Sentiment Analysis

Ansh Bordia

1 Introduction

Sentiment Analysis is the task of categorizing a piece of text in order to determine whether the writer has a positive, negative, or neutral attitude towards it. This task is extremely useful as it helps understand the wider public opinion behind different topics, which we can use to reap many benefits.

In this report, we aim to discover if tweets can help us identify people sentiment on Twitter. We begin by discussing which supervised Machine Learning methods would be well suited for this task, apply them, and use the best performer to validate our hypothesis.

2 Dataset

The dataset, curated by [1], consists of unfiltered tweets divided into training, validation and testing sets, with three classes - positive, negative and neutral. We have an approximate total of 33000 tweets, divided approximately in a 70:15:15 ratio between training, validation and testing respectively.

3 Related Literature

The problem of sentiment analysis has been addressed out by several researchers. An early example includes work done by [2]. They analyze the results of using supervised methods like Naïve Bayes, MaxEnt and Support Vector Machines on tweets; concluding SVM outperforms the others. In addition, they conclude a unigram model works best, and describe the pre-processing steps required to achieve maximum accuracy. More recent work done by [3] describes a tree kernel and new features, which are concluded to perform better than baseline measures.

4 Evaluation Metrics

We will use the following terms throughout to evaluate the proposed systems:

- Accuracy – The proportion of correctly classified tweets.
- Precision – For each class, the ratio of correctly classified positives over all the classified positives.
- Recall – For each class, the proportion of correct positives identified

4 Methodology

We use the tool Weka, developed by University of Waikato, New Zealand for the task at hand. The tool is chosen as it provides a diverse selection of classifiers, feature selection methods, evaluation metrics and its use is fairly straightforward for our purposes.

4.1 Supervised method selection

While there exist many supervised classifiers, not all are suitable for the task of sentiment analysis. The results obtained in [2], [4] and [5] show the SVM and Naïve Bayes methods work well for our task. The research done in these papers tells us text classification and, in particular, sentiment analysis problems, have sparse document vectors, few irrelevant features, and have high dimensional input space which makes SVM and Naïve Bayes good fits for classification. In addition, they are also shown to rarely overfit the data.

4.2 Pre-processing

Pre-processing is an important step as a classifier is only as good as the quality of the data. Our dataset consists of unfiltered and raw tweets, which means pre-processing is of paramount importance if we are to get encouraging results. Stemming is applied and stop words are removed. Tweets are tokenized into unigrams (words) and converted to lower case. Special characters, other than '!' – which can give information about the reader's tone – are removed. Hyperlinks are removed as well. We make use of the StringToWordVector class here. Instead of using only the 46 features provided in the data files, we use 200 features. Weka gives an option to do this, and it selects the 200 most important features. Since we have a very high frequency of unique words, it is a good idea to have more features to get a better accuracy. It is worth noting that we do stop word removal and stemming before feature engineering in order to leave out insignificant words like 'the', 'to', etc. that usually occur in high frequencies, and to also avoid dealing with different forms of words that essentially impart the same meaning, like 'love', 'lovingly', 'lovely', etc.

4.3 Naïve Bayes

Naïve Bayes is a probabilistic classifier that makes classifications using the Maximum A Posteriori decision rule in a Bayesian setting. We train the model, get the training accuracy and then use the validation set to test its performance.

4.3.1 Results

4.3.1.1 Training Results

We obtain the following results:

Accuracy	62%

Table 1: Accuracy from Naïve Bayes Training.

	Precision	**Recall**
Negative	53%	72%
Positive	57%	73%
Neutral	73%	51%

Table 2: Precision/Recall of each class from Naïve Bayes Training.

4.3.1.2 Validation Results

Evaluating the model on the validation set we get the following results:

Accuracy	42%

Table 3: Accuracy from Naïve Bayes Validation.

	Precision	**Recall**
Negative	27%	23%
Positive	35%	29%
Neutral	50%	60%

Table 4: Precision/Recall of each class from Naïve Bayes Validation.

4.3.2 Analysis

The results obtained highlight the need for improvement in the pre-processing and feature engineering process.

We investigated the predictions to get an idea where the classifier is performing well and where it is not. Upon investigating, it was found that the classifier performed very well in differentiating positives and negatives but performed poorly in differentiating positives and negatives to neutral. The table below lists out where the classifier fails to perform.

Actual	**Prediction**
Positive	Neutral
Negative	Neutral

Table 5: Analyzing Weakness

Further analysis shows us that tweets with majority of their text as hyperlinks, hashtags or stop-words are more prone to being misclassified. We link this finding to the pre-processing stage where the useless text in these tweets were wiped out, meaning the classifier will most likely not have enough words to make a correct prediction. Some tweets with proper nouns are also misclassified. We can attribute this to the fact that proper nouns don't really add to the sentiment of a sentence. Examples include:

Be the 1st to check out our new look at https://t.co/skWcZNaN9Z! Pl give ur feedback. #Ayurveda #Homeopathy https://t.co/rIUHwYruHP
'he had a good life and a wonderful wife.. may Frank Gifford RIP https://t.co/B0T5KfD3ir',positive

Table 6: Analyzing structural weakness

4.4 Support Vector Machine (SVM)

SVM is a discriminative classifier wherein it outputs an optimal hyperplane which categorizes new examples. The method is shown to work relatively well on text classification tasks [5], and as sentiment analysis is a subset of that we expect favourable results as well.

4.4.1 Results

4.4.1.1 Training Results

We get the following results:

Accuracy	72%

Table 7: Accuracy from SVM training

	Precision	**Recall**
Negative	74%	58%
Positive	76%	61%

Neutral	70%	84%

Table 8: Precision/Recall of each class from SVM training

4.4.1.2 Validation Results

Evaluating the model on the validation set we get the following results:

Accuracy	45%

Table 9: Accuracy from SVM validation

	Precision	**Recall**
Negative	28%	21%
Positive	36%	26%
Neutral	51%	65%

Table 10: Precision/Recall of each class from SVM validation.

4.4.2 Analysis

We see the SVM model performs better than Naïve Bayes. The better performance is the result of SVM handling sparse word vectors better. Upon analysis, we find that the classifier suffers from the same problems as that faced by Naïve Bayes.

5 Conclusions

We choose the SVM model to predict labels for the test set as it performs the best.
Although the method does not perform to the level of the benchmarks [4], we are still able to get knowledge from this task i.e. a writer's sentiment towards a certain issue, which we can accordingly use for a wide range of tasks. Thus, we can say our hypothesis is correct; we do get knowledge. Further, the results give us the knowledge that correctly identifying actual neutral sentiments is a much easier task. From this task, we also establish that pre-processing of the tweets and feature engineering are very important to get high quality results.

7 References

[1] Rosenthal, Sara, 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 '17). Vancouver, Canada.

[2] Go, Alec & Bhayani, Richa & Huang, Lei. (2009). Twitter sentiment classification using distant supervision. Processing. 150. [4] Hall, P. and Dowling, G. (1980). Approximate string matching. *Computing Surveys*, 12(4):381–402.

[3] Agarwal, Apoorv & Xie, Boyi & Vovsha, Ilia & Rambow, Owen & Passonneau, Rebecca. (2011). Sentiment Analysis of Twitter Data. *Proceedings of the Workshop on Languages in Social Media.*

[4] Vimalkumar B Vaghela and Bhumika M Jadav. Article: Analysis of Various Sentiment Classification Techniques. *International Journal of Computer Applications* 140(3):22-27, April 2016. Published by Foundation of Computer Science (FCS), NY, USA.

[5] Joachims, Thorsten & Ls, Informatik & Str, Baroper. (1999). Text Categorization with Support Vector Machines: Learning with Many Relevant Features.