Toxic Comment Classification Challenge

Project Design

Yangmei(Cathy) Deng

ydeng003@odu.edu

UIN:01124163

Table of Contents

Introduction **3**

Dataset Overview **3**

Executive Summary **3**

Design **3**

- **Data preprocess** 3
- **Model** 4
- **Evaluate model** 4

Disclaimer **4**

Reference **5**

Introduction

Toxic comments make people stop expressing themselves and give up on seeking different opinions online. To effectively facilitate conversations, platforms need tools to improve online conversation. Google and Conversation AI released the competition "Toxic Comment Classification Challenge"[1] on Kaggle last year, because the tool they were using still made errors. This project will build a multi-headed model that's capable of detecting different types of toxicity.

Dataset Overview

There are two csv datasets used in this project. The train.csv has 160k records, each record consists of a comment retrieved from Wikipedia, and 6 labels which were labeled by human raters for toxic behavior. The types of toxicity are: toxic, severe_toxic, obscene, threat, insult, identity_hate. The toxicity is not evenly spread out across classes. Almost 90% of the comments are clean.

The test.csv has 153k records. The model will predict a probability of each type of toxicity for each comment in the test.csv file.

Executive Summary

This project will use Hadoop mapreduce and pyspark to preprocess the comments and use two main methods for tackling this multi-label problem: problem transformation methods and adaptation methods.[2][3]

Design

Data preprocess

Design a series map reduce functions to preprocess each comment.

MapReduce 1: prepare comments, the basic algorithm is below (only include the input and output of the whole mapreduce function):

```
Load train.csv
For each comment:
                    Split into tokens
                    Convert to lowercase
                    Filter out tokens that are not alphabetic
                    Filter out tokens that are stop words
Return tokens for each comment
```

MapReduce 2: get Inverse Document Frequency(IDF) for each token, the basic algorithm is below (only include the input and output of the whole mapreduce function):

```
Take output of MapReduce 1 as input
Count number of comments containing each term
Calculate IDF for each term
Return (term, term's idf)
```

MapReduce 3: get Term Frequency(TF) for each term in each document, the basic algorithm is below (only include the input and output of the whole mapreduce function):

```
Take output of MapReduce 1 as input
Calculate TF for each term in each comment
Return ((comment, term), term's TF)
```

MapReduce 4: get TFIDF for each term in each document, the basic algorithm is below (only include the input and output of the whole mapreduce function):

```
Take output of MapReduce 3 as input, output of MapReduce2 as cache
file
Calculate TFIDF for each term in each comment
Return ((comment, term), term's TFIDF)
```

Model

Use pyspark to do machine learning.

Method 1: transformation methods. This method can be carried out in 3 different ways.

1. Binary Relevance: This method will treat each label as a separate single class classification problem. The union of all labels that are predicted will be the final output.
2. Classifier Chains: In this method, each classifier is trained on the input and all the previous classifiers in the chain.
3. Label Powerset: This approach will take all possible combinations of labels into account, which means it will needs worst case $2^6 = 64$ classifiers. This method has a high computational complexity. So we can take advantage of Apache Spark framework.

Method 2: adapted algorithm

This method will adapt the algorithm to directly perform multi-label prediction, rather than transforming the problem into subproblems.

Evaluate model

Use AUC, confusion matrix and accuracy_score to validate the model.

Disclaimer

Due to the early stage of the project, steps and task assignment might change with future knowledge development.

Reference

[1] Toxic Comment Classification Challenge

https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data

[2] Deep dive into multi-label classification..! (With detailed Case Study)

https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff

[3] Solving Multi-Label Classification problems (Case studies included)

https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification/