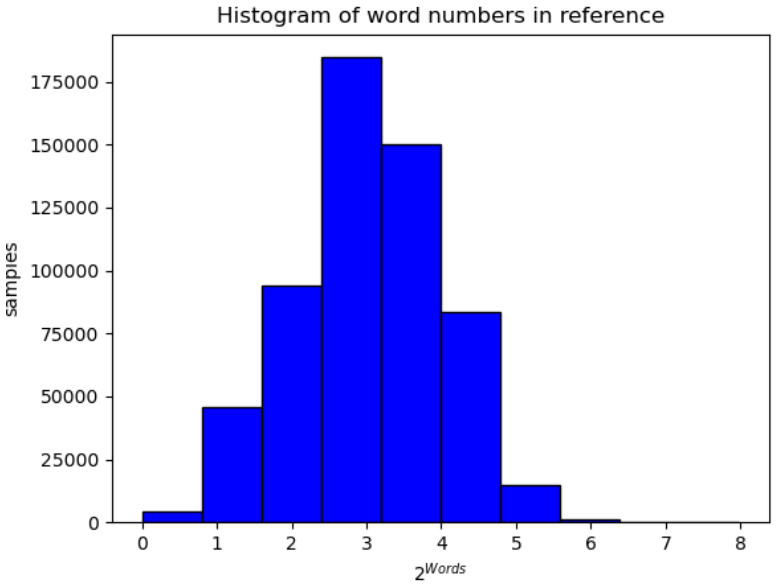
1. Exploratory Data Analysis

My first step was to analyze the data provided by the organizers (ParaNMT-detox). I decided to look at the graphs and derive useful information for further training. In this report I will provide only graphs, to see more information, which I printed in console please check *2.0-exploratory-data-analysis.ipynb* or run *visualize.py*.

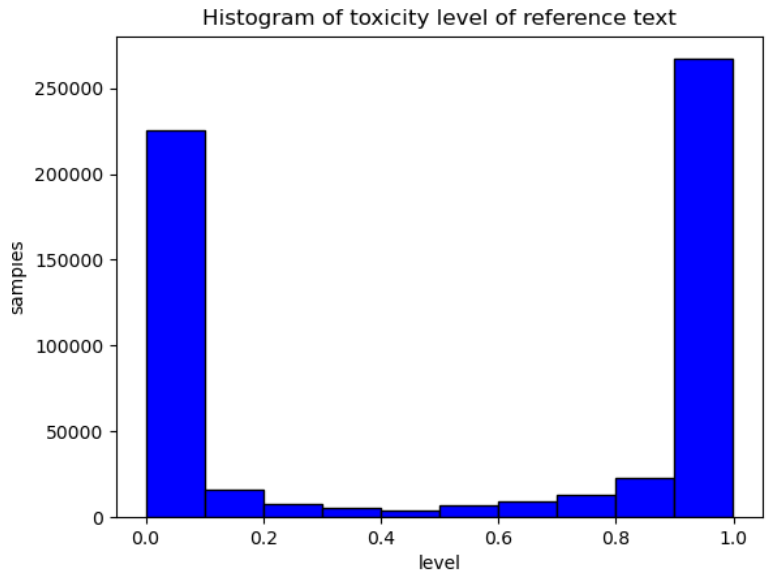
Dataset includes 577 777 samples with 6 columns (['reference', 'translation', 'similarity', 'lenght_diff', 'ref_tox', 'trn_tox']).

1.1 'Reference' column

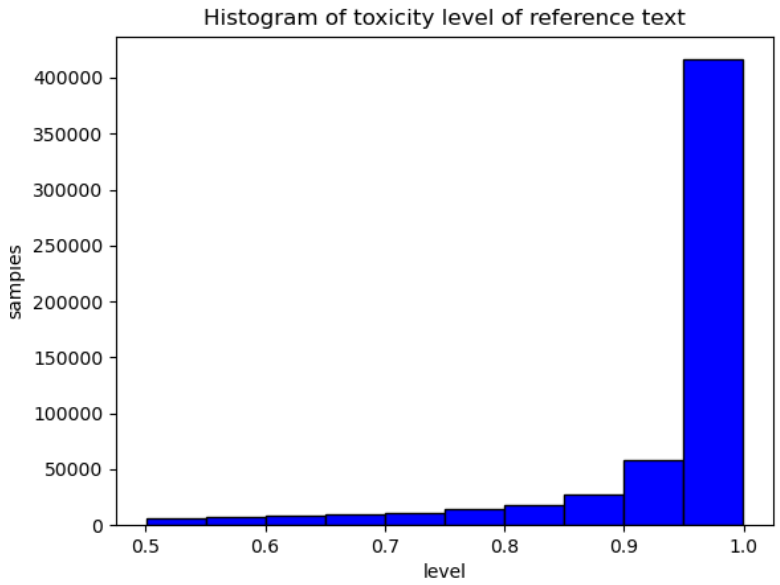
To begin with, I decided to analyze the 'reference' column. I broke the sentences into words to see some statistics. This will be useful when we choose the embedding size, for example. Here is the histogram of the distribution:



Then I checked the toxicity level of reference texts:



You can see that the ‘reference’ column has non-toxic sentences, i.e. level <= 0.5. So I decided to swap values between ‘reference’ column and ‘translation’, where level is <= 0.5. So I got this:



The next step is to check the most frequent words in toxic text. I did some preprocessing and plot wordcloud to see it.

Wordcloud toxic words ParaNMT

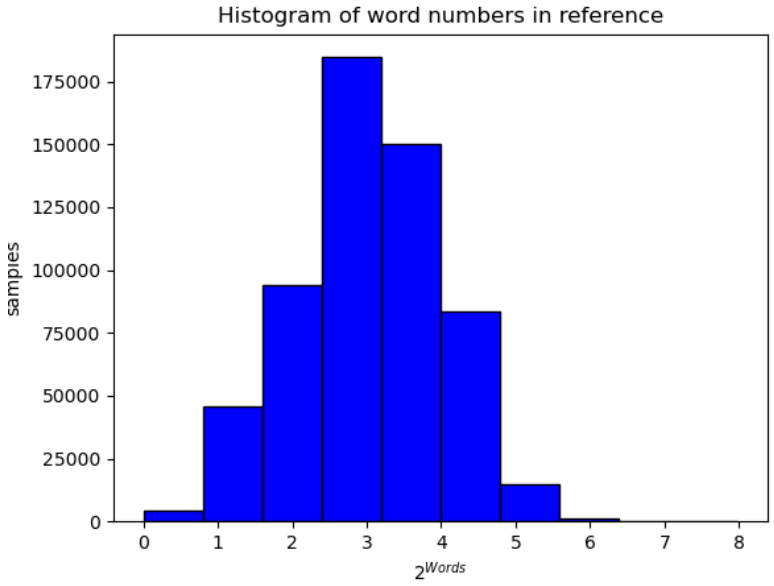
I also assumed that sentences may be repeated and there may be several translations for one toxic text. My guesses were confirmed:

Number of unique references with > 1 para: (33673,)
Number of unique references with == 1 para: (493639,)
AVG number of translation per reference with > 1 para: 2.498

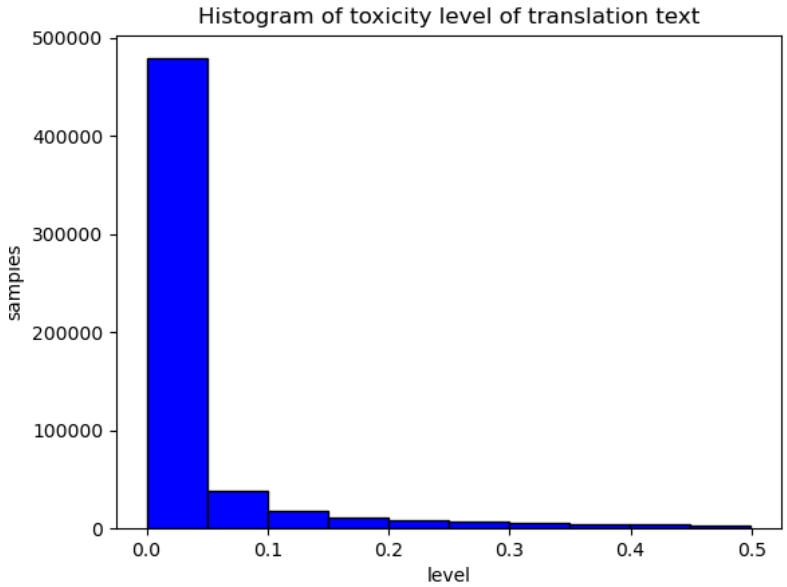
this means that you need to be careful with the metrics for assessing the quality of the model, since the model can be trained on one translation and tested on another

1.2 ‘Translation’ column

Here I did analysis in the same way as in 1.1 Section. I broke the sentences into words to see some statistics Here is the histogram of the distribution:



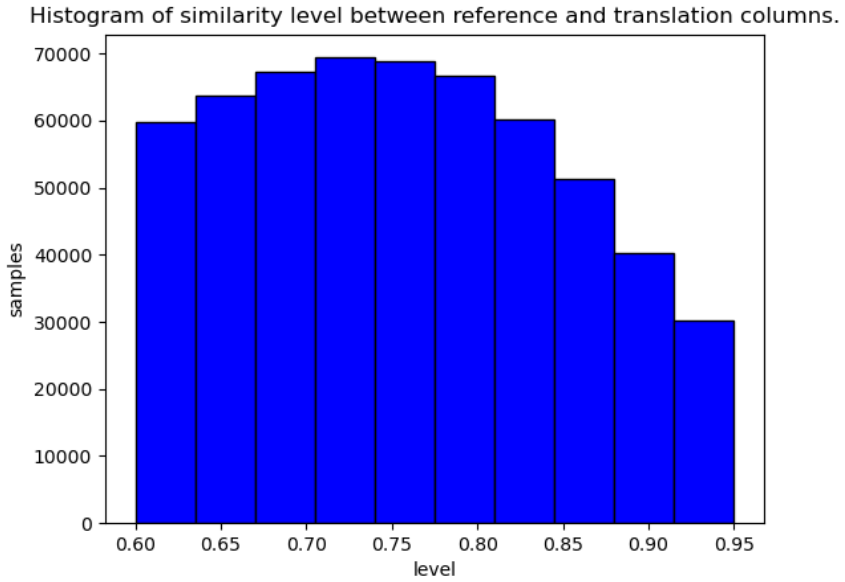
Then I checked the toxicity level of translated texts:



It’s okay here, since I already applied swapping values.

1.3 Similarity column

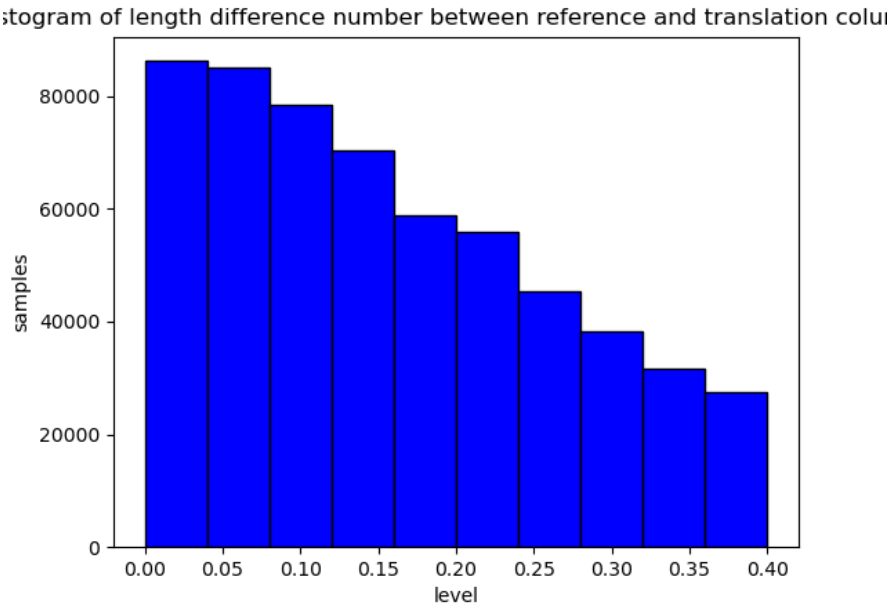
Column contains float numbers. Let’s see a histogram of the distribution:



Here we can see that they are pretty similar (> 0.6).

1.4 Length difference column

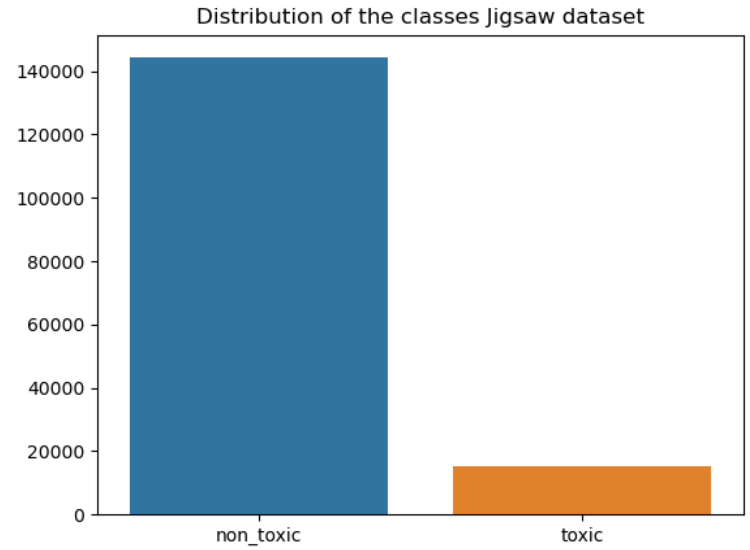
Column also contains float numbers. Let's see a histogram of the distribution:



Range of difference might be from 0% to 40% by symbols.

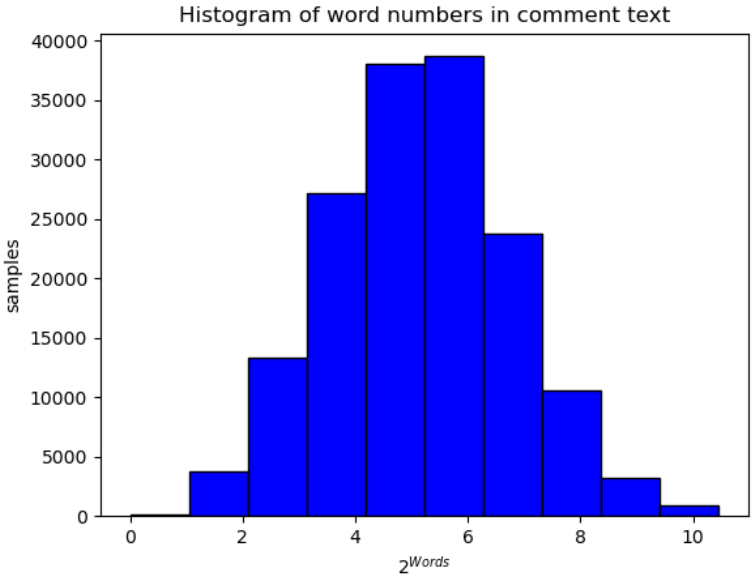
1.5 Jigsaw Dataset (External data)

Jigsaw dataset is a very popular dataset for identifying and classifying toxic online comments. I decided to take this dataset for a baseline solution, which you can check in the Final Report. Distribution of the classes.



As you can see, the data set is unbalanced. We will solve this problem in the future.

Let's see the distribution of the word numbers in the comment. I broke the comments into words to see some statistics. This will be useful when we choose the embedding size, for example.



I also plot wordcloud of toxic words to see most frequent words. After some preprocessing I got:

Wordcloud toxic words Jigsaw dataset