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

February 21, 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.

***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

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'.

2.0.1 Preprocessing and 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

For building the classifiers, the Python machine learning library `scikit-learn` [3] will be used. Implementing the Naïve Bayes algorithm as well as Laplace smoothing is available in the class `sklearn.naive_bayes.MultinomialNB`. Multinomial Naïve Bayes is the selected implementation as it is recommended by [3] for text classification problems. For this paper, 10-fold cross validation will be performed to evaluate the model's performance on the dataset. The model will be built on the entire dataset.

4 Results

The `scikit-learn` function `cross_validate` provides a comprehensive report of cross-validation which includes model fitting time, test scores, and training scores [3].

Table 3: Results of 10-fold Cross Validation

Iteration	fit_time	score_time	test_score	train_score
0	0.004986286163330078	0.0009968280792236328	0.55	0.9235294117647059
1	0.0039904117584228516	0.001996755599975586	0.6842105263157895	0.9298245614035088
2	0.003988742828369141	0.0009980201721191406	0.631578947368421	0.9239766081871345
3	0.003987789154052734	0.000997781753540039	0.5789473684210527	0.9473684210526315
4	0.003993034362792969	0.0009946823120117188	0.6842105263157895	0.935672514619883
5	0.00899505615234375	0.0019769668579101562	0.5789473684210527	0.9239766081871345
6	0.004988193511962891	0.0009969949722290039	0.5789473684210527	0.935672514619883
7	0.011967897415161133	0.0010001659393310547	0.631578947368421	0.9298245614035088
8	0.004988908767700195	0.0009946823120117188	0.8421052631578947	0.935672514619883
9	0.004984855651855469	0.0009965896606445312	0.5555555555555556	0.9302325581395349

In running 10-fold cross-validation, a warning was raised in which the least populated class (Tweet NOT related to nuclear energy) has a number of members less than the number of folds. Intuitively, this means that this member, at any iteration, will only be either at the training OR test set.

The cross-validation models were built rather quickly, possibly attributable to the dataset size.

The test scores lie in between 55% to 84% accuracy with a mean score of 60%. This can be considered poor to below average performance. The dataset may be culpable for this instability. It is skewed towards one class while the others are very underrepresented, as is the case with the "Tweet NOT related to nuclear energy". The remedy to this is clear, collect more data on underrepresented classes in order to have a balanced enough dataset for building and evaluating a model.

Finally, on the preprocessing and model-building aspects, different approaches may be worth to explore in this lack of dataset members.

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.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.