Knowledge Technologies Report

Juan Daniel Daza

Index Terms **– Twitter, Sentiment Analysis, Classifier, Machine Learning**

I. Introduction

Twitter is a microblogging platform where millions of users can express and share news, their opinions and thoughts using less than 140 characters. Using these data, researchers and firms can analyze users' behavior by various means one of which is sentiment analysis. The purpose of this project is to understand the application of Machine Learning in the task of determining the sentiment of tweets and finding patters in the dataset provided in the 2017 SemEval conference [1] which contains around 33k tweets.

The dataset has been split into three different subsets, one for training, one for development and one for testing. For the training and development sets we have the sentiment labels (positive, neutral or negative) of the thousands of tweets there contained. In addition to this, an initial feature selection has been made where the methods of mutual-information and Pearson's – squared test were applied to determine the best correlation between tokens and classes. Overall, we have 46 different classes.

II. Background and Related Research

Twitter has captured the interest of many researchers due to the availability and the openness of the data. Many different feature selection techniques have been studied along with different Machine Learning models to try and predict sentiment in tweets. In [2], the authors not only discuss the challenges of dealing with Twitter data, but propose a technique to deal with streaming unbalanced classes. In [3], the authors achieve, using various Machine Learning algorithms accuracy of over 80% by using emoticons in the training data. Similarly, [4] explores the idea of using linguistic features to detect sentiment of Twitter messages. Finally, [5] explores the different behavior of Machine Learning models when selecting n-grams as the input features. The paper concludes that using bigrams is more effective that uni-grams or tri-grams since it provides a balance between coverage and expression patterns.

The rest of the report is structured in the following way. The effectiveness and analysis behind the result of applying three different Machine Learning algorithms: Naïve Bayes (NB), Decision Tree (DT) and Random Forest (RF) is showed. The, the results using the test dataset are presented and finally the conclusions are shown.

III. Effectiveness Discussion

The following section covers the results and analysis performed on different Machine Learning based on the dataset provided. In every case, the *id* feature was removed from the dataset since it provides no value.

A. Naïve Bayes

As stated above, the Naive based algorithm was run without using the id column. The following table summarizes the performance.

Indicator	**Class**		
	Neg	**Neu**	**Pos**
False Positive	274	1,104	817
False Negative	726	855	614
True Positive	312	1,545	874
True Negative	3,614	1,422	2,621
True Positive Rate	30%	64%	59%
True Negative Rate	93%	56%	76%
Positive Predictive Value	53%	58%	52%
Negative Predictive Value	83%	62%	81%

False Positive Rate	7%	44%	24%
False Negative Rate	70%	36%	41%
False Discovery Rate	47%	42%	48%
Accuracy	80%	60%	71%

Table 1. Naive Bayes Metrics

From the table above we can conclude that Naïve Bayes' algorithm has trouble predicting correctly Neutral values. The given development set is an unbalanced one with 49% of the instances corresponding to the Neutral class. As we can see, the False Positives and consequently the false positive rate is high for the Neutral class, this means that the algorithm predicts Neutral more than it is supposed to.

We can also observe that the True Negative rate and the number of True Negatives predicted have high values which also coincides with the proportion of negative labels in the development set shown in the table below

Class	Count	Percentage
positive	1,488	30%
neutral	2,400	49%
negative	1,038	21%

Table 2. Class Propositions Development Dataset

One particular reason why NB has trouble predicting correctly positive or negative sentiments is the fact that its underlying assumption is independence between features. This assumption does not hold in this case since to construct a sentence, the words are dependent. The table below shows examples where the assumption does not hold and causes the algorithm to predict incorrectly.

ID	Sentiment	Prediction	Keywords	
669918	positive	neutral	I	
802213	positive	neutral	Love	Trump
805674	negative	neutral	a	

Table 3. Instance Keyword Analysis

From the table above it is possible to observe that the main issue with this model is the lack of coverage in the feature selection. The first instance has the keyword *"I"* yet we know that it usually comes accompanied with different words which would express the meaning of the sentence. Similarly, the second instance in the table shows both the word Love and Trump. Given that only these two words appear and have the probabilities shown in Table 4, the algorithm cannot detect the correct sentiment. *Love* has a higher positive probability while *Trump* has a higher negative probability, yet the actual meaning of the sentence is the **Love for Trump** which is why NB cannot predict correctly the sentences as it treats both independently. The lack of coverage and the fact that words are highly correlated with one another causes NB to underperform.

Keyword	Positive	Neutral	Negative
Love	0.0586	0.0072	0.0053
Trump	0.0062	0.0295	0.0685

Table 4. Keyword Mean NB

B. Decision Trees

Similar to NB, decision tree was created with the "*Id"* column removed. In addition to this to avoid overfitting, an optimized Tree Depth parameter was found and is illustrated in the following graph.

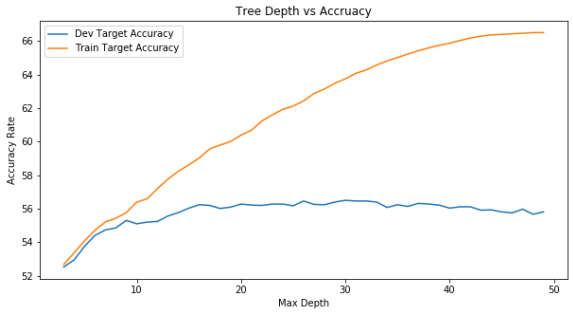

Table 5. Tree Depth Value

As it can be observed, if we select a high value of Tree Depth we increase the overall accuracy of the model but we will overfit the model to the data. This effect can be observed in the difference of curves between the Development set and the

Train set. Given this, a maximum depth parameter of 10 was selected.

	Class		
Indicator	**Neg**	**Neu**	**Pos**
False Positive	72	1857	285
False Negative	975	255	984
True Positive	63	2145	504
True Negative	3816	669	3153
True Positive Rate	6%	89%	34%
True Negative Rate	98%	26%	92%
Positive Predictive Value	46%	54%	64%
Negative Predictive Value	80%	72%	76%
False Positive Rate	2%	74%	8%
False Negative Rate	94%	11%	66%
False Discovery Rate	54%	46%	36%
Accuracy	79%	57%	74%

Table 6. Decision Tree Main Metrics

From the table above, we can see clearly the difference in performance between NB and DT. DT has a lower number of False Positives and a higher number of False Negatives which means that it is more biased towards the Neutral category in which we have a higher number of False Positives. The accuracy for Neutral class decreases compared to NB due to the bias towards this class as more instances are labeled as Neutral. We can see this effect in the rise of False Positives. Additionally, the number of True Positives for both Negative and Neutral class decreases considerably which is an indication of the troubles the algorithm has to classify correctly these classes.

The pitfall when classifying the sentiment in the Tweets is the fact that DT assumes that the sentiment can be described by a logical set of decisions based on the present features. This means that for instances which have as a feature one letter, such as "*I"*, we can determine the sentiment.

C. Random Forest

The final method to be analyzed is Random Forest. Using the same methodology as with NB, we deleted the "*Id"* column and the results are shown in the following table.

	Class		
Indicator	**Neg**	**Neu**	**Pos**
False Positive	223	1593	293
False Negative	794	380	935
True Positive	244	2020	553
True Negative	3665	933	3145
True Positive Rate	24%	84%	37%
True Negative Rate	94%	37%	91%
Positive Predictive Value	52%	56%	65%
Negative Predictive Value	82%	71%	77%
False Positive Rate	6%	63%	9%
False Negative Rate	76%	16%	63%
False Discovery Rate	48%	44%	35%
Accuracy	79%	60%	75%

Table 7. Random Forest Metrics

We can see the overall improvement in performance across all classes when compared to DT. The algorithm is less biased towards Neutral class and performs best at classifying Positives sentiment. The randomizing effect in RF clearly removes some bias which is why the True Positive rate rises when compared to DT.

IV. Test Dataset

The table below shows the results after running the different algorithms on the *test* dataset.

Method	**Positive**	**Negative**	**Neutral**
Naive Bayes	1,646	617	2,663
Decision Trees	747	137	4,042
Random Forest	940	940	1,343

Table 8. Test Dataset Results

The results show how the Neutral class has most significant importance and how all three methods behave differently. Depending on the context of the data and problem-specific domain one algorithm might be more suitable than other.

V. Conclusions

Sentiment Analysis is a challenging task which requires understanding of the underlying assumptions of the methods used to be able to critically review the quality of the output. The data used to train the algorithms has direct implications on the quality of the method and the possible biases that can arise.

Naïve Bayes is a simple yet powerful algorithm which can be used as a baseline for different classification methods comparisons.

There are multiple metrics which are required to be analyzed holistically. No indicator in isolation provides enough feedback to determine the correctness of a Machine Learning implementation.

In general, Machine Learning techniques are dependent on the task at hand however, each task, as in this case the sentiment prediction, needs to be tuned individually to achieve the best possible results.

VI. References

[1] S. Rosenthal, N. Farra and P. Nakov, "Sentiment Analysis in Twitter," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval '17).*, Vancouver, Canada, 2017.

[2] A. Bifet and E. Frank, "Sentiment Knowledge Discovery in Twitter Streaming Data," in *Proc. of 13th International*, 2010.

[3] A. Go, R. Bhayani and L. Huang, "Twitter Sentiment Classification using Distant Supervision," *CS224N Project Report*, 2009.

[4] E. Kouloumpis, T. Wilson and J. Moore, "Twitter Sentiment Analysis: The Good the Bad and the OMG!," in *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[5] A. Pak and P. Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining," *(European Language Resources Association*, 2010.