Lameck Onyango Oyare
MSc. Software Engineering

Reg.SCT313 – 3506/2017

ICS 3101: Research Methodology

Continuous Assessment

Implementing an empathic agent to detect textual Cyberbullying

4.0 Methodology

In order to detect cyberbullying I Implemented Machine Learning techniques to find out which was the most efficient and accurate one. The training dataset I used for the experiment was obtained from Sentiment140 dataset (A., Bhayani, R. and Huang, L., 2009) with 1.6 million tweets. I only selected a random sample of 1,000 tweets labeled positive and another set of 1,000 tweets labeled negative. The sample size selected was sufficient to carry out this experiment and draw a conclusion.

Sentiment classification has two approaches where the tweets are converted into binary for easy detection of whether they are either positive or negative. Binary classification is important when you want to compare two products. In this study implementation was done with respect to binary classification.

The dataset on sentiment140 tweets has been studied and the repository is stored in unstructured textual format. The data needs to be converted into meaningful order for the machine learning algorithms to be applied. The processing will involve the removal of unnecessary information, blank spaces etc. The processed data will be converted to numerical vectors, where each of the vectors corresponds to a review and entries of each vector represent the presence of feature in that particular review.

The conversion of textual data into vector is done using the following methodologies;

- Counter Vectorizer:
- Term Frequency – Inverse Document frequency (TF-IDF)

When you already have the labeled dataset we will used the Supervised machine-learning algorithm. The two different algorithms are; Native Bayes classifier and support vector machine classifier. The two classifiers were the best for analyzing the textual tweeter messages and detecting cyberbullying. They proved to be efficient and accurate in terms of handling massive tweets from different users.

4.1 Naive Bayes (NB) Classifier:

It is the simplest probability classifier; it computes the posterior probability of a class based on distribution of words. The classifier can be advantageous since it can handle a small or large amount of training data to calculate the parameters for prediction.

Instead of calculating the complete covariance matrix, only variance of the feature is computed because of independence of features.

For a given textual review ‘d’ and for a class ‘c’ (positive, negative), the conditional probability for each class given a review is P(c|d). According to Bayes theorem this quantity can be computed using the following equation:

$$P(c \mid d) = \frac{P(d \mid c) * P(c)}{P(d)}$$

To further compute the term P(d|c), it is decomposed by assuming that fi’s are conditionally independent given d’s class. This decomposition of P(d|c) is expressed in following equation:

$$P_{NB}(c \mid d) = \frac{P(c)\left(\prod_{i=1}^{m} P(f_i)|c\right)^{n_i(d)}}{P(d)}$$

4.2 Support Vector Machine (SVM) as a classifier:

SVM is a non-probabilistic binary linear classifier. The main principle of this method is to determine linear separators in the search space that can be used to separate different classes. In this study, SVM Model represents each review in vectorized form as a data point in the space. This method is used to analyze the complete vectorized data and the key idea behind the training of model is to find a hyperplane represented by $\vec{w}$. The set of textual data vectors are said to be optimally separated by hyperplane only when it is separated without error and the distance between closest points of each class and hyperplane is maximum. After training of the model, the testing reviews are mapped in-to same space and predicted to belong to a class based on which side of the hyperplane they fall on.

Let be the class (positive, negative) for a textual message dj, the equation for $\vec{w}$ is given by

$$\vec{w} = \sum_j \alpha_j c_j \vec{d}_j, \alpha_j \geq 0$$

Dual optimization problem gives the values for αj's. All the dj such that αj is greater than zero are termed as Support vectors as they are the only document vectors which are contributing to $\vec{w}$.

Confusion matrix is generated to tabulate the performance of any classifier. This matrix shows the relation between correctly and wrongly predicted reviews. In the confusion matrix, TP (True Positive) represents the number of positive movie reviews that are correctly predicted whereas FP (False positive) gives the value for number of positive movie reviews that are predicted as negative by the classifier. Similarly, TN (True Negative) is number of negative reviews correctly predicted and FN (False Negative) is number of negative reviews predicted as positive by the classifier.

Correct Labels		
	Positives	**Negatives**
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Table 4.1: Confusion Matrix

From this confusion matrix, different Performance evaluation parameter like precision, recall, F-measure and accuracy are calculated. The table of confusion matrix formation is shown in table 4.1. Precision: It gives the exactness of the classifier. It is the ratio of number of correctly predicted positive reviews to the total number of reviews predicted as positive.

$$precision = \frac{TP}{TP+FP}$$

Recall: It measures the completeness of the classifier. It is the ratio of number of correctly predicted positive reviews to the actual number of positive reviews present in the corpus.

$$Recall = \frac{TP}{TP+FN}$$

F-measure: It is the harmonic mean of precision and recall. F-measure can have best

value as 1 and worst value as 0. The formula for calculating F-measure is presented as:

$$F - Measure = \frac{2*Precision*Recall}{Precision+Recall}$$

Accuracy: It is one of the most common performance evaluation parameter and it is calculated as the ratio of number of correctly predicted reviews to the number of total number of reviews present in the corpus. The formula for calculating accuracy is given as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

The dataset, which was considered in this study, is the Sentiment140 dataset (A., Bhayani, R. and Huang, L., 2009) with 1.6 million tweets extracted using the twitter API. The tweets have been annotated (0 = negative, 2 = neutral, 4 = positive) and they can be used to detect sentiment. It contains the following 6 fields:

- **target**: the polarity of the tweet (0 = negative, 2 = neutral, 4 = positive)
- **ids**: The id of the tweet
- **date**: the date of the tweet
- **flag**: The query (lyx). If there is no query, then this value is NO_QUERY.
- **user**: the user that tweeted
- **text**: the text of the tweet

I only selected a sample of 1000 positively, labeled and 1000 negative labeled tweets (A., Bhayani, R. and Huang, L., 2009).

5.0 Proposed Approach

Labeled Twitter sentiment classification has been taken in the consideration, which consist of 1000 positive and 1000 negative reviews (A., Bhayani, R. and Huang, L., 2009). The tweets will go preprocessing step, where all the vague information is removed. From the cleaned dataset, potential features are extracted. The vectorization techniques are used to convert textual data to numerical format. Using vectorization, a matrix is created where each column represents a feature and each row represents an individual review. This matrix is used as input to classification algorithm and cross validation technique is applied to choose the training and testing set for each fold. Step-wise presentation of proposed approach is shown in the figure 5.1.

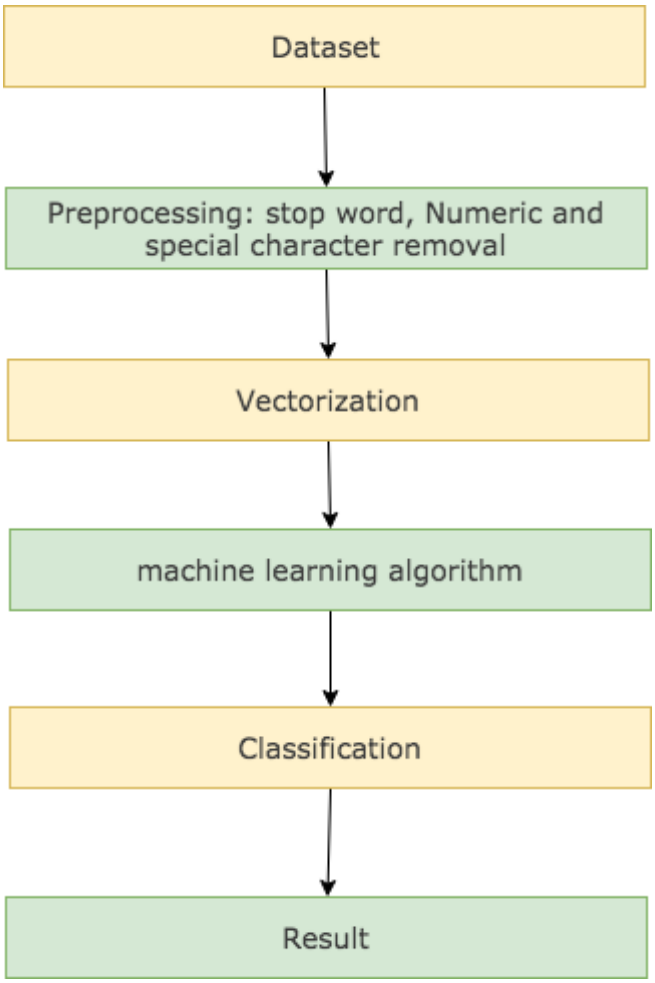

Figure 5.1: Process flow of proposed approach

5.1 Steps followed for classification

Step 1. The Twitter sentiment classification dataset is considered for analysis, which consists of 1000 positive, and 1000 negative labeled reviews. For each review a separate text file is maintained.

Step 2. For the preprocessing I corrected the spelling errors in the tweets and eliminated vague information. English words written in shorthand such as "u r nt beautiful" is converted to "you are not beautiful". The uppercase letter tweets are converted to letters lowercase. It is observed that reviewers often repeat a particular character of a word to give more emphasis to an expression or to make the review trendy. Words like yesssssssss, heheehhehehehe falls in this category. The repetition of characters is also eliminated in this step. Most of the words that do not contribute to any sentiment used in English language are termed as stop words. So, second step in preprocessing involves the removal of all the stop words of English language.

Step 3. In this step the features are tokenized word of a review. These words need to be converted to numerical vectors so that each review can be represented in the form of numerical data. The vectorization of features is done using the following two methods.

- **CountVectorizer**: It transforms the review to token count matrix. First, it tokenizes the review and according to number of occurrence of each token, a sparse matrix is created.

- **TF-IDF**: Its value represents the importance of a word to a document in a corpus. TF-IDF value is proportional to the frequency of a word in a document.
    - Calculation of TF-IDF value: suppose sentiment140 dataset contain 100 words wherein the word Awesome appears 5 times. The term frequency (i.e., TF) for Awesome then (5 / 100) = 0.05. Again, suppose there are 1 million reviews in the corpus and the word Awesome appears 1000 times in whole corpus Then, the inverse

document frequency (i.e., IDF) is calculated as log (1,000,000 / 1,000) = 3. Thus, the TF-IDF value is calculated as: 0.05 * 3 = 0.15.

Step 4. The numeric vectors can be given as input to the classification algorithm. The different classification algorithm used is as follows:

- Naive Bayes (NB) algorithm: Using probabilistic analysis, features are extracted from numeric vectors.
  These features help in training of the Naive Bayes classifier model.

- Support vector machine (SVM) algorithm: SVM plots all the numeric vectors in space and defines decision boundaries by hyperplanes. This hyperplane separates the vectors in two categories such that, the distance from each category to the hyperplane is maximum.

  Initially, the dataset was not divided between testing and training subsets. So, k-fold cross validation technique is used, the number of folds used are 10.

Step 5. After training of model, confusion matrix is generated which shows the number of positive and negative reviews that are correctly predicted and number of positive and negative reviews that are wrongly predicted. For each fold, prediction accuracy is calculated based on this confusion matrix and final accuracy is given by taking the mean of all the individual accuracies of 10 folds. However, individual accuracy of a particular fold can be much higher than the mean of all accuracies.

Step 6. For each model, values of precision recall and F-measure as performance evaluation parameters are found out. The confusion matrix and a table containing performance evaluation parameter are generated. Finally, these obtained results are compared with the values obtained by other authors in literature.

6.0. Implementation

The implementation of above mentioned algorithms are carried out on Sentiment140 dataset (A., Bhayani, R. and Huang, L., 2009). After implementing the two machine learning approaches to determine which of the classifiers used for detecting cyberbullying gave the most accurate results and is preferable over the other, we determined the accuracy of the classifier, precision, recall and f-score of the positive, negative and neutral tweets.

6.1 Naive Bayes Algorithm:

The confusion matrix obtained after implementation of Naive Bayes classification algorithm is shown in table 6.1.

Table 6.1: confusion matrix for Naïve Bayes classifier

Correct Labels		
	Positives	**Negatives**
Positive	11107	1393
Negative	2384	9666

The performance evaluation parameters obtained for Naive Bayes classifier is shown in table 6.2.

Table 6.2: Evaluation parameters for Naïve Bayes classifier

	Precision	**Recall**	**F-Measure**
Positive	0.80	0.89	0.84
Negative	0.87	0.77	0.82

Maximum accuracy achieved after the cross validation analysis of Naive Bayes classifier is **0.8953**.

6.2 Support Vector Machine Algorithm:

The confusion matrix obtained after implementation of Support Vector Machine algorithm is shown in table 6.3.

Table 6.3: Confusion matrix for Support Vector classifier

Correct Labels		
	Positives	**Negatives**
Positive	11102	1393
Negative	1688	10812

The performance evaluation parameters obtained for Support Vector Machine classifier is shown in table 6.4.

Table 6.4: Evaluation parameters for Support Vector classifier

	Precision	**Recall**	**F-Measure**
Positive	0.87	0.89	0.88
Negative	0.89	0.86	0.88

Maximum accuracy achieved after the cross validation analysis of Support Vector Machine classifier is **0.9406**.

7.0 Comparison Analysis

In this section we will compare the output obtained from the two proposed machine-learning approaches with the output from other manuscripts. The manuscripts that were taken into consideration are by Pang Lee and another by read. The two manuscripts used the same polarity dataset with 1000 positives and 1000 negatives reviews.

Table 7.0: Comparison of proposed work with existing literatures

	(Pang and Lee)	**(Read)**	**Proposed approach**
Naïve Bayes	0.864	0.789	0.895
Negative	0.8615	0.815	0.940

From the table 7.0 it is clear that the accuracy obtained in the proposed approach is more accurate than those from both manuscripts. All the other research methods had a high precision but a low recall meaning that most of the predicted labels are correct, the precision and recall are normally used to determine the quality of classifier output.

8.0 Conclusions

In this thesis we have investigated the possibilities of building a system capable of automatically identifying cyberbullying on social media using an empathic agent. This chapter summarizes what we have learned, discusses limitations of the achievements and concludes with a brief look at future work.

The goal with the project was to generate knowledge of how an automatic system for detecting bullying on social media could be constructed. During the project we have learned how to employ state of the art methods in Natural Processing Language (NLP) for bullying classification. Which is the core in any bullying detection system. We have learned the limitations of this method in the form of differentiating between common bad language and cyberbullying. We have learned about some of the challenges that we are facing when trying to automatically scan a social media platform. Most importantly we have learned that an automatic system for bullying detection is possible to some extent as shown by the implemented prototype. The greatest limiting factor is how well classification can be performed. With the knowledge gained during this project it is the author's opinion that the step to a real world application is not very far.

8.1 Future Scope

As of now this system is implemented only on textual data. In the future we plan on extending the scope of our system by incorporating cross-media detection in the form of audio, video and images too. We also plan to try to make our system be context-aware with the help of deep learning in the future.