Technology Review

John Armgardt

November 7, 2021

1 Introduction

NLTK is one of the most popular text information tools used by engineers today. Built for Python, it hosts a myriad of functionalities used for all sorts of subsets of text processing [1]. However, while NLTK boasts about being a ”leading platform” with ”easy-to-use interfaces”, just how good of a tool is it really? Some of NLTK’s most popular functionalities are the Valence Aware Dictionary and Sentiment Reasoner (VADER), a built-in pre-trained sentiment analysis model [2], and the several classification models that use machine learning to classify objects given a feature set. In this paper, I will be simultaneously analyzing the efficiency of the classification models and the accuracy of VADER, to evaluate how good NLTK is.

2 Accuracy

To first evaluate NLTK, we will look at the accuracy of VADER. To do this, we will use sentences of text data pre-labeled as positive or negative to feed into VADER, and have VADER evaluate how positive/negative it thinks the sentences are. Then, using these values as features, we will train classification models that come with NLTK and evaluate the accuracy with a set of test data afterwards. If VADER is accurate, the classification models should return high accuracy.
NLTK comes with three different classification models out of the box: Naive Bayes (NB), Decision Tree (DT), and Max Entropy (ME). We will use these three to evaluate the accuracy of VADER. Furthermore, NLTK provides users with several sets of data that can be used for testing. We will use three sets of data for our testing: Twitter samples, a list of pros/cons, and sentence polarity. Each data point is pre-labeled as positive or negative. We will have each sentence/tweet in the data sets fed into VADER, which will return numeric values of the sentence for the positivity, negativity, neutrality, and likelihood of the text being compound. 90% of these values will then be fed into each of the classification models as features for training, and the remaining 10% will be used for testing. The results are shown below.

	NB	DT	ME
Twitter	0.874	0.820	0.865
Pros/Cons	0.788	0.773	0.786
Sent. Pol	0.629	0.562	0.582

Table 1: Accuracy of each classification model under different data sets

From these results, we can see that VADER is fairly accurate under certain conditions. VADER is designed to analyze sentiment under a few sentences on social media posts, so it makes sense that it is able to come up with accurate labels for the Twitter data set. It is also notable that Naive Bayes outperforms the other models on all of the tests, but this makes sense given it thrives on numeric features, like the ones inputted in these tests.

It is important to note that VADER performs well on a few sentences, and does not perform well on longer texts like paragraphs. The data used to test was of a format with a few sentences per data point. However, results would likely widely vary under different conditions. As such, with an accuracy as low as 0.629 under the best training model, there is more to be desired out of VADER.

3 Efficiency

VADER and the classification models may be able to evaluate sentiment analysis to a certain degree, but what's also important is how efficient they are. As data sets scale, training classification models can take a long time, so it is important to implement these models efficiently as to not waste time. To evaluate efficiency on NLTK, we will time (in seconds) how long it takes for each model to fully intake the training data and fully train the model. This will be done under 5 iterations for each model, and the average time will be taken.

The results are shown below.

	NB	DT	ME
Twitter	0.044	2.457	66.253
Pros/Cons	0.192	13.659	309.901
Sent. Pol	0.053	4.232	72.921

Table 2: Training time of each classification model under different data sets

From these results, we can see a large variance amongst the models training time. First off, it is important to note that Max Entropy undergoes 100 iterations (by default) of training before being able to be tested. As such, it has a much longer training time. Furthermore, Decision Tree must construct the tree data structure it uses, increasing its training time as well.

As mentioned before, the feature set is a set of numeric values that are (theoretically) independent of each other, which Naive Bayes thrives on and the others do not. As such, to see Naive Bayes perform as quickly as it does is satisfactory to see. While the others may be slower to train, their performance was still comparable to Naive Bayes. Under more favorable data sets, their training time would be more forgivable with their higher accuracy.

4 Conclusion

Overall, the classification models provided in NLTK are notably efficient. The models and algorithms themselves may have a lot of complex steps involved, thus adding on to the training time. The implementations themselves, however, appear to be fast. Furthermore, despite being under less favorable features for the slower classification models, their performance was still on par with Naive Bayes, the best performing model. Nonetheless, the accuracy performed lower than desirable. Under ideal conditions, being short social media posts, VADER outputted features that resulted in an accuracy as high as 0.874, and as low as 0.562. While this is not poor performing, there is still more to be desired out of VADER as a professional sentiment intensity analyzer. As such, we can conclude that NLTK is efficient, but not as accurate as hoped.

References

[1] *Natural Language Toolkit.* URL: `https://www.nltk.org/index.html`.

[2] C.J. Hutto and Eric Gilbert. “VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text”. In: *Proceedings of the International AAAI Conference on Web and Social Media* 8.1 (May 2014).