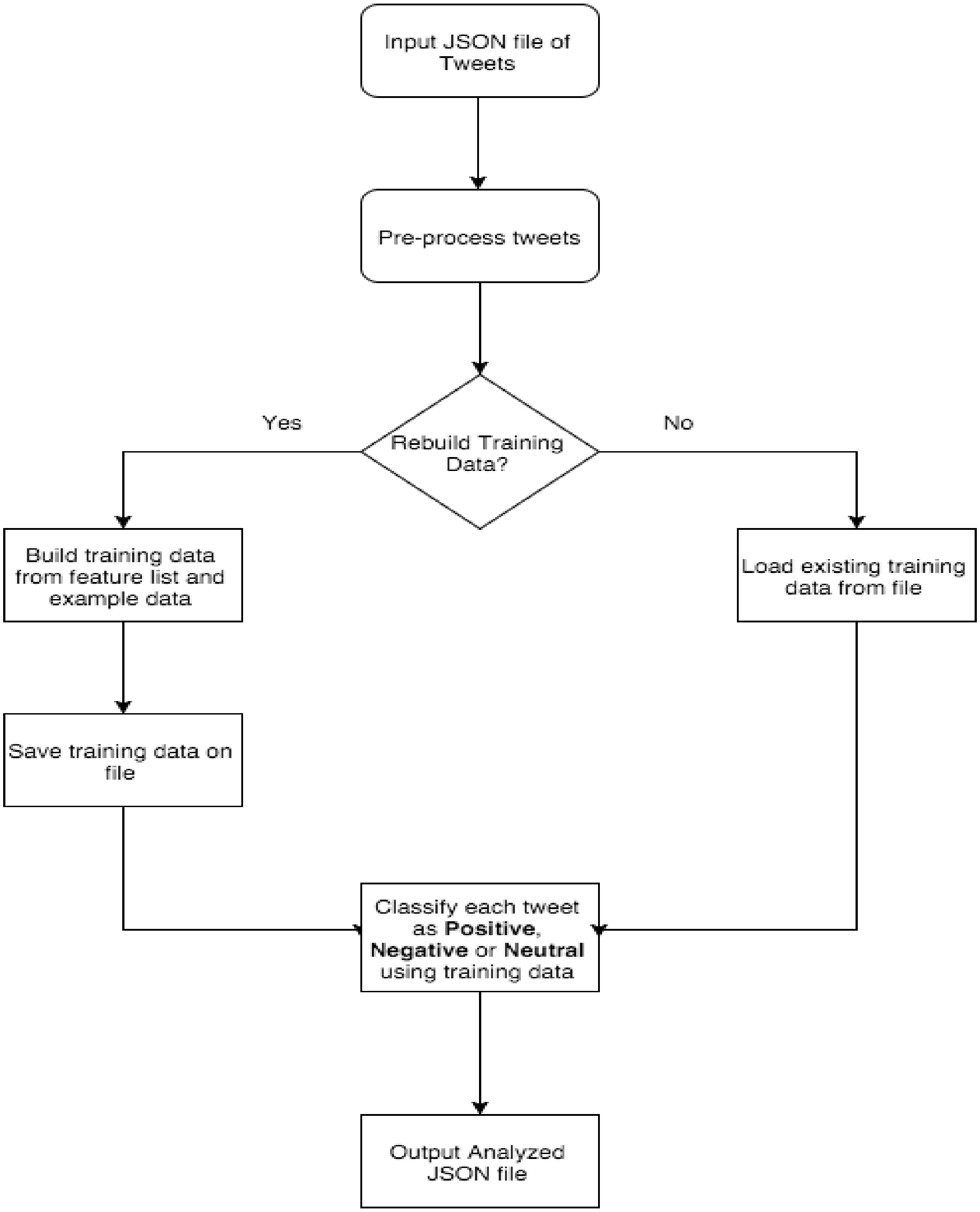
Input JSON file of Tweets
Pre-process tweets
Yes
Rebuild Training Data?
No
Build training data from feature list and example data
Load existing training data from file
Save training data on file
Classify each tweet as **Positive**, **Negative** or **Neutral** using training data
Output Analyzed JSON file

Figure 1.