IRSW Project Synopsis

Twitter Emotion Analysis

Department of CSE/IT
Jaypee Institute of Information Technology, Noida

Submitted to:

Dr. Neetu Sardana

Submitted By:

Abhinav Modi	**19104035**	**B12**
Rishabh Kaushik	**19104036**	**B12**
Harsh Jain	**19104037**	**B12**
Govind Yadav	**19104038**	**B12**

Introduction

Sentiment analysis, also refers as opinion mining, is a sub machine learning task where we want to determine which is the general sentiment of a given document. Using machine learning techniques and natural language processing we can extract the subjective information of a document and try to classify it according to its polarity such as positive, neutral or negative. It is a really useful analysis since we could possibly determine the overall opinion about a selling objects, or predict stock markets for a given company like, if most people think positive about it, possibly its stock markets will increase, and so on. Sentiment analysis is actually far from being solved since the language is very complex (objectivity/subjectivity, negation, vocabulary, grammar,...) but it is also why it is very interesting to work on.

In this project we choose to try to classify tweets from Twitter into "positive" or "negative" sentiment by building a model based on probabilities. Twitter is a microblogging website where people can share their feelings quickly and spontaneously by sending a tweets limited by 140 characters. You can directly address a tweet to someone by adding the target sign "@" or participate to a topic by adding an hashtag "#" to your tweet. Because of the usage of Twitter, it is a perfect source of data to determine the current overall opinion about anything.

Data

To gather the data many options are possible. In some previous paper researches, they built a program to collect automatically a corpus of tweets based on two classes, "positive" and "negative", by querying Twitter with two type of emoticons:

- Happy emoticons, such as ":)", ":P", ":)" etc.

- Sad emoticons, such as ":(", ":'(", "=(".

Others make their own dataset of tweets by collecting and annotating them manually which is very long and fastidious. Additionally to find a way of getting a corpus of tweets, we need to take off having a balanced data set, meaning we should have an equal number of positive and negative tweets, but it also needs to be large enough. Indeed, the more data we have, the more we can train our classifier and the more accurate it will be.

Technologies Used

❖ Pre-processing :

From the corpus of tweets and all the resources that could be useful, we can preprocess the tweets. It is very important since all the modifications that we are going to during this process will directly impact the classifier’s performance. The pre-processing includes cleaning, normalization, transformation, feature extraction and selection, etc. The result of pre-processing will be consistent and uniform data that are workable to maximize the classifier's performance.

❖ Machine Learning :

Once we have applied the different steps of the preprocessing part, we can now focus on the machine learning part. There are three major models used in sentiment analysis to classify a sentence into positive or negative: SVM, Naive Bayes and Language Models (NGram). SVM is known to be the model giving the best results but in this project we focus only on probabilistic models that are Naive Bayes and Language Models that have been widely used in this field. Let’s first introduce the Naive Bayes model which is well known for its simplicity and efficiency for text classification.

❖ Training & Testing of our Model :

Once the training set and the test set are created we actually need a third set of data called the validation set. It is really useful because it will be used to validate our model against unseen data and tune the possible parametersof the learning algorithm to avoid underfitting and overfitting for example. We need this validation set because our test set should be used only to verify how well the model will generalize. If we use the test set rather than the validation set, our model could be overly optimistic and twist the results.

To make the validation set, there are two main options:

● Split the training set into two parts (60%, 20%) with a ratio 2:8 where each part contains an equal distribution of example types. We train the classifier with the largest part, and make predictions with the smaller one to validate the model. This technique works well but has the disadvantage of our classifier not getting trained and validated on all examples in the data set (without counting the test set).

● The K Fold cross validation. We split the data set into k parts, hold out one, combine the others and train on them, then validate against the heldout portion. We repeat that process k times (each fold), holding out a different portion each time. Then we average the score measured for each fold to get a more accurate estimation of our model's performance.

Conclusion

Nowadays, sentiment analysis or opinion mining is a hot topic in machine learning. We are still far from detecting the sentiments of the corpus of texts very accurately because of the complexity in the English language and even more if we consider other languages such as Chinese.

In this project we tried to show the basic way of classifying tweets into positive or negative categories using Naive Bayes as a baseline and how language models are related to the Naive Bayes and can produce better results. We could further improve our classifier by trying to extract more features from the tweets, trying different kinds of features, tuning the parameters of the naïve Bayes classifier, or trying another classifier all together.