

"ICT4D-inspired Text Classification: Building a Naive Bayes-based Community Message Board Filter for Abusive Comments Detection"

Shouju Wang

Abstract—With the advancements in technology, particularly in the field of artificial intelligence, an increasing number of people have witnessed improvements in their lives. However, it is undeniable that these technological advancements have predominantly revolved around developed countries, mainly in Europe and America. Consequently, this has resulted in issues such as technology bias[1], including disparities in access and opportunities[2]. For instance, the development of social media has greatly bridged the information gap between developing regions and countries, enabling easier access to news and information from around the world. However, the presence of racial and gender discrimination, along with other forms of abusive language, on social media message boards often hinders the further progress of social media technologies in these regions. In response to this situation, the field of ICT4D[3] within human-computer interaction aims to utilize information and communication technology (ICT) to foster social and economic development, improve the quality of life, and provide sustainable development opportunities in developing countries and regions. In this paper, we propose leveraging the ICT4D principles[4] and employing the naive Bayes method from the field of machine learning to develop a community message board filter. The aim is to create a better community media environment by addressing the challenges of abusive content. This approach aligns with the overarching goal of ICT4D to harness the potential of ICT for promoting inclusive and sustainable development in underserved regions.

ICT4D, Text Classification, Naive Bayes, Abusive Comments, Machine Learning, Big Data Analysis

1 INTRODUCTION

Human interaction with social networks, forums, and online blogs has increased dramatically in recent years. These platforms create a strong community centered around communication through messages, chats, and comments.

Comments provide an informal and interactive means for individuals to express their personal viewpoints. Commenters can freely share their sentiments, respond to others, and contribute to collective knowledge. Readers benefit from additional information provided in the comments section and often engage in discussions by leaving replies. Users can also provide quick feedback on comments using features like the "thumbs up" or "thumbs down" sign. Moreover, in-depth conversations can be initiated through comment threads, allowing for more detailed exchanges. As a result, comments foster a sense of collective interest, with minimal barriers to participation, and bridge the information and technology gap between developing and developed regions, benefiting the poor and the marginalized.

However, the comment framework in social networks is an essential part that allows anonymous posting, and the results of abusive comments can be multifarious. Readers may lose their passion and enthusiasm while filtering good comments from spam, while an ocean of spam could discourage normal commenters from posting their valuable contents. The content producer may also receive less feedback, leading to lower quality content.

Unlike other community features, comments are hard to manage. Moderators have a critical undertaking in securing the comments of a forum. Typically, human moderators are responsible for reading and classifying each comment as abusive or non-abusive. How-

ever, the manual review and identification of offensive comments is an arduous and time-consuming task, making it impractical, unreliable, and inefficient in practice.

To address this issue, automated software such as "Appen" and "Internet Security Suite" have been developed to detect and deter abusive comments [5]. However, these software packages can interrupt the readability and usability of the website and fail to identify subtle insulting contents. The purpose of this research is to detect and block abusive comments without compromising the readability of the web pages and contents.

To achieve this goal, the English "Hate Speech and Offensive Language Dataset" was collected from an existing Kaggle dataset. This dataset contains Twitter data, classified as hate speech, offensive language, or neither. The class labels are defined as follows: 0 - hate speech, 1 - offensive language, 2 - neither. A Naïve Bayes classifier was trained on this dataset, and the 10-fold cross-validation technique was applied to measure the accuracy of the classifier.

2 RELATED WORK

Text mining with the machine learning has been conducted and marked by many scientists. Previous studies have explored the detection of abusive and offensive comments in various ways. Abusive and offensive comments classification research began with Yin using a supervised machine learning technique. In the research, text are illustrated based on word frequency features, sentiment features and features which take the similarity to neighboring posts. An early study on tackling abusive language was conducted, where a supervised classification technique was

employed alongside N-gram models. Additionally, the researchers manually crafted regular expression patterns and contextual features that considered the offensive nature of preceding sentences.

Dinakar [6] have collected a dataset from YouTube videos covering various topics that included comments, and they applied both binary and multiclass classifiers. The experimental findings demonstrated that topic-sensitive binary classifiers improved the performance of generic multiclass classifiers. In another study, Dadvar [7] utilized a rule-based expert system, a supervised machine learning model, and a hybrid approach to automatically detect instances of cyberbullying. The results showed that the expert system outperformed the machine learning and hybrid approach models. Another approach was proposed by Nahar [8], who introduced a semi-supervised learning method involving the enlargement of training data samples and the use of a fuzzy SVM algorithm. This improved training method automatically extracted and augmented the training set from unlabeled streaming text, and learning was conducted using a limited initial input of a training set. The experimental results demonstrated that the proposed improved technique outperformed other methods and was applicable in practical scenarios where a sufficient labeled dataset was unavailable for training.

The authors [9] devised and implemented a novel method for annotating cyberbullying, which captured the presence and severity of cyberbullying, the role of the post author (harasser, victim, or bystander), and various fine-grained categories associated with textual harassment, such as insults and threats. The experimental results demonstrated the feasibility of detecting fine-grained instances of cyberbullying. Reynolds [10] collected data from the social networking site "FromSpring" and applied a machine learning algorithm to this dataset. They utilized Amazon Web Services' Turk platform to label the collected data. Their technique achieved an accuracy of 78.5% in identifying true positives.

Abdul [11] proposed a new method to detect offensive language expressed in Bangla, taking into consideration the perspective of ICT4D. However, the aim of this research is to detect abusive comments expressed in languages other than English.

3 METHODOLOGY

In this research, the Naïve Bayes classifier is employed to identify abusive comments written in Bangla, aiming to accomplish the following objectives. The methodology involves three main steps: 1) data acquisition and preprocessing, 2) feature extraction, and 3) model selection. The feature selection stage presents significant challenges when utilizing a text mining approach to differentiate hostile content.

3.1 Data Acquisition and Preprocess

Our objective is to propose a text classification system that encompasses languages spoken worldwide. However, the availability of datasets varies across regions due to unequal development. The majority of datasets are primarily based on English. For our research, we employed our methodology using English datasets, assuming that other languages have already been translated into English. Fortunately, with advancements in NLP research and machine learning, there are existing datasets related to abusive comments from the internet that we can directly download for our research purposes.

For the convenience of research, we download the labeled data set which already have been preprocessed from kaggle directly.

Dataset using Twitter data, is was used to research hate-speech detection. The text is classified as: hate-speech, offensive language, and neither. Due to the nature of the study, it's important to note that this dataset contains text that can be considered racist, sexist, homophobic, or generally offensive.

This data set consist of 6 columns. Which are explained as follows.

#	# count	# hate_spe...	# offensive...	# neither	# class	A tweet
0	3	0	0	3	2	!!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house, & as a man...
1	3	0	3	0	1	!!!! RT @elsew7: boy dats cold...tyga dem bad for cuffin dat hoe in the 1st place!!
2	3	0	3	0	1	!!!! RT @urkindoffrand Dang!!!! RT @BibababyLife: You ever fuck a bitch and she start to cry? You...

Fig. 1: overview of data set

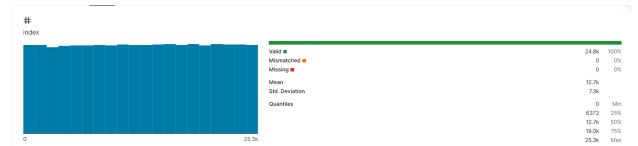


Fig. 2: data set column 'index'

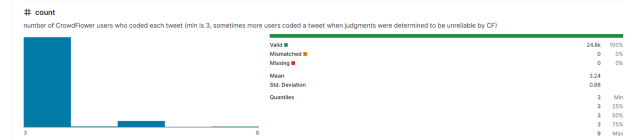


Fig. 3: data set column 'count'

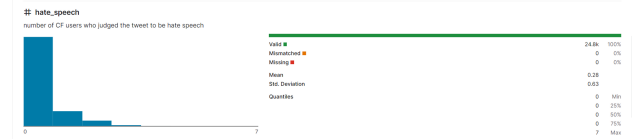


Fig. 4: data set column 'hate speech'

3.2 Model Explain

To moderate a website, a classifier can be used instead of relying on human moderators or users to flag comments, providing faster operation. The task in relation to the dataset used in this paper involves binary classification. Hence, the Naïve Bayes approach is selected for classifying abusive comments. The Naïve Bayes approach is simple to implement and computationally efficient. Naïve Bayes is a subset of Bayesian decision theory. The Bayes Theorem allows us to calculate the likelihood of an event leading to a particular outcome. Let C be the set of i classes and D be a document to be classified. The probability of a class C given a

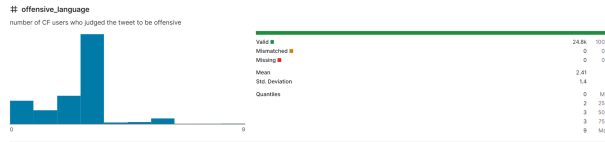


Fig. 5: data set column 'offensive language'

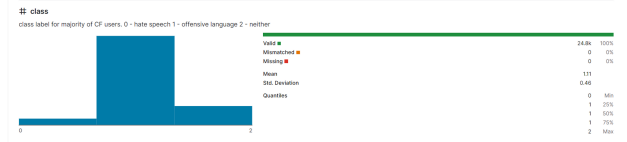


Fig. 7: data set column 'class'

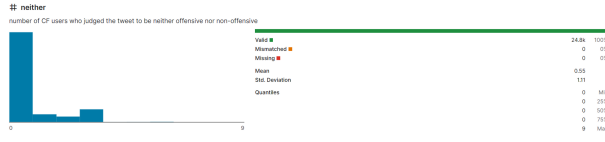


Fig. 6: data set column 'neither'

```

10 class_labels = class_labels.tolist()
11 split_text_data = []
12
13 for text in text_data:
14     words = text.split()
15     split_text_data.append(words)

```

With the prepared data, we then split the data set into training data and test data

document D can be calculated using Bayes' Theorem, which can be expressed as follows:

$$P(C|D) = \frac{P(D|C)P(C)}{P(D)} \quad (1)$$

If the document D is expanded into individual features, the probability of a document $P(D|C)$ can be calculated as follows:

$$P(D|C) = P(d_1, d_2, \dots, d_n|C) \quad (2)$$

The assumption is that all the words are independently likely, which is known as conditional independence, and the probability is calculated as follows

$$P(d_0|C_i) \times P(d_1|C_i) \times \dots \times P(d_n|C_i) \quad (3)$$

Using the above definitions, the Bayesian classification criterion can be defined as follows:

- if $P(C_1|D) > P(C_2|D)$ then it belongs to class c_1 ,
- if $P(C_1|D) < P(C_2|D)$, then it belongs to class c_2 .

3.3 Preparation for the process

In the original dataset, each data item is categorized into classes, where 0 represents "hate speech," 1 represents "offensive speech," and 2 represents "neither," which refers to "normal" comments. In order to apply the Naïve Bayes model, we use Python code to map 0 and 1 in the "class" column to 1, and 2 to 0. The code is as follows:

```

1 #modify class column
2 data_pre = pd.read_csv('labeled_data.csv')
3 column_name = 'class'
4 data_pre[column_name] = data_pre[column_name].
5   replace({2: 0, 0: 1})
6 data_pre.to_csv('modified_data.csv', index=False)

```

After modifying the "class" column, the data is further processed by removing punctuation marks and special characters from the text. The text is then split into individual characters.

```

1 #
2 data_mid = pd.read_csv('modified_data.csv')
3 text_data_mid = data_mid['tweet']
4 class_labels = data_mid['class']
5 #
6 text_data = text_data_mid.apply(lambda x: re.sub
7   (r'[\W\s]', '', x))
8 text_data = text_data.str.lower()
9 #
10 class

```

3.4 Training process

- 1) First, we constructing word vectors from text

```

1 def createVocabList(dataSet):
2     vocabSet = set([]) # create empty set
3     for document in dataSet:
4         #
5         vocabSet = vocabSet | set(document) #
6         union of the two sets
7     return list(vocabSet)
8
9 def setOfWords2Vec(vocabList, inputSet):
10    returnVec = [0] * len(vocabList) #
11    [0,0,...]
12    #
13    for word in inputSet:
14        if word in vocabList:
15            returnVec[vocabList.index(word)]
16            = 1
17    return returnVec

```

- 2) Then, we train the algorithm from computing probabilities from word vectors

```

1 def trainNB0(trainMatrix, trainCategory):
2
3     numTrainDocs = len(trainMatrix)
4
5     numWords = len(trainMatrix[0])
6
7     pAbusive = sum(trainCategory) / float(
8       numTrainDocs)
9     p0Num = ones(numWords)
10    p1Num = ones(numWords)
11
12    p0Denom = 2.0
13    p1Denom = 2.0
14    for i in range(numTrainDocs):
15        if trainCategory[i] == 1:
16            p1Num += trainMatrix[i]
17            p1Denom += sum(trainMatrix[i])
18        else:
19            p0Num += trainMatrix[i]
20            p0Denom += sum(trainMatrix[i])
21    p1Vect = log(p1Num / p1Denom)
22    p0Vect = log(p0Num / p0Denom)
23    return p0Vect, p1Vect, pAbusive

```

- 3) Then, we use the algorithm to do the text classification. The function argument `vec2Classify` refers to our `test_data`

```

1 def classifyNB(vec2Classify, p0Vec, p1Vec,
2   pClass1):
3     p1 = sum(vec2Classify * p1Vec) + log(
4       pClass1)
5     p0 = sum(vec2Classify * p0Vec) + log(1.0
6       - pClass1)
7     if p1 > p0:
8         return 1
9     else:
10        return 0

```

4 EXPERIMENTS AND RESULTS

4.1 Test the data

To test all the data we obtained from Kaggle, we use the following code.

```

1 #
2 mytrainList = createVocabList(train_data)
3 trainMat = []
4 for postinDoc in train_data:
5     trainMat.append(setOfWords2Vec(mytrainList,
6     postinDoc))
7
8 #
9 p0V, p1V, pAb = trainNB0(array(trainMat),
10 train_labels)
11 #
12 testMat = []
13 for postinDoc in test_data:
14     testMat.append(setOfWords2Vec(mytrainList,
15     postinDoc))
16 testMatrix = array(testMat)
17 re = []
18 for i in range(len(testMatrix)):
19     result = classifyNB(testMatrix[i], p0V, p1V,
20     pAb)
21     re.append(result)

```

4.2 Detecting accuracy

In the previous introduction, we divided the data into training data and testing data, and also split the labels. To evaluate the accuracy of the model's performance, we use a function to compare the accuracy of the predictions.

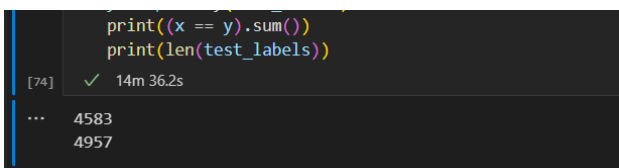
```

1 x = np.array(re)
2 y = np.array(test_labels)
3 print((x == y).sum())
4 print(len(test_labels))

```

4.3 Result

Run the code, we get the following result



```

print((x == y).sum())
print(len(test_labels))

```

[74] ✓ 14m 36.2s

... 4583
4957

Fig. 8: Runing result

4.4 Evaluation

Based on the running result, we calculate the accuracy.

$$acc = \frac{4583}{4957} = 92.46\% \quad (4)$$

The complete code can be found in the my following GitHub repository. CODE HERE

5 CONCLUSIONS

Inspired by the principle ICT4D, in this paper, we use the Naïve Bayes method from machine learning to address the abusive comments expressed in non-English language from community message board. We aim to filter the offensive content for the developing region people to have a better media enviroment. By applying the advancement in natural language processing and machine learning, we developed a text classification system which can classify the absive comments in all different language.

Our methodology involved three main steps : data acquisition and preprocessing, model explain, preparation for the process, and training process. For the convience of research, we doenload a labeled dataset of twitter data from Kaggle, which is the base of our research. The comments in the dataset are classified as hate speech, offensive speech and neither. During the data preprocess-ing, we combine the hate speech and offensive language to be abusive, which is labeled to 1, and neither to be 0, which means normal comments. We also split the text into individual characters and removing punctuation marks and special characters.

For the process of training the calssifier, we first use Naïve Bayes to construct word vectors from `train_data` to produce the probilities `p0V`, `p2V` `pAb`. Because the Naive Bayes assumes conditional independence of words and calculates the probability of a document belonging to a particular class.

We can use the calculated probabilities to produce the classifi-cation of `test_data`.

To evaluate the accuracy of our model, we compare the predicted labels with the actual labels of the test data. Durig this process, we use the code

The experiment result shoed that our model's accuracy is 92.46%. This manifest that our Naive Bayes classification model is efficient in detecting and classifying abusive comments.

To make our result futher, we can use this technique to the media around the world to create a safer and more inclusive enviroment to bridge the technological gap around the world.

In conclusion, our research, basing on the value of ICT4D, leverage machine learning techniques to address the challenages of abusive comments in comminuty message boards. Our approach demonstrates the potential of using technology to foster social and economic development and promote inclusive and sustainable development in developing regions. Future research could focus on expanding the dataset to include more languages and further improving the accuracy and robustness of the classifier.

ACKNOWLEDGMENT

I would like to express my great gratitude to all people who inspired me to complete this research.

First and foremost, I would like to thank my Dig Data Analysis teacher Dan Chen, who give me the fundamental tool and methoadd to use basic maching learning to do data analysis. His great teaching skill has enable me to finish this works.

We would also like to acknowledge the researchers and scientists who have dedicated their time and effort to the field of ICT4D (Information and Communication Technology for Development) within human-computer interaction. Their pioneering work has paved the way for utilizing information and communication technology to foster social and economic development, improve the quality of life, and provide sustainable development opportunities in developing countries and regions.

And I also like to thank the creators and contributors of the kaggle platform, where I get the "Hate Speech and Offensive Language Dataset" that served as the foundation for our research. We would like to express our gratitude to the authors of the scientific papers and studies mentioned in the "Related Work" section. Their insightful research and findings have provided valuable insights into the detection of abusive and offensive comments, and have guided our approach in developing a community message board filter.

REFERENCES

- [1] A. Díaz Andrade and C. Urquhart, "Unveiling the modernity bias: A critical examination of the politics of ict4d," *Inf. Technol. Dev.*, vol. 18, no. 4, p. 281–292, oct 2012. [Online]. Available: <https://doi.org/10.1080/02681102.2011.643204>
- [2] A. Díaz Andrade and C. Urquhart, "Unveiling the modernity bias: a critical examination of the politics of ict4d," *Information Technology for Development*, vol. 18, no. 4, pp. 281–292, 2012.
- [3] W. Chipidza and D. Leidner, "Ict4d research—literature review and conflict perspective," 2017.
- [4] D. Thapa and M. Hatakka, "Introduction to ict4d: Icts and sustainable development minitrack," 2017.
- [5] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, "Detecting offensive language in social media to protect adolescent online safety," in *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing*, 2012, pp. 71–80.
- [6] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of textual cyberbullying," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 5, no. 3, 2011, pp. 11–17.
- [7] M. Dadvar, D. Trieschnigg, and F. De Jong, "Experts and machines against bullies: A hybrid approach to detect cyberbullies," in *Advances in Artificial Intelligence: 27th Canadian Conference on Artificial Intelligence, Canadian AI 2014, Montréal, QC, Canada, May 6-9, 2014. Proceedings 27*. Springer, 2014, pp. 275–281.
- [8] V. Nahar, S. Al-Maskari, X. Li, and C. Pang, "Semi-supervised learning for cyberbullying detection in social networks," in *Databases Theory and Applications: 25th Australasian Database Conference, ADC 2014, Brisbane, QLD, Australia, July 14-16, 2014. Proceedings 25*. Springer, 2014, pp. 160–171.
- [9] C. Van Hee, E. Lefever, B. Verhoeven, J. Mennes, B. Desmet, G. De Pauw, W. Daelemans, and V. Hoste, "Detection and fine-grained classification of cyberbullying events," in *Proceedings of the international conference recent advances in natural language processing*, 2015, pp. 672–680.
- [10] K. Reynolds, A. Kontostathis, and L. Edwards, "Using machine learning to detect cyberbullying," in *2011 10th International Conference on Machine learning and applications and workshops*, vol. 2. IEEE, 2011, pp. 241–244.
- [11] M. Awal, M. Rahman, and J. Rabbi, "Detecting abusive comments in discussion threads using naïve bayes," 10 2018, pp. 163–167.