TOXIC COMMENT CLASSIFICATION

Project Proposal

Prepared by : Nupur Baghel
January 14th, 2018

INTRODUCTION

People express themselves freely and without reluctance at online platforms only when they feel comfortable. Any threat of abuse or harassment will make them leave the conversation and prohibit them from participating in any good conversations in future. It is hence, a vital requirement for any organisation or community to have an automated system which can identify such toxic comments and report/block the same immediately.

This problem falls under the category of Natural Language Processing where we try to identify the intention of the speaker, and act accordingly. Research on modelling a solution to this problem has already begun. I personally feel, it is important to handle any such nuisance and create a more user friendly experience with regard to online conversation for each one of us.
The idea of the project and the dataset has been taken from kaggle. It is being conducted as research by the Conversation AI team, an initiative by Jigsaw and Google.

PROBLEM STATEMENT

Given a group of sentences or paragraphs, which was used as a comment by a user in an online platform, our task is to classify it to belong to one or more of the following categories - toxic, severe-toxic, obscene, threat, insult or identity-hate with approximate probabilities or discrete values (0/1). Hence it is clearly a Multi-label classification problem. Using labelled dataset, we can determine the accuracy, hamming-loss, etc of our model.

DATASETS AND INPUTS

The dataset consists of the following fields-

- id : An 8-digit integer value, to identify the person who had written this comment
- comment_text : A multi-line text field containing the exact comment
- toxic : binary label containing 0/1 (0 for no and 1 for yes)
- severe_toxic : binary label containing 0/1
- obscene : binary label containing 0/1
- threat : binary label containing 0/1
- insult : binary label containing 0/1
- identity_hate : binary label containing 0/1

Out of these fields, the comment_text field will be preprocessed and fitted into different classifiers to predict whether it belongs to one or more of the labels/outcome variables (i.e. toxic, severe_toxic, obscene, threat, insult and identity_hate).

I will be dividing the original **training** dataset (from kaggle) having a total of 65534 samples into training, testing and validation. The original **testing** dataset will not be used, since it contains unlabelled data (only id and comment_text), which cannot be tested for performance.

SOLUTION STATEMENT

There can be multiple ways to approach this classification problem such as using-

- Problem transformation methods like binary relevance method, label power set, classifier chain and random k-label sets (RAKEL) algorithm
- Adapted algorithms like the AdaBoost MH, AdaBoost MR, k-nearest neighbours, decision trees and BP-MLL neural networks.

My aim will be to analyse 2 or more of these models, with at least one from each category and determine their performance using label-based, example-based metrics, etc.

BENCHMARK MODEL

A support vector machine with a radial basis kernel will be used as the bench mark model. We will also be using this model as the base model on which our transformation and adapted algorithms will be applied.

A number of papers have already been published to compare different models but these have been evaluated on Java-based softwares Mulan and Meka. My aim will be to use scikit-learn or any other python-based library. Also, this dataset is new and unexplored, and hence will serve an important purpose with regard to good online conversations.

EVALUATION METRICS

Label based metrics include one-error, average precision, etc. These are calculated separately for each of the labels, and then averaged for all without taking into account any relation between the labels.

Example based metrics include accuracy, hamming loss, etc. These are calculated for each example and then averaged across the test set. Information about metrics has been obtained from this research paper. Let-

Yi - predicted subset of labels for the i^{th} instance

Zi - true subset of labels for the i^{th} instance

D - multi label dataset

L - full set of labels used in dataset

$$Accuracy = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Yi \cap Zi|}{|Yi \cup Zi|}$$

Accuracy is defined as the proportion of correctly predicted labels to the total no. of labels for each instance.

$$hamming\text{-}loss = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Yi \Delta Zi|}{|L|}$$

Hamming-loss is defined as the symmetric difference between predicted and true labels, divided by the total no. of labels.

On having a glance 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. Hence we have skewed data, and accuracy as metric will not give suitable results. Therefore, I plan to use hamming-loss as the evaluation metric.

PROJECT DESIGN & WORKFLOW

The project will comprise broadly the following steps -

STEP 1: PREPROCESSING

- Convert each comment_text field to lower case letters
- Remove any punctuation marks (?!#,.etc)
- Using count vectorizer, generate a vector showing the frequency of each word in a comment
- Divide the data into training and testing

STEP 2: CREATING METHODS FOR EVALUATION METRICS

- find_accuracy
- find_hamming_loss

STEP 3: WORKING WITH THE BENCHMARK MODEL (svm with radial kernel)

- Import the necessary files and create the model classifier
- Fit it to the training data using 5 fold cross-validation
- Predict using test data
- Calculate various evaluation metrics

STEP 4: USING TRANSFORMATION METHODS

- Create a method for binary-relevance method using benchmark model as base classifier
- Create a method for either LP (Label Power Set) or CC(classifier chain)
- Fit classifier to training data
- Predict labels
- Calculate various evaluation metrics

STEP 5: USING ADAPTATION ALGORITHMS

- Create classifiers for adaboost MH and MR models
- Try out different base classifiers for the above defined models
- Fit classifier to training data
- Predict labels and calculate various evaluation metrics

STEP 6: IMPLEMENTING BP-MLL NEURAL NETWORKS

- Study and define the basic model architecture
- Compile your model using optimisers and accuracy as metric
- Run the model for certain epochs and batch size
- Calculate accuracy

STEP 7: COMPILING RESULTS

- Compare the results for different models
- Decide which is the best model for given dataset

REFERENCES :

1. Wikipedia : https://en.wikipedia.org/wiki/Multi-label_classification
2. Kaggle challenge page for datasets and ideas : https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge
3. Conversation AI git page : https://conversationai.github.io/
4. Research Paper titled “Multi-label Classification: Problem Transformation methods in Tamil Phoneme classification” : https://ac.els-cdn.com/S1877050917319440/1-s2.0-S1877050917319440-main.pdf?_tid=eced1a38-f8fa-11e7-b8ef-00000aab0f27&acdnat=1515914406_0f244d3e6313bb049c435bf43504bd52
5. Research Paper titled “Benchmarking Multi-label Classification Algorithms” : http://ceur-ws.org/Vol-1751/AICS_2016_paper_33.pdf