NAME OF THE PROJECT

MALIGNANT-COMMENTS-CLASSIFIER

Submitted by:

VARSHA V. SHINDE

ACKNOWLEDGMENT

We first would like to thank god for giving us the opportunity to be here and make this project. Secondly, we thank our families for their unwavering support and great sacrifices in order to help us reach this moment. We would also like to thank the faculty of Mohd Kashif sir and FLIP ROBO for their work and efforts to give us the best education possible in order to reach our full potential, especially our SME Mr.Mohd Kashif for his guidance and assistance in not only this but Throughout our entire journey.

INDEX

INTRODUCTION

Problem Definition

Project Overview

Online platforms when used by normal people can only be comfortably used by them only when they feel that they can express themselves freely and without any reluctance. If they come across any kind of a malignant or toxic type of a reply which can also be a threat or an insult or any kind of harassment which makes them uncomfortable, they might defer to use the social media platform in future.

Thus, it becomes extremely essential for any organization or community to have an automated system which can efficiently identify and keep a track of all such comments and thus take any respective action for it, such as reporting or blocking the same to prevent any such kind of issues in the future. This is a huge concern as in this world, there are 7.7 billion people, and, out of these 7.7 billion, more than 3.5 billion people use some or the other form of online social media.

Which means that every one-in-three people uses social media platform? This problem thus can be eliminated as it falls under the category of Natural Language Processing. In this, we try to recognize the intention of the speaker by building a model that's capable of detecting different types of toxicity like threats, obscenity, insults, and identity-based hate. Moreover, it is crucial to handle any such kind of nuisance, to make a more user-friendly experience, only after which people can actually enjoy in participating in discussions with regard to online conversation.

Problem Statement

Given a number of tweets (in Twitter) or any kind of other comments, sentences or paragraphs being used as a comment by a user, our task is to identify the comment as whether it is a malignant comment or no. After that, when we have a collection of all the malignant comments, our main task is to classify the tweets or comments into one or more of the following categories – toxic, severe-toxic, obscene, threat, insult or identity-hate.

This problem thus comes under the category of multi-label classification problem. There is a difference between the traditional and very famous multi-class classification, and the one which we will be using, which is the multi-label classification. In a multi-class classification, each instance is classified into one of three or more classes, whereas, in a multi-label classification, multiple labels (such as – toxic, severe-toxic, obscene, threat, insult or identity hate) are to be predicted for the same instance. Multiple ways are there to approach this classification problem.

It can be done using –

→ Multi-label methods which belong to the problem transformation category: Label Power Set (LP), Binary Relevance and classifier chain.

→ Base and adapted algorithms like: J48 (Decision Tree), Naïve Bayes, k-Nearest-Neighbour (KNN), SMO (Support Vector Machines), and, BP-MLL neural networks. Further, out of the total dataset used for experimenting these algorithms.

Evaluation Metrics

- Label bases metrics include one-error, average precision, etc. These can be calculated for each labels, and then can be averaged for all without taking into account any relation between the labels if exists. Average Precision (AP): Average precision is a measure that combines recall and precision for ranked retrieval results.
- For one information need, he average precision is the mean of the precision scores after each relevant document is retrieved, where, and are the precision and recall at the threshold.
- Example based metrics include accuracy, hamming loss, etc.
- These are calculated for each of the examples and then averaged across the test set. Let – Accuracy is defined as the proportion of correctly predicted labels to the total no. of labels for each instance. Hamming-loss is defined as the symmetric difference between predicted and true labels, divided by the total no. of labels.
- When we have a look at the data, we observe that every 1 in 10 samples is toxic, every 1 in 50 samples is obscene and insulting, but the occurrences of sample being severe toxic, threat and identity hate is extremely rare. Thus, we have skewed data, and accuracy as metric will not give us the required results. Thus, we will be using hamming-loss as the evaluation metric.

ANALYSIS

Data Extraction

One of the most time-consuming tasks in Data science is collection and labelling of data. Even when we collect data, a major problem we face is the availability of useful data in the collected dataset. What we noticed was that if we gave the track parameter as (" ") that is space or "a" which is available in all comments, out of 100000 tweets collected around 14000 tweets were toxic which means that many of them were vague or non-toxic. We could not use this collected data for our analysis.

So, we collected data using mainly cuss words in the track parameter because of which almost all of the data we collected could be classified as one of the malignant comment classes and we could perform analysis on the predicted results. In order to achieve this result we performed the following steps:-

- First, we set the authorization using the following commands:
- We are then creating a twitter Stream as follows
- From the twitter Stream we will be filtering the data with the required keywords as follows
- I am extracting the following useful information from the twitter Stream.

ANALYTICAL PROBLEM FRAMING

Exploratory Visualisation

- From the first visualization we can observe that comments have varying lengths from within 200 up to 1200. The majority of comments have length up to 200, and as we move towards greater lengths, the number of comments keeps on falling. Since including very long length comments for training will increase the number of words manifold, it is important to set a threshold value for optimum results.
- From the second visualization, we can observe the number of words falling under the six different outcome labels toxic, severe toxic, obscene, etc. along with their lengths. Here also similar to the first plot, we observe that most of the abusive comments have lengths under 200, and this number falls with the length of comments.
- Based on these plots, taking comments having lengths up to 400 for training is a good estimation of our data and can be expected to give acceptable results on testing later. Hence before pre-processing, we will be removing all comments with length more than 400 which will serve as our threshold.

Algorithms & Techniques

As discussed in the Problem Statement section, any multi-label classification can be solved using either Problem Statement methods or Adaptation Algorithms.

PROBLEM TRANSFORMATION METHODS:

- Binary Relevance Method: This method does not take into account the interdependence of labels. Each label is solved separately like a single label classification problem. This is the simplest approach to be applied.
- Classifier Chain Method: In this method, the first classifier is trained on input data and then each of the next classifier is trained on the input space and previous classifier, and so on. Hence this method takes into account some interdependence between labels and input data. Some classifiers may show dependence such as toxic and severe toxic. Hence it is a fair deal to use this method.
- Label Power set Method: In this method, we consider all unique combinations of labels possible. Any one particular combination hence serves as a label, converting our multi label problem to a multi class classification problem. Considering our dataset, many comments are such that they have 0 for false labels all together and

many are such that obscene and insult are true together. Hence, this algorithm seems to be a good method to be applied.

Adaption Algorithms:

- MLKNN: This is the adapted multi label version of K-nearest neighbours. Similar to this classification algorithm is the BRkNNaClassifier and BRkNNvClassifier which are based on K-Nearest Neighbours Method. This algorithm proves to give superior performance in some datasets such as year's gene functional analysis, natural scene classification and automatic web page categorization. Since this is somewhat similar to the page categorization problem, it is expected to give acceptable results. However, the time complexity involved is large and therefore it will be preferable to train it on smaller dataset.

- BP-MLL Neural Networks: Back propagation Multi-label Neural Networks is an architecture that aims at minimizing pair-wise ranking error.

MODEL\S DEVELOPMENT AND EVALUATION

Benchmark Model As we proposed in our proposal, we will be using Support Vector Machine with radial basis kernel as the benchmark model (using Binary Relevance Method) Implementation of this model has been done along with other models using the Binary Relevance Method in the implementation section. Since we have a large dataset, other classifiers using the bag of words model such as the Multinomial and Gaussians are expected to work than this model. But, in practice, it performs quite well.

Methodology

Data Pre-processing

The following steps were taken to process the data:

→ A string without all punctuations to be prepared:

- The string library contains punctuation characters. This is imported and all numbers are appended to this string. Our comment text field contains strings such as won't, didn't, etc. which contain apostrophe character ('). To prevent these words from being converted to wont or didn't, the character ' represented as \' in escape sequence notation is replaced by empty character in the punctuation string.
- make_trans (in tab, out tab) function is used. It returns a translation table that maps each character in the intab into the character at the same position in the out tab string

→ Applying Count Vectorizer:

- To convert a string of words into a matrix of words with column headers represented by words and their values signifying the frequency of occurrence of the word Count Vectorizer is used.
- Stop words were accepted, convert to lowercase, and regular expression as its parameters. Here, we will be supplying our custom list of stop words created earlier

→ Splitting dataset into Training and Testing:

- Since the system was going out of memory using train_test_split, I had jumbled all the indexes in the beginning itself.
- The shuffle function defined here performs the task of assigning first 2/3rd values to train and remaining 1/3rd values to the test set.

CONCLUSION

To conclude this paper, we will do a quick review of all the work we have done so far since the beginning of this project. Originally, we first started to think about developing right away software to solve the given problem, but we quickly realized that we would have to investigate at least a little to know what has already been done in the topic: therefore, we realized a state-of-the-art of existing methods.

Then, after providing a comparison of the-then methods, we agreed on one, which was the Long Short Term Memory neural networks, first forking an existing solution and afterwards tried to apply our ideas onto it. Eventually, after uploading our solution here, we get a score of 0.9772, as we write those lines, and thus get a better score than almost 40% of the entrants of the competition.

As a general conclusion for this project, it was very rewarding for us to take part in this competition because it allowed us to use what we had learned in class directly onto real-life problems, but also to make our own research, to investigate and to try to come up with new ideas when we did not find anything that would suit what we wanted to achieve. It was for all of us an interesting first step into the world of research, and it provided us a hands-on experience of what can be done with machine learning techniques in general.

Our research has shown that harmful or toxic comments in the social media space have many negative impacts to society. The ability to readily and accurately identify comments as toxic could provide many benefits while mitigating the harm. Also, our research has shown the capability of readily available algorithms to be employed in such a way to address this challenge.

In our specific study, it was demonstrated that an LSTM solution provides substantial improvement in classification versus a baseline Naıve Bayes based solution. To recall, the True Positive Rate of LSTM was almost 20% higher than Naıve Bayes method.

Additionally, the followings are some suggested studies to be considered as future work in this area:

We suggest a plan to improve the NLP classifiers: first by using other algorithms which such as Support Vector Clustering (SVC) and Convolution Neural Networks (CNN); secondly, extend the classifiers to the overall goal of Kaggle competition which is multi-label classifiers. In the current study, the problem simplified into two classes but it worth to pursue a main goal which is 7 classes of comments.

- We also suggest using SVM for text processing and text classification. It requires a grid search for hyper-parameter tuning to get the best results.
- Using Other DNN techniques (CNN)) because some recently published papers such as have shown that CNN proves to have a very high performance for various NLP tasks

Free Form Visualization:-

The hamming-loss and log-loss of different models used. We plotted in 2 scatter plots:

- Thus if we compare all the models on hamming-loss: The best model will be LP-MultiNB i.e. Label Power set Model with Multinomial classifier

- If we compare all the models on log-loss: The best model will be BP-MLL model with params {nodes in hidden layer = 16, learning rate = 0.001, epochs = 10, batch size = 64}

Reflection

- This is the summary of the things we followed in this project:

- The first step involved collecting data and deciding what part of it is suitable for training: This step was extremely crucial since including only very small length comments would give poor results if the length was increased whereas including very long length comments would increase the number of words drastically, hence increasing the training time exponentially and causing system (jupyter kernel) to go out of memory and die eventually.

- The second major step was performing cleaning of data including punctuation removal, stop word removal, stemming and lemmatizing: This step was also crucial since the occurrence of similar origin words but having different spellings will intend to give similar classification, but computer cannot recognize this on its own. Hence, this step helped to a large extent in both removing and modifying existing words.

- The third step was choosing models to train on: Since I had a wide variety of models(3 for problem transformation) and classifiers(not bounded) along with number of adaptation models in BP-MLL, selecting which all models to train and test took lots of efforts.

- Finally comparing on the basis of different evaluation metrics: The two major evaluation metrics I planned to compare on were hamming-loss and log-loss. Hence the final model selection was done on the basis of the combination of both these losses.

Improvement:-

- Although we have tried quite several parameters in refining my model, there can exist a better model which gives greater accuracy.

- Yes. We were unable to find a clear implementation of the Adaboost.MH decision tree model which we had planned to use. The scikit-multilearn library doesn't even mention of such a model. Also, the research papers were a little vague regarding implementation details.

Future Scope:-

The current project predicts the type or toxicity in the comment. We are planning to add the following features in the future:

- Analyse which age group is being toxic towards a particular group or brand.
- Add feature to automatically sensitize words which are classified as toxic.
- Automatically send alerts to the concerned authority if threats are classified as severe.
- Build a feedback loop to further increase the efficiency of the model.
- Handle mistakes and short forms of words to get better accuracy of the result.