Toxic Comment Challenge Milestone

Monica Bernard
monica.bernard@knights.ucf.edu

Andrew Stewart
ajstewart1205@knights.ucf.edu

1. MOTIVATION

This semester-long project covers the group's attempt at the "Toxic Comment Challenge". The team will implement two separate pipelines toward the achieving acceptable results metrics. The higher performing approach will be used as the team's submission in the competition.

This reports highlights and briefly explains the initial aspects and direction for the implementation procedure. Section 3 discusses the related work that has helped derived the solutions found in this project. Section 4 discusses the team's techniques for arriving at a solution for the problem. Section 5 discusses initial results of preliminary modeling tests and any open ended discussions or problems that are currently being worked upon. Section 6 discusses the team's expected outcomes and risk analysis. Section 7 discusses the further work scheduled for this project. Section 8 discusses the individual tasking and workload assigned to each team member. Section 9 presents a general summary of what has been discussed in this report. Section 10 lists all references for this project so far.

2. PROBLEM STATEMENT

Comment rating was mostly performed by humans and heavily biased machines to determine the toxicity of the comment. This rating also influenced the consequences of the comment. However, as the comments and their complexity grew exponentially compared to the availability of human raters and the incapable machines, the demand for a better classifier came to be. This led to the creation of the "Toxic Comment Challenge" hosted on Kaggle.com [1].

As stated in the overview in the Toxic Comment Classification Challenge on Kaggle.com, the gist of the challenge is to detect and identify different types of toxicity in a data set of comments pulled from Wikipedia. The main goal is to build a model capable of classifying the variable toxic comments better than the challenge's provided models. The comments are expected to be classified into the following labels, which define the level of toxicity of the comment.

- toxic
- severe_toxic
- obscene
- threat
- insult
- identity_hate

A graph of the data labels is provided in Figure 1 to highlight their distribution. The results of the team's model will be measured against the competition's metrics to determine a ranking of accuracy against all submissions. The source code for this project is hosted on Github (https://github.com/drewart1205/toxic_comment_challenge).

3. RELATED WORK

The challenge contains a multitude of different approaches and work [2][4]. Weighing the successes and failures of a few Kaggle submissions, influenced the team's approach to a well defined pipeline. While similarities may be seen in preprocessing, the training shall be different in order to highlight the advantage and/or disadvantage of the team's approach. Given the influence of previous Kaggle submissions, the outcome of this project are expected to achieve similar results. However, this is not without the hopes of being in the positive end of a confidence interval of +5%/-5% for evaluation metrics.

4. DESIGN AND IMPLEMENTATION

Figure 2 shows the general pipeline for work and data flow that will be implemented in this project. The major components come down to the following subject areas: data preprocessing, word representation, model training, and validation/testing.

This project shall host two separate pipeline implementations where the team will compare the results to determine the most favorable outcome at the end of the semester. Figure 3 shows the first checkpoint's pipeline. In this checkpoint the team implemented a Continuous Bag of Words Vectorizer which performs data representation and further helps train a Naive Bayes classifier. Finally, the team validates the trained model on the test data and compares this to the evaluation metrics given in Section 5. The following discusses the general pipeline and the project-specific actions taken in each step.

ACM ISBN X-XXXXX-XX-X/XX/XX.
DOI:

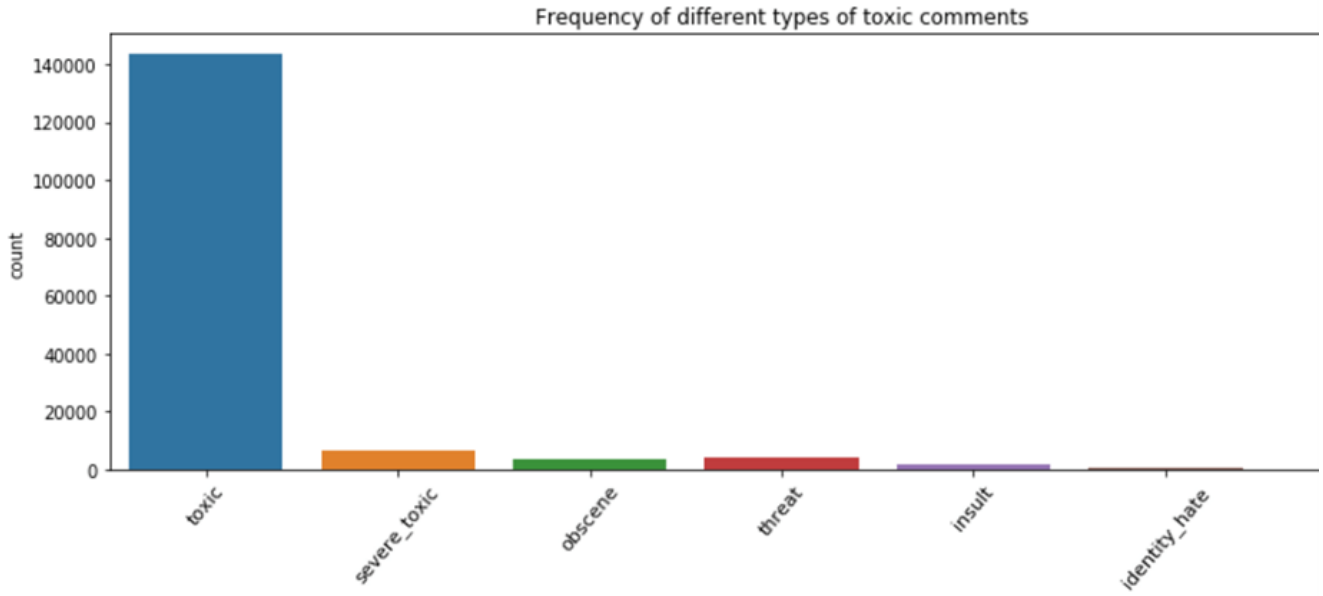

Figure 1: Data Distribution of Target Labels

The first stage in the pipeline involves preprocessing the given data (toxic comments). The team will clean the initial data set to be a refined input for the model. The preprocessing stage includes removing stop words such as 'the', 'in', 'and' so on and so forth. These words are removed as they do not add any value in determining final class label for the comments. Further, any sort of special characters and punctuation marks such as ",", "/", ":", so on and so forth are removed from the dataset.

Once the dataset is preprocessed, then next stage in the pipeline is to create a word representation. In this step, the team created a bag of words. More specifically, the implementation includes the use of a countVectorizer to create Bag-of-Words with a maximum of 5000-most frequently used words. This is done to avoid dataset sparsity and thus remove noise. Using a countVectorizer, text data is converted to numerical data which can further be mapped into output variables.

Next step in the pipeline is the train the model using the data. For this checkpoint, the team used a Multinomial Naive Bayes Classifier. A multi output classifier is used since each sentence has to be classified as one of the six output variables. The formula for this is shown in Equation 1[5]. i represents the actual even or label in this case. p is the probability of the given condition or label. x_i represents the frequency of the label. The output generated $p(x|C_k)$ gives the probability of the label given the input conditions.

$$p(x|C_k) = \frac{(\sigma_i x_i)!}{\Pi_i x_i!} \Pi_i p_{ki}^{x_i} \quad (1)$$

The final stage in the pipeline is to validate the trained model on a test dataset. The results for the test data are probability values assigned for the different class labels assigned. If the results are unfavorable, then some changes were made to the preprocessing and/or training phase. This is expected to be an iterative cycle until results reach the required acceptance values.

Probability predictions are done on the test data and dumped into a CSV submission file. This file will be used as the submission for the challenge, but only after both files for each checkpoint are compared and a clean winner is determined.

5. EVALUATION

To evaluate the model, the training set utilized contains about 160K comments and the test dataset contains about 153K comments. It was noticed that there is some class imbalance in the training data. There are six classification labels and they are toxic, severe toxic, obscene, threat, insult and identity hate. Out of all these class labels, toxic comments were the most common with about 10% frequency in the training dataset.Severe toxic, obscene, threat, insult and identity hate were 1%, 5%, 0.3%, 5% and 0.9% of the training dataset, respectively.

The dataset used has the data distribution shown in Figure 1. Even with a 20% training and 80% test split, the distribution still has a nearly similar skew of the toxic label to other labels.

To evaluate the performance of the implementation, Area Under the ROC Curve (AUC) metrics is utilized. Receiver Operating Characteristic (ROC) is a graph showing the performance of the classification. AUC measures the entire two-dimensional area under the ROC curve. The ROC-AUC score indicates the degree of separability/distinctness or the degree of crossover/interminable characteristics between the predictions. The higher the ROC-AUC score, the lower the crossover between the labels and thus better the classification [3].

Specifically, for this project both average accuracy and the mean ROC-AUC score were used as determining factors for evaluation. The validation criteria for this project is chosen as an accuracy over 70% and an mean AUC value over 60%. The team's Naive Bayes model was able to achieve an accuracy of 81.38% and a mean AUC value of about 0.6319.

6. EXPECTED OUTCOMES AND RISK MANAGEMENT

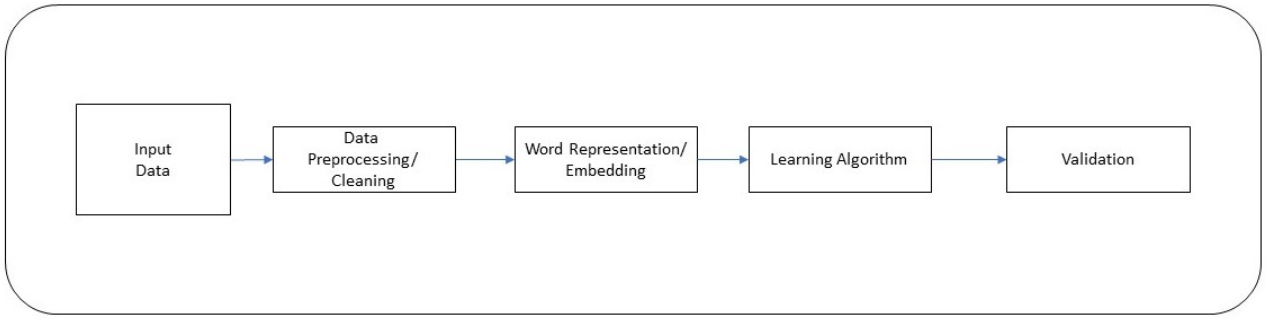

Figure 2: General Data and Workflow Pipeline

The team's main expectation/goal of the project is to produce an acceptable version of the Toxic Comment Classifier that performs more favorably than a human classifier. The team's goal was to produced a model to have a comparative accuracy similar to their influential counterparts.

One main risk for this project is the wrongful classification of a comment. A comment classified as a threat, an obscenity, or identity hate can have worse repercussions than a less toxic counter part. So if a comment classified as a threat is not actually a threat, then this can lead to possible systemic abuses. This is managed by the validation criteria addressed earlier in Section 5. The hope for this criteria selection is to minimize the possibility of false positives, therefore, lowering the possible occurrence of wrongfully classified comments.

7. FUTURE WORK

As previously stated, the team's plan for the next half of the project is to be able to implement the general pipeline once again but with a different representation and a model. This second pipeline is shown in Figure 4. For the next implementation of the pipeline, the team plans to use a word2Vec vectorizer, which will interpret the data and help train a Logistic Regression classifier. These are chosen hoping for a better performance both in the interpreting and classifying stage of the next pipeline implementation.

The main reason for the inclusion of two separate approaches is to observe and understand the differences in the results of the two representations and models. These observations will aid in the decision on which model to choose for submission. At the end of implementing another model, both models are to be compared to determine the representation and model that works better for the toxic comment challenge.

8. PLAN AND ROLES OF COLLABORATORS

The timeline and division of work for this project is shown in Figure 5. The "Coding" part of the project (development, training/testing, and individual model evaluations) will be performed by each member. Now that we've finished implementing the pipeline for the first checkpoint, the team plans on collectively working towards implementing the pipeline again for the next project submission. All "Write-Up" and model evaluation comparison work shall be split equally as seen fit by the team members.

Each team member dedicates at least 2 to 4 hours each week towards the project. During these hours, the team works on understanding the project at hand, performing research (reading research papers relating to the topic and going through other Kaggle submissions), developing and implementing the code, debugging errors, working on the project checkpoint paper, and planning the work ahead. The team hosts weekly phone calls for discussion on individual progress, updates to the project, and any open issues at hand.

Each milestone is planned/organized to be finished and submitted by each due date.

9. CONCLUSION

In summary, the project for this challenge involves two iterations of a general natural language processing pipeline. The first invokes Continuous Bag of Words representation with a Naives Bayes learning model. The second utilizes a word2Vec representation with a Logistic Regression learning model. The team has completed a few iterations of the Naives Bayes model presently with satisfying results. The Logistic Regression model is soon to be developed. The results of both are compared at the end to determine which will be used for submission for the challenge.

10. REFERENCES

[1] Kaggle.com *Toxic Comment Classification Challenge* https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

[2] Morgan, Adam *Fork of Basic word2vec using Gensim* https://www.kaggle.com/twistedtensor/fork-of-basic-word2vec-using-gensim

[3] Radečić, Dario *ROC and AUC — How to Evaluate Machine Learning Models in No Time* https://towardsdatascience.com/roc-and-auc-how-to-evaluate-machine-learning-models-in-no-time-fb2304c83a7f

[4] Sen, Anirban *JTCC_Bag_Of_Words* https://www.kaggle.com/anirbansen3027/jtcc-bag-of-words

[5] Wikipedia.com *Naives Bayes Classifier* https://en.wikipedia.org/wiki/Naive_Bayes_classifier

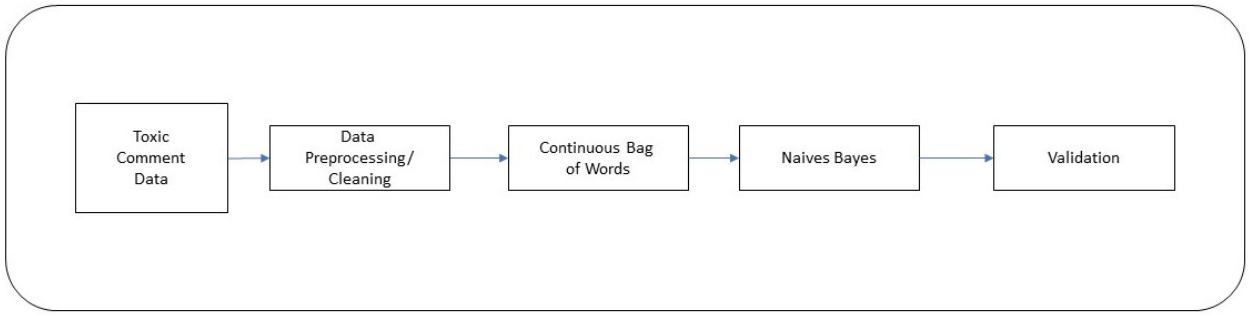

Figure 3: First Checkpoint Pipeline

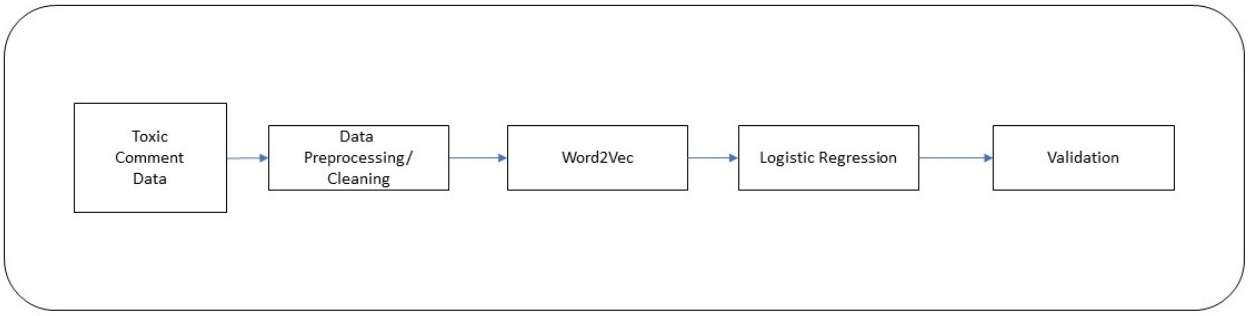

Figure 4: Second Checkpoint Pipeline

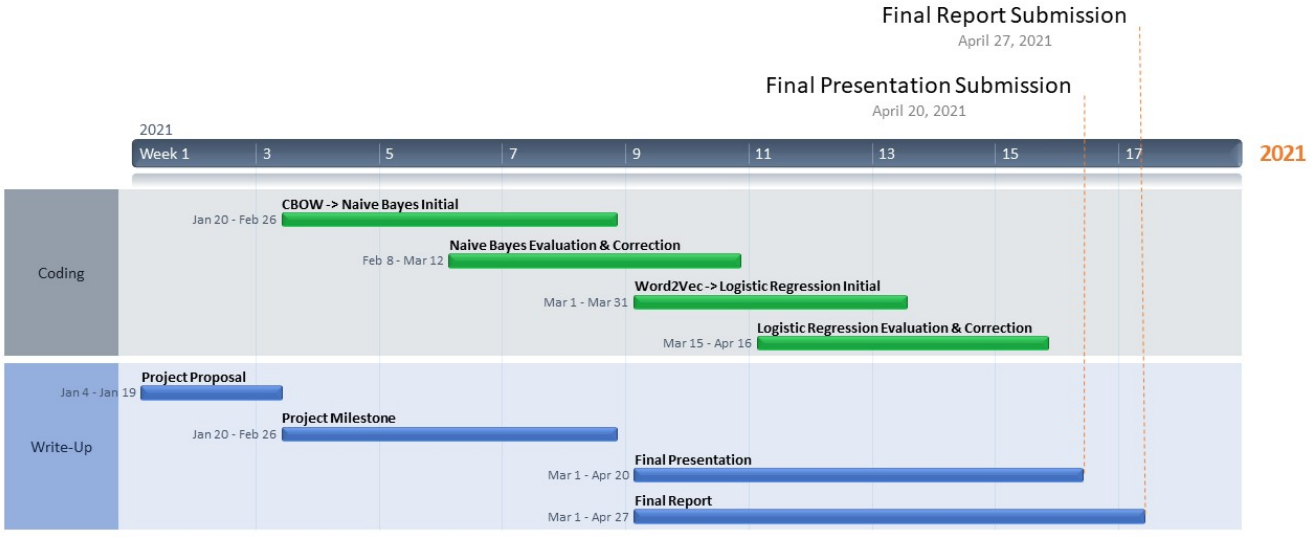

Figure 5: Gantt Chart for Project Timeline