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. This posed many challenges to us.

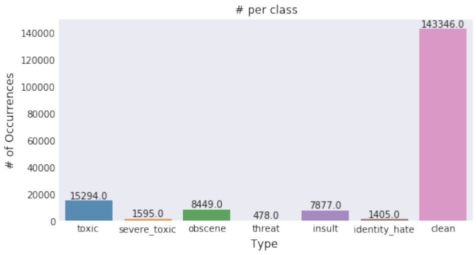

Figure 1: Figure from Kaggle [1]

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 Feature based models

3.1 Feature Extraction

Our first approach was to look at the data and try to extract some features by hand. These features include things like:

- 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 we produced about ten features.

We used these features to train various other models on. We will evaluate each model seperately, but in the end, none of these feature based models managed to produce convincing results.

3.2 Feature Extraction Part 2

After a meeting with our supervisor, we thought that a problem with our feature extraction was that we might be using to 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.

3.3 Models

TODO: include a table with the results of these models using the most up to date feature extraction (if someone has time left, use both feature extractions, to compare and see if there is much improvement)

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.

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.

4 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.

```
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

We implemented a 1 dimensional convolutional neural network.

5 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?

6 Reflection

After the first competition, we have already learned many things. In this section we will briefly discuss a few things that stood out during this competition.

Imbalanced dataset: The one thing that stood out from our dataset is that it was very imbalanced. This made it hard to test if our model was performing correctly as predicting that every comment was non-toxic already resulted in an accuracy of ≈ 0.96. This made it hard to train our models as during training time the model quickly returned a small loss.

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 After this first project we are more accustomed to each of our group members (i.e. knowing each other's strengths/weaknesses). This makes it easier for us to work together and enables us to divide our tasks accordingly. To efficiently work together, we're going to utilize an existing git strategy, such as git flow. We found great benefits in the weekly meetings and thus we're going to continue doing so.

References

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