Detection of Insulting Comments in Online Discussion

Wasi Uddin Ahmad
Department of Computer Science
University of Virginia
Charlottesville, VA 22904
wua4nw@virginia.edu

Md Masudur Rahman
Department of Computer Science
University of Virginia
Charlottesville, VA 22904
mr5ba@virginia.edu

Abstract

In this project, we aim to detect the comments that may appear insulting to other participants in an online discussion or conversation. We will use text processing techniques to generate features and then we will use supervised learning techniques to detect insulting vs non-insulting comments from the text. We are planning to use naive Bayes and logistic regression to set a baseline. After setting the baseline, we are aiming to use models like support vector machine, random forests. Finally we will use adaptive ensemble classifier to maximize accuracy. In order to avoid over-fitting, we will tune and evaluate our model via K-folds cross validation. The dataset is obtained from the popular data science competition portal, Kaggle.

1 Introduction

With the prosperity of the Internet, ample amount of ways or spaces for public discussion has emerged which changes how do people communicate with each other. Either it is a section for comments or reviews for a news article or a forum to discuss some aspects of a particular product or event, these online discussion gives us opportunity to share our opinion and findings, as well as to know about the thoughts of others. Even though these discussions are expected to be productive, it allows people to post insulting or inappropriate comments. As a result, these comments can hurt others feelings and often create a hostile or uncomfortable environment for some users, who might stop visiting the site in future. So, our aim is to focus on insulting comments of the online (ex. blog/forum) discussion.

However, the comments containing insults but are targeted to a non-participant of conversation (like a celebrity etc.) are not marked as insults. Insults are of many types like: Taunts, reference to handicaps, improper language, slurs and racism which are aimed to attack the other person in an online discussion. While some other type of insults which mainly aim to embarrass the reader (not an attack) like crude language, provocative words, sarcasm, indirect reference [4]. An effective solution to mitigate this problem is to build a system that can detect whether a comment is insulting or not. So, we are primarily interested in detecting comments that are intended to be insulting to other participants in an online discussion. The objective of this project is to do: build a machine learning system that can accurately classify online comments as insulting to other participants or not.

2 Previous solutions for the target task

Different attempts have been made to classify the insulting comments in online discussions. The work by Ellen Spertus [1] used static dictionary approach to build a feature vector for training but it suffers from high false positive rates. Another work by Altaf Mahmud et. al [2] used semantic rules but couldn't distinguish between the insults directed to non-participant and participant of conversation. The work by Razavi et. al [3] makes use of insulting and abusing language dictionary on top of bag-of words features in their proposed three-level classification machine learning model.

Priya Goyal and Kalra [4] used support vector machine and logistic regression to train their model in their work but there were some false positives. In the work by Heh [5], logistic regression and stochastic gradient descent is used for classification and training their parameter vector respectively. Prashant Ravi found SVM, the best classifier in terms of the most number of correctly classified instances in his work. So, in our work, we aim to build an efficient ensemble classifier excelling previous works to detect insults involving machine learning approaches.

3 Why is this related to machine learning?

As we all know machine learning explores the study and construction of algorithms for learning to do tasks. The learning that is being done is always based on some sort of observations or data, such as examples (the most common case in this course), direct experience, or instruction. So in general, machine learning is about learning to do better in the future based on what was experienced in the past.

The emphasis of machine learning is on automatic methods. In other words, the goal is to devise learning algorithms that do the learning automatically without human intervention or assistance. Since we are trying to build an automated system to detect insults, we need to particularly investigate and analyze different techniques in the field of machine learning to come up with an efficient and accurate classifier. So, this work is entirely based on machine learning techniques.

4 Proposed Method

The basic strategy of our work is depicted in the following flow chart. The following sub-sections discuss each step in detail.

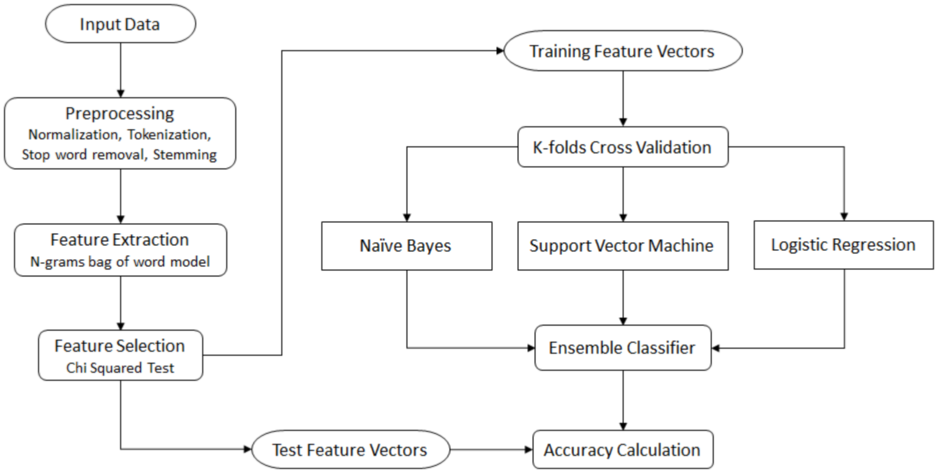

Figure 1: Flow chart of our classification approach

4.1 Preprocessing

The dataset obtained from kaggle cant be used directly for learning models. Some pre-processing of data is necessary and we need to convert it to the desired format of various machine learning algorithms. However, pre-processing should not lead to loss of information. So depending on the task, we have identified that the following pre-processing steps are required.

4.1.1 Normalization

We have used various techniques for this dataset after observing its nature. They are given below.

1. Removed punctuation symbols from the comments.
2. Removed unwanted strings like \\xa0, \\xc2, \\n, etc. and some unwanted html tags. These may put bias on the results if not removed.

4.1.2 Tokenization

Tokenization means slitting the text into tokens. Tokens can be characters, 'words' or n-grams which are the sequence of n consecutive words. We take words as tokens. So, far we have considered unigrams (words) and bigrams (2 consecutive words) for the feature vector construction.

4.1.3 Stop Words Removal

This is optional. We are going to test our model with and without removing stop words. Stop words usually refer to the most common words in a language and they do not carry significant information. We are using a popularly used list of stop words: Smart system's stop word list [8].

4.1.4 Correcting Common Words

People prefer to write short forms of words like ”ur” for ”your”, ”nope” for ”no” etc. If we can convert these types of words to their original correct form, then it reduces the size of feature set and also improves the accuracy of models. We got a dictionary of approximately 500 words containing all the possible ways an insult word can be written and when these words are encountered we convert then to their true form using this dictionary. [4]

4.1.5 Stemming

Stemming involves reducing the words to their root or stem form like ”running” to ”run”, ”beautiful” to ”beauti” etc. For large dataset this is necessary because otherwise it results in unnecessary increase in the number of features. However, for our project, this is optional because this might result in the loss of information from dataset as it may wrongly reduce some words which effect insult detection to their root.

4.2 Feature Extraction

The strings should be converted to a numeric vector so that they can be used by machine learning algorithms. We use the various N-grams model to construct the feature vector. We have used the following two different measure to construct the feature vector.

4.2.1 Term Frequency

We have counted the frequency (number of times) of each token occurs in a comment. We constructed a (generally sparse) matrix of size N by V where N is the size of the training data (number of comments in our case) and V is the size of the vocabulary (the length of feature vector constructed over the whole training set using n-grams) representing all the text strings (comments in our case) where the number of occurrences of each token is a feature for that text string.

4.2.2 TF–IDF Weight

TF-IDF, short for term frequency-inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. We constructed a similar matrix of size N by V using tf–idf weight as we did using term frequency also.

4.3 Feature Selection

Considering N-grams model, we are getting features in the order of a hundred thousand which is too big a number to be handled efficiently by algorithms like SVM and Logistic Regression. So we need to select a few best features from of our set of features. We will apply a statistical test known as ”Chi Squared Test” to our feature matrix to select best k number of features where k will be a parameter for our model.

4.4 Model Selection

After constructing the feature vectors and extracting the top features, we have applied naive bayes algorithm to set a baseline. We also want to use SVM, logistic regression, Decision tree and random forests which is an ensemble learning method on the features. Based on our observation, finally we will develop an ensemble learning model of different machine learning algorithms to use majority vote to predicted target labels. This way we will be able to achieve better accuracy. In order to avoid over-fitting, we will tune and evaluate our model via K-folds cross validation.

5 Evaluation Metrics

To evaluate our proposed method, we will use F-measure which is a measure of a test's accuracy. It is also known as F_1 score or F-score [7]. It considers both the precision *p* and the recall *r* of the test to compute the score: *p* is the number of correct positive results divided by the number of all positive results, and *r* is the number of correct positive results divided by the number of positive results that should have been returned. The F_1 score can be interpreted as a weighted average of the precision and recall, where an F_1 score reaches its best value at 1 and worst at 0.

The traditional F-measure or balanced F-score (F_1 score) is the harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

6 Details of the dataset

We obtained data for training and testing from Kaggle [6], which is a popular website that hosts machine learning competitions. The training data set contains 3947 examples and the testing data set contains 2647 examples, each of which consists of the text of a particular comment and its desired label. A label of 1 represents an insulting comment, while a label of 0 represents a neutral comment. For instance, two examples from the training data set are:

- Text: ”Either you are fake or extremely stupid...maybe both...”, Label: 1
- Text: ”We afford what we HAVE to afford, Marco.”, Label: 0

In the training data, total 1049 of the examples are labeled as ”insulting”, while the remaining 2898 examples are labeled as ”neutral”. In the testing data, total 693 of the examples are labeled as ”insulting”, while the remaining 1954 examples are labeled as ”neutral”.

7 Why we are the right team for implementing this plan?

We are doing this project for our machine learning course. We are learning machine learning techniques and how they can be employed in different applications. To grasp these concepts cogently, we need to implement some of these techniques in a practical work. Through this project, we will be able to learn how ensemble classifier works and some of the well established machine learning techniques like support vector machine or logistic regression to handle text data. With the implementation of this project, we will improve our breadth in the field of machine learning.

References

[1] Ellen Spertus. 1997. Smokey: Automatic recognition of hostile messages. In Proceedings of the Ninth Conference on Innovative Applications of Artificial Intelligence, pages 1058 - 1065.

[2] Altaf Mahmud, Kazi Zubair Ahmed, and Mumit Khan, 2008. Detecting flames and insults in text. In Proceedings of the Sixth International Conference on Natural Language Processing.

[3] Amir H. Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. 2010 Offensive language detection using multi-level classification. In Proceedings of the 23rd Canadian Conference on Artificial Intelligence, pages 1627.

[4] Dr. Amitabha Mukherjee, Priya Goyal and Gaganpreet Singh Kalra. 2013. Peer-to-peer insult detection in online communities.

[5] Kevin Heh. 2013. Detection of insults in social commentary.

[6] For dataset – www.kaggle.com/c/detecting-insults-in-social-commentary/data

[7] F-score – https://en.wikipedia.org/wiki/F1_score

[8] Smart system's stopword list – http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/english.stop