SENTIMENT CLASSIFICATION FOR TWEETS ON NUCLEAR ENERGY USING THE NAÏVE BAYES ALGORITHM

CMSC 191 - MACHINE LEARNING

Harold R. Mansilla
Department of Physical Sciences and Mathematics
College of Arts and Sciences
University of the Philippines Manila
hrmansilla@up.edu.ph

Virgilio M. Mendoza III
Department of Physical Sciences and Mathematics
College of Arts and Sciences
University of the Philippines Manila
vmmendoza1@up.edu.ph

March 4, 2019

ABSTRACT

With the rising concerns over the use of nuclear energy as a source of energy, judges have expressed their thoughts and opinions over the matter through social media such as Twitter. This paper analyzes a dataset containing such tweets and makes use of a Naïve Bayes classifier to do so. The study resulted with test scores between 55% to 84% accuracy with a mean score of 60%. The researchers have recommend to collect more data for underrepresented classes for a more accurate model.

***K*eywords** Naïve Bayes · Sentiment Classification · Twitter · Nuclear Energy

1 Introduction

1.1 The Naïve Bayes Learning Algorithm

The Naïve Bayes learning algorithm is a simple technique for the creation of classifiers, models capable of assigning class labels, obtained from a finite data set, to data instances represented as a feature vector. The class label, C_i, given to the instance, X, is the class which has the highest probability given the probabilities of classes and the data of the instance. To compute for the said probability, the Bayes' Theorem is given as:

$$P(C_i \mid X) = \frac{P(X \mid C_i)\, P(C_i)}{P(X)}$$

The theorem takes on the assumption that each attribute of a class is independent. This assumption simplifies computing for $P(X \mid C_i)$ since it can now be written as

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$

where x_k is a component of X. This makes resulting classifier less computationally expensive when compared to the formula without the assumption.

1.2 Sentiment Classification

Li et al (2010) described sentiment classification as "a special task of text classification whose objective is to classify a text according to the sentimental polarities of opinions it contains, eg., favorable or unfavorable, positive or negative" [1].

Over the years, many online groups and sites host reviews and discussions of certain topics. Labeling these articles using sentiment classification would provide readers with summaries that provide sufficient information. Recommender systems and business intelligence applications can also make use of sentiment classification. Lastly, the classification can also be applied to message filtering systems [2].

2 The Dataset and Preprocessing

2.1 The Dataset

The dataset was retrieved from figure eight's "Data For Everyone" page, a collection of public datasets collected and uploaded by various users (https://www.figure-eight.com/data-for-everyone/). The dataset is entitled 'Judge emotions about nuclear energy from Twitter'. Originally, it contains the tweet, the sentiment, and 'an estimation of the crowds' confidence that each category is correct'.

2.2 Preprocessing the Dataset

The Python libraries `pandas` and `nltk` were used for preprocessing.

The sentiment confidence summary column was dropped leaving only the actual tweet and the sentiment classification it belongs to.

Table 1: Tweets classified per sentiment

Sentiment	**Count**
Negative	19
Positive	10
Neutral / author is just sharing information	160
Tweet NOT related to nuclear energy	1

Table 1 shows the number of tweets per sentiment in the dataset.

The sentiment classifications were re-encoded into numbers (Table 2)

The actual tweets were then stripped off their punctuation marks and transformed into lower case. This is followed by splitting the tweet into a list of words in a process known as *tokenization*. Lastly, common stop words were removed from the lists such as 'a' and 'the' by comparing the lists to `nltk`'s `stopwords corpus` and removing words found between the lists and the stopwords.

Table 2: Re-encoding the sentiment categories

Sentiment Category	**Re-encoded Value**
Negative	0
Positive	1
Neutral / author is just sharing information	2
Tweet NOT related to nuclear energy	3

3 Bayesian Classifier Construction

The Naïve Bayes classifiers were made from scratch using Python. The authors also included implementation of Laplace smoothing, which is simply appended into the original Naïve Bayes formula as an alpha value for calculating the posterior probabilities. With this, Laplace smoothing will be used in training the classifier (alpha = 1). For comparison purposes, a reduced vocabulary will be used along with the unaltered vocabulary. The reduced vocabulary was obtained by removing words with a frequency (count) of less than three at each class. Intuitively, this means that a word may or may not be present in all four classes.

An 80-20 train-test split was imposed on the dataset. For model evaluation, it will be calculated simply by getting the number of correctly classified test cases over the total number of test cases in the dataset.

4 Results

For the classifier with Laplace smoothing on the unaltered vocabulary, 808 words were used for training. This classifier resulted in an 84.21053% accuracy on the test set. Meanwhile, for the classifier on the reduced vocabulary, with 61 words, the accuracy obtained on the test set was the same at 84.21053%.

It was demonstrated that a reduction of vocabulary did not affect the accuracy of the model in this classification problem. For purposes of comparison, another classifier instead without Laplace smoothing was created. It resulted in a 23.68421% accuracy on the test set.

This performance gap in the models may be attributable to the dataset itself. The few number of entries as well as the unbalanced distribution among classes may have lead to the advantage of using Laplace smoothing for training. Without it, we are left with a very limited vocabulary nonetheless and even with a small frequency in each class. The 88% accuracy posted can be expected to be not reliable entirely, as model underfitting for a new set of data may be observed. With this, improvements lie on collecting more data on underrepresented classes in order to have a balanced enough dataset for building and evaluating the models.

References

[1] S. Li, S. Y. M. Lee, Y. Chen, C.-R. Huang, and G. Zhou, “Sentiment classification and polarity shifting,” in *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 635–643, Association for Computational Linguistics, 2010.

[2] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: sentiment classification using machine learning techniques,” in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79–86, Association for Computational Linguistics, 2002.