# Toxic Comment Classification - Report

Mick van Hulst Dennis Verheijen Roel van der burg Brian Westerweel
Joost Besseling

April 11, 2018

## 1 Problem statement

We have chosen the toxic comment classification challenge on Kaggle. For this challenge we have to classify about 160000 comment. There are 6 different classes and each comment can be labeled with any of these classes. This means our problem is a multilabel classification problem.

## 2 Dataset

The dataset that is provided by Kaggle consists of some 160000 comments with their respective class labels. They also provide a test set of about the same size, without the labels. Our task is to predict the labels of the test set.

One important characteristic of our data is that the set is very imbalanced (figure 1). This posed many challenges to us.

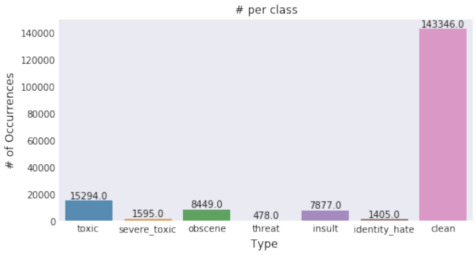

Figure 1: Figure from Kaggle [1]

### Data Preprocessing

Since the data consists of raw Wikipedia comments, we have to do some preprocessing to convert the words to lowercase and change words that are spelled erroneously to the correct spelling. We use the TweetTokenizer to handle this for us.

## 3 Models

In this section, the different approaches will be described. Starting with feature extraction, followed by Neural Network approaches including language modeling with an LSTM and a convolutional approach.

### 3.1 Feature Extraction

The first approach that was used was feature extraction. The goal was to extract meaningful features from the text to classify samples separately per class. The features were handcrafted, such that they are tangible features (i.e. not generated by a Neural Network).

#### 3.1.1 Feature Based Approach v1

In the first iteration of this approach, more generic features were used:

- Ratio of capitals vs total characters
- Ratio of punctuation characters
- Total length in characters, words and in sentences
- Total amount of some special characters: ?, (, ), ! and some other characters.
- Amount of unique words

In total, about ten features were generated. These features were used to train various classifiers, which will be described in section 3.1.3. Each model was evaluated separately. However, as may be noted

| Method | AUC |
|---|---|
| Feature Based Approach v1 | 0.59 |
| Feature Based Approach v2 | TBA |
| Convolutional Neural Network | 0.4902 |
| Vanilla LSTM | TBA |
| Biirectional LSTM | 0.96 |

Table 1: Summary of the achieved results, the Area Under the Curve (AUC) are computed using Kaggle.

from the results but in the end, none of these feature based models managed to produce convincing results.

### 3.1.2 Feature Based Approach v2

After a meeting with our supervisor, we thought that a problem with our feature extraction was that we might be using too little features. Since we are trying to predict 6 classes separately, and we are using a quite complex set, the dimensionality of the set is probably higher than 10. That is why we decided to introduce some extra features.

- For a list of swear words (since we are doing *toxic* comment classification), we added a feature denoting whether that particular word occurred in the comment.
- More features??

This greatly improved our results.

### 3.1.3 Classifiers

**Multilayer Perceptrons** The multilater perceptron (MLP) is a very simple neural network, using only fully connected (or dense) layers. In our case, the input consisted of the total number of features, and we used 6 units as output, each denoting the probability that that one of the six respective was active for the current sample.

We experimented with various configurations of this setup. We varied the number of layers, and the width of the hidden layers.

**Support Vector Machines** We only used a linear kernel, but the learning time of the SVM was so high, that we quickly decided not to investigate this approach further.

```
A forward and backwardsentence
sentence backward and forward A
```

Figure 2: An example of how the Bidirectional layer would feed the data to the LSTM

**Random Forests** The random forest classifier also didn't get good results on the small feature set, we have not YET tested it on the big feature set).

**1D Convolutional Network** Another approach we tried is the 1-dimensional convolutional network. However, because this network showed weak results (merely 0.4902 ROC AUC score) we decided to drop this approach.

## 3.2 Neural Networks

Having implemented these feature based approaches, we decided it might be better to use a neural network model. We decided to use the LSTM (Long Short Term Memory), because LSTM's are recurrent neural networks, so it can learn the context of words in sentences. First we tried to use the most simple LSTM we could think of.

After that, we found very well performing LSTM based approach on Kaggle (a bidirectional LSTM). This means that we feed the sentence twice to the LSTM, once normally, from front to back, and once flipped.

We implemented a 1 dimensional convolutional neural network.

# 4 Data Validation

DO we want to do something in this chapter? It is something we struugled with in the beginning?

Maybe it can be merged with the next steps chapter?

## 5 Next Steps

After the first competition, we have already learned many things. In this section we will briefly discuss a few things that we want to do differently than we did in the first competition.

**Ensemble Methods** During the first competition, we found out that ensemble models are extremely powerful for machine learning tasks (e.g. during the project presentations some groups used these). We have decided that we also want to use ensemble methods in the next competition.

**Collaboration** Our group was a bit slow in the beginning, so unfortunately we wasted a lot of time. Even later on, it was not clear for everyone what could and what should be done. That is why we have decided that a clear division of tasks, in smaller groups, will be beneficial to our results. We will also try to meet once a week.

Finally, we will try to utilize an existing git strategy, such as git flow. Now we are not using any strategy and everyone is committing to the master branch, but this leads to unnecessary conflicts and has a high chance of making merge mistakes.

## References

[1] Jagan, *Stop the S@#$ - Toxic Comments EDA*, https://www.kaggle.com/jagangupta/stop-the-s-toxic-comments-eda