Hate Speech Detector

Yuxiang Wang, Jingxi Liu, Wenkai Luo, Yuetong Liu
ywang594, jliu238, wluo14, yliu390

Abstract

The development of network makes the spread and negative influence of hate speech break through the limitation of region and time; however, it also provides a platform to detect and limit hate speech by artificial intelligence. In this study, we will use skip-gram based word2vec method or n-gram with tfidf as vectorizers using PMI as an indicator to realize feature selection. Then we use Word Count dictionary (LIWC) and LDA to complete our feature extraction and apply logistic regression and support vector machine (SVM) as text classification techniques to develop a model that can distinguish tweets containing hate speech key words into three categories (hate speech, offensive speech, and normal speech).

1 Project choice

Choose either a **methods** or **applications** project, and a subarea from the below table.

☒ **Applications**				
☐ Genomics data	☐ Healthcare data	☒ Text data	☐ Image data	☐ Finance data
☐ **Methods**				
☐ Fairness in ML	☐ Interpretable ML	☐ Graphical Models	☐ Robust ML	☐ Privacy in ML

2 Introduction

Hate speech is hard to be detected as there is no legal definition of "hate speech" under U.S. law. However, it is generally believed that speech based on race, religion, nationality, gender, sexual orientation, disability and other identity characteristics, which deliberately incites violence and prejudice against individuals or groups, may constitute hate speech [1]. Hate speech contributes to a general climate of intolerance, which in turn makes attacks against specific groups more likely. If hate speech is tolerated without limitation, minority groups will suppress their expression because of fear, thus becoming more silent and marginalized. Many countries have laws against hate speech: the public order act of the United Kingdom and the criminal law of Canada prohibit the public expression of incitement to racial and religious hatred. Germany banned public statements of Holocaust denial and hatred against ethnic minorities. Japan also legislated against hate speech [2].

With the development of Internet usage, although negative emotion expressions increase for internet's virtuality and lack of regulation, it makes the detection and supervision of hate speech possible to achieve by artificial intelligence. But just like what Mark Zuckerberg said, as the detection of hate speech depends on context and domain, it is difficult to detect it. Hate speech will try to evade or poison such machine learning classifiers [3]. Another problem the researchers found is that some classifiers tend to confuse offensive speech with hate speech, resulting in false positives. They found that a single algorithm with three tags (hate speech, offensive speech, and normal speech) performed better in avoiding false positives than two tags [4]. But it is still difficult to eliminate false positives completely.

In our study, we try to figure out the two key challenges for automatic hate speech detection on social media. The first goal of the study is to decide whether a tweet post is a hate speech. The second goal is to separate hate speech and other offensive languages. To come up with an algorithm for classifying hate speech, we first need to teach machine "what is a hate speech." To some extent, tagging is still a very subjective behavior. Therefore, we use the dataset that contains more than 20 thousand English sentences in tweet posts as observations and their labels judged by CrowdFlower users as our input to train and test our models. We then use logistic regression and SVM as our text classification techniques to predict the label for each tweet. We will try to improve the accuracy by fine-grained labels and different ways of feature extraction. We feed our models with the features that reflect users' language by employing the LIWC dictionary, LDA topics, and N-gram features. And we will divide twitter into three categories: hate speech, offensive language, or neither. On the result

page, we will conclude with a model to distinguish these categories, and then analyze the results to better understand how to distinguish them.

3 Dataset and Features

3.1 Descriptions of dataset

We have 24783 examples in total, and we use a 2:1:1 ratio to split our data set into training/validation/test sets. So training data set has 12391 examples, and both validation set and test set have 6196 examples.

Each dataset contains 6 colums:
count = number of CrowdFlower users who coded each tweet (min is 3)
hate speech = number of CF users who judged the tweet to be hate speech.
offensive language = number of CF users who judged the tweet to be offensive.
neither = number of CF users who judged the tweet to be neither offensive nor non-offensive.
class = class label for majority of CF users. 0 - hate speech 1 - offensive language 2 - neither
tweet = the specific content of each tweet

3.2 Pre-processing of dataset

Since our task is to analyze the content of tweets, we need to apply relevant processing of text data:

1 Delete duplicate tweets
2 Remove stop words for each tweet
3 Remove punctuation and excessive whitespace
4 Convert tweet content to lowercase

3.3 Feature extraction and selection

For this task, the training labels correspond to "class" column in our dataset. (the results of the CF user's evaluation of each tweet)

For each tweet, we use the skip-gram method in word2vec for word embedding process. (Ensure that our word vector can contain contextual information and word order information). We may also try n-gram method with tfidf to compare the effect of vectorizing. Then we use PMI as an indicator to select a fixed number of the most important features. We can use chi-square test as an alternative of the feature selection.

After text vectorizing and feature selection, we use LIWC dictionary as the pre-defined category to measure all the textual content of tweets to extract lexico-syntactic features. Besieds, we also consider using LDA method to add topic features of our model.

Finally, we would try to input different combinations of features derived from above encoding methods and figure out the best combination. For example, one possible combination is skip-gram+LIWC+topic modeling(LDA).

3.4 Example in our dataset

count: 3
hate speech: 0
offensive language: 0
neither: 3
class: 2
tweet: !!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house. & as a man you should always take the trash out...

4 Methods

The machine learning methods applied in hate speech detection can be divided into two main streams including the classic method such as logistic regression and SVM, and deep learning method. We will explore both methods in this project. The classic method can be generalised as document classifying task, which requires the manually feature generation as the input. The classic method performs state of art in this field with different feature sets, which has been proved in many researches. However, a more elaborate features sets might help to improve the performance using classic method. In this project, we propose a new feature set with different combination of features. This part will build off the project that generate DWMW17 dataset [4], with the modification of features set. The logistic regression model with new features set will be considered as baselines model for our project. The second part of our project is to explore the application of deep learning in this field. We will implement Multilayer Perceptron (MLP) and Convolutional Neural Networks (CNN) with the use of python libraries Pytorch. The deep learning method isn't proposed in DWMW17 paper, but it seems like a promising method that deserve more exploration.

4.1 Classic Method

Based on the prior work and literature review. Both logistic regression and SVM seem promising methods for this classifying application. In this project, we will be focusing on logistic regression with L2 regularization. A one-versus-rest framework will be applied in this project where each classifier will be trained separately for the specific class. During the testing stage, the class with the highest predicted probability will be labelled for the corresponding sample.

In term of logistic regression. Its hypothesis class is linear functions. Even though the NLP data might not be linear, which violate the assumptions of logistic regression, the feature-extraction of the text sequence will tackle this issue to make the samples linearly separable. In this specific application, the weights for corresponding features represent the importance of the feature to the classifier. For example, the TF-IDF for the specific offensive token could be a significant feature. Each classifier model can be written in the following form:

$$P(Y \mid \mathbf{X}, \mathbf{W}) = P(Y = 0 \mid \mathbf{X}, \mathbf{W})^{\delta(Y-0)} P(Y = 1 \mid \mathbf{X}, \mathbf{W})^{\delta(Y-1)}$$

where $\delta(Y - u) = 1$ if and only if $Y = u$, 0 otherwise. For the logistic regression with L2 regularization, the loss function is the log loss, which can be written in the following form:

$$L = E_w(Y, x) + E_w(w)$$

Where

$$E_w(x) = -\sum_{i=1}^{N} [y_i log(\sigma(\mathbf{W}\mathbf{x})) + (1 - y_i) log(1 - \mathbf{W}\mathbf{x})]$$

$$E_w(w) = -\lambda \sum_{c=1}^{C} ||\vec{w}_c||_2^2$$

$$\sigma(\mathbf{W}\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{W}^\mathbf{T}\mathbf{x}}}$$

The log loss can also be considered as the cross entropy, which somehow measure the unexpectedness of the model, which make our object function suitable for logistic regression. One of advantage of using log loss is that it converts the non-convex loss function into a convex function, which is a safeguard for global optimum using optimization method. The other advantage of log loss is that it has the desired mathematical properties for this specific model.

To prevent over-fitting of the model, we also introduce the L2 regularization into the loss function. With the introduction of L2 regularization, training process will add more penalty to those sparse weights with high peak value, thus it makes the model favour the small weighs. Moreover, we might use L2 regularization to quantify the importance of every features and conduct feature selection based on penalty.

In term of the optimization method, we will apply gradient descent method to find the global minimizer of the loss function. There are two main gradient descent methods including batch gradient descent (BGD) and stochastic gradient descent (SGD).

We will choose suitable method based on the number of training samples. Compared with the SGD, BGD is able to find the global minimizer of the object function accurately, but its training rate is much slower with large number of training sample. SGD can compensate the shortcoming of BDG, but it will produce noisy update, which decrease the accuracy of minimizer. Thus, we need to balance this trade-off to obtain a good model with higher accuracy. This part of project will be coded in Python and our major code course is scikit-learn.

4.2 Deep Learning

We have seen the combination of Multilayer Perceptron (MLP) and Convolutional Neural Networks (CNN) did a really good job in image classification, which might also be adapted into hate speech detection. The hypothesis class of neural network is universal function approximator. The nonlinearity of model is introduced by the activation function for each layer.

There are several common activation functions including sigmoid function, tanh function and ReLu function. The sigmoid function also introduces the gradient vanishing issues into the model, which might create dead perceptron. Based on that, ReLu is more likely to be used in neural network. The Loss function of our model is also cross entropy, which has been discussed in precious section

The power of the neural network relies on the structure of network including the number of layers, type of layer, channel number and so on. With the elaborated design of model, our model can learn the abstract features that can be used to make the correct classification. A more complicated neural network has stronger learning ability; however, it also increases the risk of over-fitting, thus we need to make a balance between this trade-off. To regularize the model, we can have two main strategy. One is to modify the dataset itself such as data augmentation. The other one is to modify our training process. We can have early stopping for the training process, dropout of some perceptron during each step and adding noise into the inputs. We will use such methods based on the situation we encounter when we train our model.

In term of optimization method, we will use error backpropagation method to update the parameters in the model. The Pytorch libraries provide different optimizing method, which we will do experiment to find the suitable one. The batch size and the learning rate are the two important part for this optimization method. It is like mini-batch gradient descent, which balance the noisy update and the accuracy.

4.3 Evaluation

To compare the performance, we will utilize F1-score to quantify the measure of performance. The sklearn libraries provide f1-score function, which we will use in our project.

F1-score is generally used to measure the performance of classification. F1-score can be computed by the following formula:

$$F_1 = 2 \cdot \frac{\textbf{precision} \cdot \textbf{recall}}{\textbf{precision} + \textbf{recall}}$$

where precision and recall can be computed by following:

$$\textbf{precision} = \frac{TP}{TP + FP}$$
$$\textbf{recall} = \frac{TP}{TP + FN}$$

where TP, FP and FN are the number of true positive, false positive and false negative respectively.

5 Deliverables

5.1 Must accomplish

1. Data Pre-processing: pre-process text data including deleting duplicates, removing stop words or punctuation, and converting tweet content to lowercase.
2. Feature Extraction: Combine skip-gram, LIWC and LDA methods to extract features.
3. Feature Selection: Use PMI on the training data to quantified the importance of each feature. Set up a threshold to choose the most beneficial features.

5.2 Expect to accomplish

1. EDA: conduct exploratory data analysis using Word Clouds or bar graph on polarity. Moreover, conduct correlation analysis between features.
2. Prediction: Use the F1 value on the Davidson, T. et al. (2017) as a baseline, accurately classify the text comment into hate speech, offensive language or neither. Otherwise, interpret the model to explain the mis-classification.
3. Model Selection: Besides logistic regression model, use MLP + CNN as a comparisons to train data and make model selection based on their performance. Use cross validation to find optimal hyper-parameters.

5.3 Would like to accomplish

1. Classification Bias: Reduce model bias since the class is imbalanced. Beside improving the precision, choose model based on evaluation metrics.
2. Unsupervised learning: clustering hate speech to identify major topics, including race, colour, sex.
3. Improvement: link data with CF users to study who are more likely to use hate speech/offensive language. Integrate the frequency of a user detected for posting hate speech and the user's interaction with others.

References

This section should include citations for: (1) Any papers on related work mentioned in the introduction. (2) Papers describing methods that you used which were not covered in class. (3) Code or libraries you downloaded and used.

[1] Ward, K. (1998). Free Speech and the Development of Liberal Virtues: An Examination of the Controversies Involving Flag-Burning and Hate Speech. Retrieved from https://repository.law.miami.edu/umlr/vol52/iss3/4/

[2] Muntarbhorn, V. (2011). "Study on the prohibition of incitement to national, racial or religious hatred: Lessons from the Asia Pacific Region." Retrieved from https://www.ohchr.org/Documents/Issues/Expression/ICCPR/Bangkok/StudyBangkoken.pdf

[3] Gershgorn, D. (2018). "Mark Zuckerberg just gave a timeline for AI to take over detecting internet hate speech." Retrieved from https://qz.com/1249273/facebook-ceo-mark-zuckerberg-says-ai-will-detect-hate-speech-in-5-10-years/

[4] Davidson, T. & Warmsley, D. & Macy, M& Weber, I. (2017) "Automated Hate Speech Detection and the Problem of Offensive Language". Retrieved from https://arxiv.org/pdf/1703.04009.pdf

[5] Davidson, T. (2017) Data.world: Hate Speech and Offensive Language. Retrieved from https://data.world/thomasrdavidson/hate-speech-and-offensive-language

[6] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.