COMP30027 Assignment 2: Report

Anonymous

1 Introduction

Sentiment analysis is a field of natural language processing focusing on the classification of text by its perceived sentiment. It speeds up measuring the opinions of groups of people and can be applied in many fields such as marketing, politics or sociology. My goal is to develop and train several reliable Twitter sentiment classifiers. This involves developing methods to extract and select useful features from a dataset of posts, then to choose and evaluate highly reliable classifier models.

1.1 Dataset

The given dataset provided contains two lists of Twitter posts (tweets) made on the platform prior to 2017 (Rosenthal et al., 2017). The two files enclosed in the dataset are a `Train.csv` for training and a `Test.csv` for testing. Each tweet is an instance in the dataset. The training set contains 21802 labelled instances and the testing set contains 6099 instances. For each instance, included is the tweet text and tweet ID. The tweets included vary in content. For example, some tweets are not in English: "*season in the sun versi nirvana rancak gak..slow rockkk...*". In the training file is also a column containing the true sentiment of each tweet. Tweets can either have a "positive", "neutral" or "negative" sentiment. the distribution of the sentiments across the training set is shown in Figure 1.

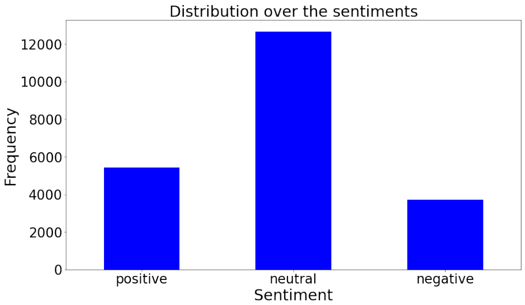

Figure 1: Distribution of the sentiments

2 Methodology

This section contains a breakdown of the process through which I develop the final models chosen in Section ??. The following methods are developed with reference to prior works on Twitter sentiment analysis by Go et al. (2009) and Barbosa and Feng (2010).

2.1 Instance Cleaning

Some of the features extracted rely on the text in the tweets to be pruned of unwanted characters and words. I have opted to generate a separate list containing the cleaned versions of tweets. Cleaning involves removing stopwords (Section 2.1.1), links, tweet hashtags, tweet mentions, numbers, non-alphanumeric characters. Also performed is the reduction of repeated letters with more than two occurrences to just two, as suggested by Go et al. (2009).

2.1.1 Stopwords

Manual stopword list construction is tedious and often inexhaustive (especially if I can only determine English stopwords). Therefore, I have opted to start with the Python Natural Language Toolkit's (`NLTK`) defined stopword list, which includes multiple languages (Bird et al., 2009). However, since I do not want to blindly rely on this list, I generate a set of word clouds over the training cleaned tweets (using the `NLTK` stopword list) grouped by sentiment. This then informs whether I need to manually add a few terms which the used list does not include.

2.2 Feature Extraction

2.2.1 Twitter Features

First, relying on personal usage experience with Twitter, I extract the main platform-specific features from the tweet text. These are:

- **Hashtags** (e.g. `#term`): Used to associate tweets with a certain term on the platform.
- **Mentions** (e.g. `@user`): Used when addressing a specific user on the platform.

- **Links** (e.g. `https://t.co/id`): Used to link to a website with an id on the platform's redirect service.

These are integrated within the Twitter platform, meaning they are widely used by users. Therefore, these features may be strongly correlated to the sentiments and should be isolated for use in the final models.

2.2.2 Linguistic Features

Next, linguistic features are extracted and used to tokenize the tweet in different ways.

- **Part-of-Speech Tags**: Extracting the grammar types of words.
- **Words**: Words used in the tweet.
- **Word 2-Grams**: Word pairs used in the tweet.
-

These feature types may all have certain features which are strongly correlated to a certain sentiment. For example, a positive `big win` versus a negative `big disaster` word pairing.

Neither stemming nor lemmatization are used in the final models as both are too language-dependent. An example for this is the stemming of `bare` to `bar`, removing meaning from the word. The best option, a multilinguistic lemmatizer, requires a neural network and training (Fonseca, 2019).

- **Punctuation** (`.?!,:;-()[]{}"'/`): Certain punctuation such as exclamation marks may indicate a non-neutral tweet, for example.
- **Emoticons** (e.g. `:(` or `:)`): The rise of `ASCII` emoticons allows users to quickly express their emotions, which correlate strongly to the sentiment of their text.

Extracting emoticons is done differently to the method used in the Go et al. (2009) research, since there are emoticons which are not considered, such as the backwards happy `(:` emoticon or the cutesie `:3`. Instead, emoticons are defined as a string comprised of eye characters (`;:8=`), optional middle characters (`,'-"*`), and mouth characters divided into four categories: happy (`)3]`), sad (`/([`), neutral (`pl|`) and surprised (`vo`). There are also defined backwards versions of the happy and sad mouths. On top of the basic emoticons, others that can be detected are: `;3`, `:'o` or `]":`. The detected emoticons are then simplified into one of four emoticons based on their mouth's category: happy `:)`, sad `:(`, neutral `:|` and surprised `:o`.

-

2.2.3 Metric Features

These are largely numeric counts of other features in a tweet.

- **Number of Words**: More words could indicate a more sentimental tweet.
- **Number of Characters**: It may be that longer tweets contain more non-neutral sentiments than shorter ones.
- **Number of Alphabetic Characters**: This will be correlated to more writing, which has a higher chance of being sentimental.
- **Number of Links**: More linking could indicate a less sentimental tweet.
- **Number of Hashtags**: More hashtags could indicate a more sentimental tweet.
- **Number of Mentions**: More mentions could indicate a more sentimental tweet.
- **Number of Emoticons**: More emoticons could indicate a more sentimental tweet.
- **Retweets** (e.g. `"text"`): Whether a post quotes another post on the platform. This form of quoting could be used for argumentation.
- **Average Word Length**: Tweets with longer words may tend in a certain direction.

2.2.4 Vectorization

For all features except the metric features, they can be vectorized by TF-IDF, or occurrence counts. At the time that the data was collected, tweets were limited to 140 characters (Pardes, 2017). This suggests that most features (such as word pairs) are not likely to appear the same time more than once in a tweet (except for stopwords). Therefore, most features are vectorized with raw counts. Certain features that may appear more than once per tweet have their vectorization method chosen using the bar-graph comparison test described in Section 2.3.1.

2.3 Feature Selection

There are many candidates for features to analyse the sentiments of the tweets. Using all of them may result in overfitting of the models and increases the time and space complexity of model construction. To avoid these issues, a certain subset of the features is selected using the following tests.

2.3.1 Bar Graph Comparison Test

One way I use to determine the predictive potential of a feature is through comparitive bar graphs. I generate bar graphs comparing the distributions over the top 10 features in a feature set by the averages of their values, such that:

$$average_{s\subset S} = \frac{\sum f_{s\subset S}}{\sum f_S}$$

where s is a subset of all the sentiments S, f is the vector of values per tweet for a specific feature in a feature type, and N is the number of instances. If too many of the same features appear in all sentiments' bar graphs, then the feature type is removed from the final feature list.

2.3.2

3 Analysis

4 Conclusions

References

Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 36–44, USA. Association for Computational Linguistics.

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Erick Fonseca. 2019. State-of-the-art multilingual lemmatization. *Towards Data Science*, March.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *Processing*, 150, 01.

Arielle Pardes. 2017. A brief history of the ever-expanding tweet. *Wired*.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada, August. Association for Computational Linguistics.