Machine Learning Engineer Nanodegree
Capstone Project

Ashvin Srinivasan

April 4, 2018

1 Domain Background

Extracting the public opinion from social media text provides a challenging and rich context to explore computational models of natural language processing. Which further motivates new research in computational linguistics[3]. The major break throughs were first seen in 2010 when E Junque de Fortuny et. al in 2010 proposed a way to scrap relevant text from internet and perform sentiment analysis. The Belgian elections of 2010 were the topic of study. The body of text that were used for this analysis comprised of news articles published in online versions. A web crawler was built for this purpose[3].

For sentiment analysis, a module consisting of a subjectivity dictionary of over 3000 dutch adjectives occurring frequently were used. Each word was manually given a score of polarity. In this approach, the mention of party is located and polarity count of adjectives is calculated[3].

Sentiment analysis as its name suggests is the art of extracting words from a text that are meant to reflect the sentiment of the text. In our project we use supervised learning model based approach, in which we develop a model that predicts the sentiment of a text. In this day and age, sentiment analysis is used widely in ads marketing, tweet analysis among others. Collecting tweets and analysing about items, personalities is an amazing way to employ sentiment analysis!

2 Problem Statement

The problem comes under sentiment analysis. It is also referred to as opinion mining. The basic problem is trying to understand the attitude of the speaker and categorising it into different moods. The simplest being; 'positive' or 'negative'. The best way to understand the problem is through a simple example. The statement 'I like this movie very much' expresses a positive attitude towards this movie. In essence the speaker is having a positive attitude towards the movie. Where as the statement, ' I found this movie to be cliched and unpleasant' expresses a negative attitude towards the movie. The whole idea of sentiment analysis is based on coming up with models that could automatically identify the sentiments and emotions of the speaker/writer.

Sentiment analysis is widely applicable in today's society. Such as gathering negative and positive reviews of movies, gathering negative and positive tweets about politicians that could help in their political campaign. It could also be used for Ad campaigning. Sentiment analysis is widely used in today's marketing campaigns. For our problem, given an input text, we need to come up with a model that classifies them as either 'positive' and 'negative'. We quantify our model's performance by using the 'accuracy' metric. Our evaluating metric, the 'accuracy' score, is obtained by calculating the percentage of correctly predicted sentiments.

3 Datasets and Inputs

The data used for sentiment analysis in this project is obtained from $'https://pythonprogramming.net/static/downloads/short_reviews/'$. It consists of two text files; one contains text that are all positively classified, while the other contains texts that are termed negative. Our analysis reduces to a supervised classification problem. The inputs are a set of several words as features while the target label assumes two values; 'Positive' or 'Negative'. We have 5300 texts each for positive and negative examples. We shall divide the total sets into training and testing sets, and use training set to train our model and test it out on our testing set.

Each statement will have words constituting different parts of speech. And these words will be used as input features for training our model. Each sentence has different parts of speech. There are a total of 8 parts of speech.

A brief description about these parts of speech is a necessary condition to perform sentiment analysis. The different parts of speech are best explained with examples!

1) Noun: It typically is the name of a person, place, thing or an idea. My name is Ashvin, I received a letter from my teacher. Here Ashvin is a proper noun, letter and teacher are nouns too. A noun generally can be a subject, indirect object[1]

2) Pronoun: It is a word used in place of a noun. For, eg. words like she, we, they, it and so on. Udacity's machine learning nano degree is awesome. It is a must for people learning ML. 'it' is a pronoun.

3) Verb: A verb expresses an action or being. For e.g., 'He brought me a macbook pro', here brought is a verb.

4) Adjective: It describes a noun or a pronoun. For. eg., 'Udacity is an awesome website, it provides an amazing ML nanodegree'. Awesome and amazing are adjectives that describes something about Udacity and MLND.

5) Adverb: It describes a verb. For e.g., Udacity reviewers/mentors swiftly respond to your queries. Swiftly is an adverb.

6) Preposition: it is a word placed before a noun/pronoun to form a phrase modifying another word in the sentence. Words like 'by', 'with', 'about', 'until' are prepositions.

7) Conjunction: A conjunction joins words, phrases and clauses. For e.g., words like 'and', 'but', 'or'.

8) Interjection: An interjection is a word used to express emotion. For e.g., 'Oh!', 'Wow', 'Oops!'

These are the parts of speech typically used in the English language. Understanding the parts of speech helps us to narrow down our features. There is direct relation between the parts of speech and sentiment of a sentence. For eg, an adjective, adverb describes the quality of a noun, verb and thereby contributing more to the sentiments as compared to maybe a preposition. Our data set contains such sentences, and it is our model's job to accurately classify them. Thus, our datasets basically contains words from sentences as our input features and target label is a categorical variable; 'positive' or 'negative' that describes the sentiment of our text, which directly falls into the sentiment analysis paradigm.

4 Solution Statement

A good approach to the problem would be to divide the entire data set into training and testing sets. We use the training set to come up with a model that learns to extract important features(words) and collectively associates them, and classify them into either a 'positive' or a 'negative' statement. For training our model, we can use classification algorithms like logistic regression, Support Vector Classifier(SVC), random forest classifier, and Bayesian classifiers. Concretely, for any given text, we extract features from the text and subject them as inputs to our trained model and output them as 'positive' or 'negative' sentiment. Such kind of target labels are easily measurable and reproducible

5 Benchmark Model

A naive predictor is a good benchmark model that relates to our problem statement. In order to compare our results obtained from training, we can stack it up against a naive predictor. When we have a model that always predicts 'positive' (i.e. the text conveys a positive sentiment) then our model will have no True Negatives(TN) or False Negatives(FN) as we are not making any negative('negative') predictions. Therefore our Accuracy in this case becomes the same as our Precision(True Positives/(True Positives + False Positives)) as every prediction that we have made with 'positive' that should have 'negative' becomes a False Positive; therefore our denominator in this case is the total number of records we have in total[2]. The naive predictor simply predicts a constant value without taking any input features into account, hence the name naive predictor. Mathematically it is given as:

$$accuracy_{naive} = TruePositives/(TruePositives + FalsePositives)$$

By obtaining this naive accuracy, we could then stack it up against the accuracy obtained from our trained model and thereby quantifying our model's performance.

6 Evaluation Metrics

We shall be using two metrics to quantify our model; one is the accuracy and the other is f_β score. We quantify the solution through 'accuracy', which is mathematically defined as:

$$accuracy := sum(I)/n$$

where I is an indicator function=1, if $y = y_{pred}, else 0$

y is the ground truth, y_{pred} is the predicted class from our trained model, and n is the total number of examples in the testing set. Accuracy is a simple yet effective metric to quantify our model's efficacy. Another metric that will be used is f_β score mathematically defined as:

$$f_\beta = (1 + \beta^2) * precision * recall/((\beta^2 * precision) + recall)$$

Where Precision is $TruePositives/(TruePositives+FalsePositives)$ and Recall is $TruePositives/(TruePositives + FalseNegatives)$. For our analysis, beta is typically set to '0.5'. An ideal value would be to have a F-score of 1. So the higher the value of F-score, the better is the model.

7 Project Design

The theoretical workflow could broadly be classified into the following stages:

7.1 Data Preparation

We initially pre process the data since each text consists of many words and the words occurring in every other text may be different from each other. Thus, we shall use the most common occurring words as features. So we first extract words from sentences, calculate the most commonly occurring words. We then randomly shuffle ad split our entire data set into training and testing data sets, for e.g., in the ratio of 80 % as training and 20 % as testing data sets. One of the strategies we could consider in our project is to use only those commonly occurring words that is likely to contribute to the sentiment. For e.g., parts of speech like prepositions are less likely to contribute when compared to adjectives, adverb so we consider only important parts of speech.

7.2 Model Evaluation Performance

We use the naive predictor as a benchmark model in order to evaluate our future trained model. For our trained model, we shall use a class of classifiers, for e.g., multinomial naive Bayes classifier, linear Support Vector Classifier(SVC), Nu SVC, random forest classifier, logistic classifier. We then use majority voting system formally defined as

$$y_{pred} := argmax_{j \in [neg,pos]} \sum_{i=1}^{C} d_{i,j}$$

Which translates to return that class 'j' for which the above expression is maximised, where C is the number of classifiers, in our case 5, and j is the class, negative or positive. Concretely speaking we return that class which is predicted the maximum number of times so if 3 classifiers predict the output as 'positive' while two classifier predicts as 'negative', then output is positive. This method is more preferred because as opposed to using a single classifier as it raises the reliability of our prediction accuracy.

8 References

[1] http://www.butte.edu/departments/cas/tipsheets/grammar/parts_of_speech.html

[2] Udacity tutorial, finding donors assignment

[3] https://sites.google.com/site/adreamsentimentanalysis/home/twitter-sentiment-analysis