WikiPedia Online Attacking Detection

CSCE 633 Project Final Report

Xing Zhao
Texas A&M University
College Station, Texas
xingzhao@tamu.edu

Sirui Li
Texas A&M University
College Station, Texas
lsr1993@tamu.edu

1 INTRODUCTION & MOTIVATION

21th century is the age of the Internet. However, many problems appear with the growing number of net users. One of them which impairs the user experience most is the personal attacks. Many platforms try their best to curb the phenomenon. However, it remains super difficult to recognize the offensive comments from large scale dataset. With the rise of social media platforms, online discussion has become integral to people's experience of the internet. Unfortunately, online discussion is also an avenue for abuse. A 2014 Pew Report highlights that 73% of adult internet users have seen someone harassed online, and 40% have personally experienced it. Platforms combat this with policies concerning such behavior. For example Wikipedia has a policy of "Do not make personal attacks anywhere in Wikipedia" and notes that attacks may be removed and the users who wrote them blocked.The challenge of creating effective policies to identify and appropriately respond to harassment is compounded by the difficulty of studying the phenomena at scale. Typical annotation efforts of abusive language, such as that of Warner and Hirschberg, involve labeling thousands of comments, however platforms often have many orders of magnitude more; Wikipedia for instance has 63M English talk page comments. Even using crowd-workers, getting human-annotations for a large corpus is prohibitively expensive and time consuming. We implemented a new method based on the paper [5], which combines the crowdsourcing and machine learning to analyze personal attacks at scale. This topic is hot and attractive. We applied most methods learned in class and gained much from this experience.

Our motivation derived from the efficiency behind the algorithm. The result of the suggested method is extemely good. However, it's not possible to operate for our students, as the size of input comments is too large. Instead of simply cutting down the dataset, we decided to figure out a new way to translate the input data. Beyond that, we want to relate more models learned in class to this experiments. It's an interesting feeling that what we studied in class is actually useful in practice.

2 BRIEF LITERATURE REVIEW

The author treated the problem of identifying personal attacks as a binary text classification problem. They explored logistic regression(LR) and multi-layer perception(MLP). Bag-of-words representations based on either word- or character- level n-grams was applied on the data preprocessing. The cross-entropy was defined as the loss function.They evaluated N-gram type, label type and decided to use word-bag and ED method is the best in Wiki Online Assessment. In data fetching part, how to identify the offensive comment is an important question. The process invovles:

- 1. generating a corpus of Wikipedia discussion comments,
- 2. choosing a question for eliciting human judgments,
- 3. selecting a subset of the discussion corpus to label,
- 4. designing a strategy for eliciting reliable labels.

In model part: They rely purely on features extracted from the comment text instead of including featrues based on the authors' past behavior and the discussion context. They showed that simple n-gram features are more powerful than linguistic and synatactic features, hand-engineered lexicons, and word and paragraph embeddings. In all of the model architectures, they have a final softmax layer and use cross-entropy as the loss function. The cross-entropy function is defined as: $H(y, \hat{y}) = -\sum_i y_i log(\hat{y}_i)$. They have introduced a methodology for generating large-scale, longitudinal data on personal attacks in online discussions. After crowdsourcing the identification of personal attacks within a sample of discussion comments, machine learning classification is leveraged to scale the identification process to the whole corpus. They explored methods for aggregating multiple human judgments per comment into training labels, compared different model architectures and text features, and introduced a technique for comparing the performance of machine learning models to human annotators.

3 PROBLEM FORMULATION

In this project, we would like to improve the methods used in [5], which is the classification methods based on N-gram features, and discover new features from the given dataset to distinguish the attacking comments from others. Specifically, we intended to answer the following research questions:

- Q1: Can we use other models, such as Random Forest to detect attacking comments based on N-gram approach? How do they perform?
- Q2: Can we use some optimal features, such as linguistic features, to optimally detect attacking comments rather than N-gram features?
- Q3: How do the new features perform comparing with N-gram features, in terms of running time or memory space required?
- Q4: Which classification methods should be used to apply such new features? Which metric should be used to evaluate the performance of each method?

We give the following terms for easily formulating our research problem. Let G_C be the global dataset of online comments, which consists of attacking comments set A_C and non-attacking comments set N_C. Let U be the annotators set and L be the label set given by these annotators for comments. $L_{ij} = 1$ means annotator i labeled comment j as an attacking comment. Now, we define the attacking comments set $A_C = \{c|cinG_C, \frac{|L_{c\_}=1|}{|L_{c\_}|} >= 0.5\}$, and non-attacking comments set $N_C = G_C \setminus A_C$. Let F_N and F_L be the N-gram features set and linguistic features set of comment in G_C, respectively.

Problem Formulation Split global set G_C into training and testing dataset. The aims of this project are (a) using multiple classification methods based on N-gram features in F_N (*Research Question 1*); (b) discovering linguistic features in F_L using NLP analysor (*Research Question 2&3*); (c) using multiple classification methods based on linguistic features in F_L (*Research Question 4*).

4 DATA DESCRIPTION

Our data sets consist of 150k comments in total and each is labeled with 'attack' or 'non-attack'. The maximum length of comments is 400 hundred words. Some comments contain emoji and special characters. However, how we label each comments is a challenge task. Different people may judge the same comment differently. Unsurprisingly, we see this in the annotation data: most comments do not have a unanimous set of judgments, and the fraction of annotators who think a comment is an attack differs across comments. The set of annotations per comment naturally forms an approximate empirical distribution (ED) over opinions of whether the comment is an attack. A comment considered a personal attack by 7 of 10 annotators can thus be given a true label of [0.3, 0.7] instead of [0,1]. Using ED labels is motivated by the intuition that comments for which 100% of annotators think it is an attack are probably different in nature from comments where only 60% of annotators consider it so. In the case of a model trained on ED labels, the attack score represents the predicted fraction of annotators who would consider the comment an attack. In the case of a model trained on OH labels, the attack score represents the probability that the majority of annotators would consider the comment an attack. We compared OH and ED label approaches and decided to divide comments into two class. We followed the standard descibed in the previous section.

5 PROPOSED SOLUTION

To answer the Research Question (1) to (4), we design and run the experiments as four parts: Data Annotation, Processing N-gram Features, Processing Linguistic Features, and Selecting Classification Methods, shown as following.

5.1 Data Annotation

In this part, we fully follow the same processes as the paper [5] used. As we described in Section 4, we have over 100k discussion comments from Wikipedia in English, where every comment has been labeled by around 10 annotators on whether it is a personal attack or not. Based on the definition we shown on Section 3, we define attacking comments as those who have be annotated as personal attacking more than half of annotation times. After these processes, there are around 14,032 comments annotated as personal attacks out of 115,864 comments in total. Details are shown as Table 1.

Table 1: WikiPedia Comments Dataset Statistics

Set Name	Size	Fraction
Attacking Comments (A_C)	14,032	12.11%
Non-Attacking Comments (N_C)	101,832	87.89%
Global Comments (G_C)	115,864	100%

5.2 Processing N-gram Features

In this part, we implement the experiments to process the N-gram features of each comment as paper [5] did. We split the comments into word-bag and generate matrix to calculate N-gram vectors. However, the matrix is too big to compute. Figure 1 shous the word appeared frequency and its fractions.

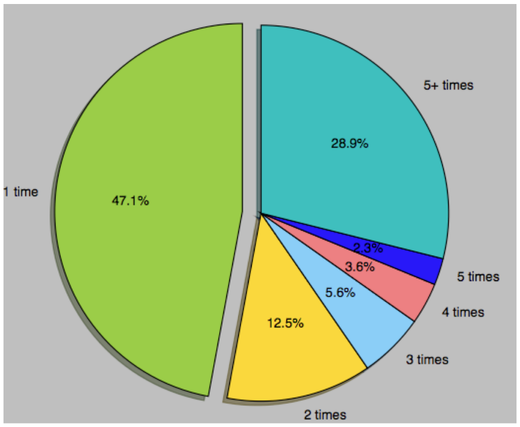

Figure 1: Word Occurrence and their Fraction

Therefore, we analyzed the word-bag and found that words with one appearance are likely to be typos. That is to say, we can simply filter those unnecessary but large set of words. In order to compress more on the matrix, we applied regex and porter stemming on the words. This helps us reduce a 160k words bag to 40k word-bag. Even though, the matrix size is still too large for personal computer. The description of N-gram features is shown on Table 2, including the feature size and required memory size after each kind of processing.

Table 2: N-gram Features of WikiPedia Comments

Preprocessing	Feature Size	Memory
Original N-gram	$\tilde{1}$60,000	$\tilde{1}$36 GB
Drop words appeared only 1 time	$\tilde{4}$0,000	$\tilde{3}$4 GB
Drop words appeared <= 2 times	$\tilde{2}$6,000	$\tilde{2}$3 GB

5.3 Processing Linguistic Features

As Table 2 shows, the N-gram features are always over the size of client devices' memory. Therefore, we would like to discover new NLP features of the WikiPedia comments to help the classifications. We used four group of advanced NLP features as follows.

Language Patterns (LP). Recalling the definition of the online personal attacking, we hypothesize that the attacking comments should have some patterns due to user's writing habits, like using excessive punctuations. We introduced Part-of-Speech Tagging [1] to analyze the language patterns of these comments.

Communication Sentiment (CS). We make a hypothesis that attacking comments include a certain set of emotions to fully express and vent writers' feelings. To understand the sentiment of these comments, we use the IBM Watson Tone Analyzer [2]. *Emotion*, a subset of these features, shows the likelihood of a writer being perceived as angry, disgust, fear, joy and sadness. Another subset of features, *Language Style*, shows the writer's reasoning and analytical attitude about things, degree of certainty and inhibition. And the feature set *Social tendency* will help us to prove our hypotheses that this kind of people have specific social properties in terms of openness, conscientiousness, etc.

Figure 2 and 3 show the features of emotion and social tendency of WikiPedia comments. We can see the clear difference between attacking comments and non-attacking comments, e.g. in terms of *Anger*, *Disgust*, *Emotional Range*, etc.

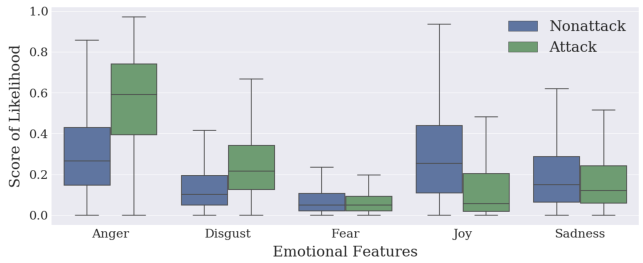

Figure 2: Sentiment Analysis: Emotions

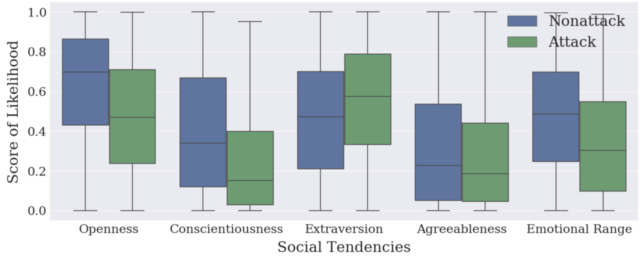

Figure 3: Sentiment Analysis: Social Tendencies

Content Relevance (CR). Instead of using N-gram (Word2Vec) features, due to its huge size of word matrix, here we try Doc2Vec [4] for learning a distributed representation [3], using either hierarchical softmax or negative sampling. The Doc2Vec model return a vector (size of 100) for each comments and we can use such vector to compare the similarity/distance between two comments.

Latent Topics (LT). To analyze the latent topic of personal attacking comments, we apply the LDA model [4], which allows both LDA model estimation from a training corpus and inference of topic distribution on new, unseen documents. We set the hyper-parameter $\#topics = 20$ so that the model can return a vector of likelihoods of each topic a tweet belongs to.

Table 3 shows the linguistic features we used in our experiment and their size. It is obvious that the new linguistic features have much smaller size than the N-gram features. In Section 6, we will show the performance of classifiers using such linguistic features and compare them with classifier using original N-gram features.

Table 3: Linguistic Features of WikiPedia Comments

Feature Name	Abbr	Feature Size
Language Structure	LS	100
Communication Sentiment	CS	14
Content Relevance	CR	25
Latent Topic	LT	20
Total (LS+CS+CR+LT)	all	159

5.4 Classification Methods Selection

We test four classification algorithms: *Logistic Regression*, and *Support Vector Machine (SVM)*, *Random Forest*, and *Multi-Layer Perceptron*. We alter various settings of each classification algorithm: we try 2 split criteria in random forest ("gini," "entropy"), 4 solvers in logistic regression ("newton-cg," "lbfgs," "liblinear," "sag"), and 4 kernels in SVM ("linear", "polynomial", "RBF", "sigmoid"). We also implement three popular measures for feature selection ("F-value", "Chi-square" (if applicable), and "Tree-based estimator" (if applicable)). To evaluate, we do 10-fold cross validation and measure the AUC score score. The reason why we use the AUC score for evaluating the performance of our classifiers is the dataset are unbalanced, recalling we have around 14,000 attacking comments and 100,000 non-attacking comments.

6 RESULTS & DISCUSSION

In the Section 5, we introduced our methodologies for the classification tasks. In this section, we briefly show our classification results and answer the four research questions mentioned in Section 4.

Q1: Can we use other models, such as Random Forest to detect attacking comments based on N-gram approach? How do they perform?

Using N-gram features as the input, we tested three different classification methods: Logistic Regression, Multi-Layer Perceptron, and Random Forest. The first two methods have been used in [5], and we added the third one for comparison and improvement. Table 4 shows the performance of these three classification methods. The reason why we did not get the exactly same performance as [5] is because we did not have a powerful enough machine to run all N-gram features. As we mentioned in Section 5.2, we dropped the words which only appeared once in the whole documents. Again, this fact gives a boost to us to use more optimal and sufficient linguistic features rather than using basic N-gram features to do the classification. Besides, we find that the classification method Random Forest performed better than other two which are used in [5] in terms of AUC score and related standard deviation.

Table 4: Classification Performance using N-gram Feature

	LR	MLP	RF
AUC (Std)	0.9145 (0.0050)	0.8915 (0.0076)	**0.9206 (0.0052)**

Q2: Can we use some optimal features, such as linguistic features, to optimally detect attacking comments rather than N-gram features? Yes, as we showed on Section 5.3, we used advanced NLP analysor to extract linguistic features of the comments, including four major groups: language structure, communication sentiment, content relevance, and latent topics. These features have much smaller size (159 in total) than the N-gram features (size more than 40,000) used in [5]. We will show the performances of classifications using these features and compare them with the classification using N-gram features in the following.

Q3: How do the new features perform comparing with N-gram features, in terms of running time or memory space required?

Table 5 shows the performance of classifications using linguistic features mentioned in Section 5.3. For each feature set and classification algorithm, we report the best result among all tested settings. According the classification results and parallel comparing the four group features, on the one hand, we see that communication sentiment plays the most important role in distinguishing the attacking comments from global dataset. On the other hand, adding feature latent topic slightly improved the performance (AUC score) from 0.9330 to 0.9356, which indicates that there is no obvious difference of topics between attacking comments and non-attacking comments.

Table 5: AUC scores of Attacks VS Non-Attacks on Wikipedia Dataset

	LR	SVM	RF	MLP
CS	0.8859	0.8752	0.8851	0.8953
LS	0.7120	0.6022	0.7178	0.7514
CR	0.9052	0.8982	0.8895	0.8875
LT	0.6277	0.6031	0.7997	0.6494
LS-CS	0.8389	0.8465	0.8943	0.8989
LS-CS-CR	0.8657	0.9038	0.9215	0.9330
LS-CR-LT	0.8034	0.8207	0.8799	0.8942
all	0.8646	0.9063	0.9244	**0.9356**

Q4: Which classification methods should be used to apply such new features? Which metric should be used to evaluate the performance of each method?

For the whole classification tasks, we used the AUC score as the metric to evaluate the performance of each classification method, since the data distribution are not balanced. According to the results shown in Table 5, we find that method MLP performs better than other methods when we combine all linguistic features together, which is reached $AUC = 0.9356$, which indicates that neural network is a good method to perform NLP text classification task.

Overall, the linguistic features performs super well in matrix compression, comparing with N-gram features. The linguistic feature size was cut down from 40k (N-gram) to 159, while the memory usage scales down from 36G (N-gram) to 0.34G. The advantage of this new approach is obvious. The original matrix is very sparse and a lot of computing resources are wasted due to the large number of zero. Beyond that, it relates different words inside each comments, which means the weights of single word are count less. Furthermore, the classification using linguistic features are performs better than using N-grams features. Table 6 shows the comparison of using linguistic features and N-grams features in terms of classification performance (AUC score) and required memory size.

Table 6: Comparison of using N-grams features and Linguistic features

Feature used	**Highest AUC**	**Memory Size**
N-grams	0.9206	36GB
Linguistic	0.9356	0.34GB

7 CONCLUSION & FUTURE WORK

In this project, we implemented Wulczyn's experiment, and tried one more classification model, Random Forest, which performs better than the two models used in their papers [5]. We designed a new approach for detecting personal attacking online using some reasonable linguistic features, such as language structure, communication sentiment, etc. Our new approach gained better classification performance (AUC score) than original N-gram approach. And, our new approach needed much less memory size. Input matrix scales dramatically and significantly decline from 110000*40000 to 110000*159.

Overall, the classification performance based on our experiments are slightly less than the results shown in [5], because 1) we did not use full size N-grams features due to the memory requirement; 2) their detail experiment settings are not mentioned in [5]. For the future work, we could research more on how to compress n-gram method. How to translate typos into correct forms? How to modify models to handle large size of inputs? Also, the paper indicates that the long short-term memory recurrent neural networks is a good method, which is also our future target.

REFERENCES

[1] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, 42–47.

[2] IBM. 2016. Watson Tone Analyzer, https://www.ibm.com/watson/services/tone-analyzer, Last Access: 10/10/2017.

[3] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 1188–1196.

[4] Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer.

[5] Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1391–1399.