Hate speech classification

Paweł Flis, Karol Krupa, Rafał Chabasiński

April 2022

1 Introduction

Social media platforms such as Facebook, Twitter, Instagram and YouTube are experiencing a continuous growth in number of users. In recent years, partially due to COVID-19 pandemic even older generations turned to those platforms as a way to experience wider range of human interactions. Along with all the advantages social media brings, there are number of hazards related to harmful actions of some of its users encouraged by their perceived anonymity.

Especially recently hate speech has been under the scope of investigation to make the internet a safe and welcoming space for everyone. This topic garnered scientific attention due to the potential of applying machine learning for hate speech recognition, which would in turn help counteract cyberbullying. Although the very idea of using automatic censors is highly questionable their counselling can be of invaluable aid. In this case the hate speech detector accuracy is of the essence as too rigorous model could severely affect freedom of expression while high-tolerance one would maintain unsafe environments.

2 Literature analysis

Number of Natural Language Processing in conjunction with Deep Learning solutions have been proposed during the years, however majority of these solutions requires great amount of labelled data, which has to be manually labeled by a human, which eventually which limits the cognitive capabilities of the model. It is also really hard to identify semantic nuances which often decide on the overall sentiment. There also have been attempts to tackle the problem using statistical models such as n-grams, SVM or Logistic Regression. These methods require vector as an input, so the corpus of text has to be earlier converted - this is commonly done using methods such as word2vec encoder using semantic similarity of words or doc2vec for entire sentences.

The task at hand is hate speech detection in polish twitter data. Direct solution to the same problem were presented during PolEval2019 [5]. At a time statistical models such as SVM with tokenization or even Deep Learning solutions such as LSTM were able to achieve only minor improvements over baseline classifier which always predicts most common class.

Bidirectional Encoder Representations from Transformers model has shown a great prowess in various NLP tasks, in many cases outperforming its predecessors. Mozafari et al. [3] proposed a semi-supervised solution based on pre-trained BERT model with broad contextual knowledge and using transfer learning to fine tune a model on labelled samples from target domain of hate speech.

This approach considers two variants of BERT networks. $BERT_{base}$ having 12 layers and 110 million parameters and $BERT_{large}$ hacing double the layers and triple the parameters. The

output is a sentence representation in a 768-dimensional vector, which is an input to domain specific classifier.

In paper number of fine-tuning approaches are presented:

1. **BERT based fine-tuning**
   $BERT_{base}$ model with one fully connected layer at the top and softmax activation function with no hidden layers.

2. **Insert nonlinear layers**
   $BERT_{base}$ model with multilayer perceptron with two hidden layers at the top level.

3. **Insert Bi-LSTM layer**
   $BERT_{base}$ model with Bi-LSTM recurrent neural network and one fully connected layer on the output.

4. **Insert CNN layer**
   Output of each transformer from $BERT_{base}$ is concatenated to produce a matrix that is an input to CNN, later max pooling is performed and output is deduced using one fully connected layer once agian.

The authors performed data pre-processing to make the text input data more suitable for BERT model - emoticons were changed into <tags>, multiple word hashtags were separated and everything has been turned into lowercase. It is important to notice, that by doing that some relevant information may be lost, however we trust that it is a calculated decision and it is of benefit to used BERT model.

Version with CNN layer proved to perform the best across all metrics (Precision, Recalll, F1-score). It outperformed other solutions designed for the very same datasets used in [2] [7] [8]. It is worth noting that BERT with nonlinear layers performed surprisingly poorly(much worse than with no hidded layers) on dataset with samples labelled as Hate[1430] Offensive[19190] Neither[4163] so dataset heavily biased towards 'just' offensive content.

3 Exploratory data analysis

3.1 Data examples

- non-harmful
  - anonymizedaccount anonymizedaccount Zaległości były, ale ważne czy były wezwania do zapłaty z których się klub nie wywiązał.
  - anonymizedaccount Tym w sejmie też? Banda hipokrytów i złodziei.
- cyberbullying
  - anonymizedaccount anonymizedaccount Czas pokaże kto jest kim, mącicielu, szkodniku, haha do gazu haha
  - anonymizedaccount anonymizedaccount anonymizedaccount Pis mial być inny, jesteś idiotą
- hate-speech
  - RT anonymizedaccount anonymizedaccount Pasujecie do siebie antypolskie kanalie.
  - anonymizedaccount Pierwszy priorytet zadowolić Jarkacza, drugi priorytet ssać kasę wszystkimi otworami, mają rozdwojenie jaźni

We can see that subjects are anonymized and are easily recognizable, which will be helpful for our algorithms.

3.2 Label data split

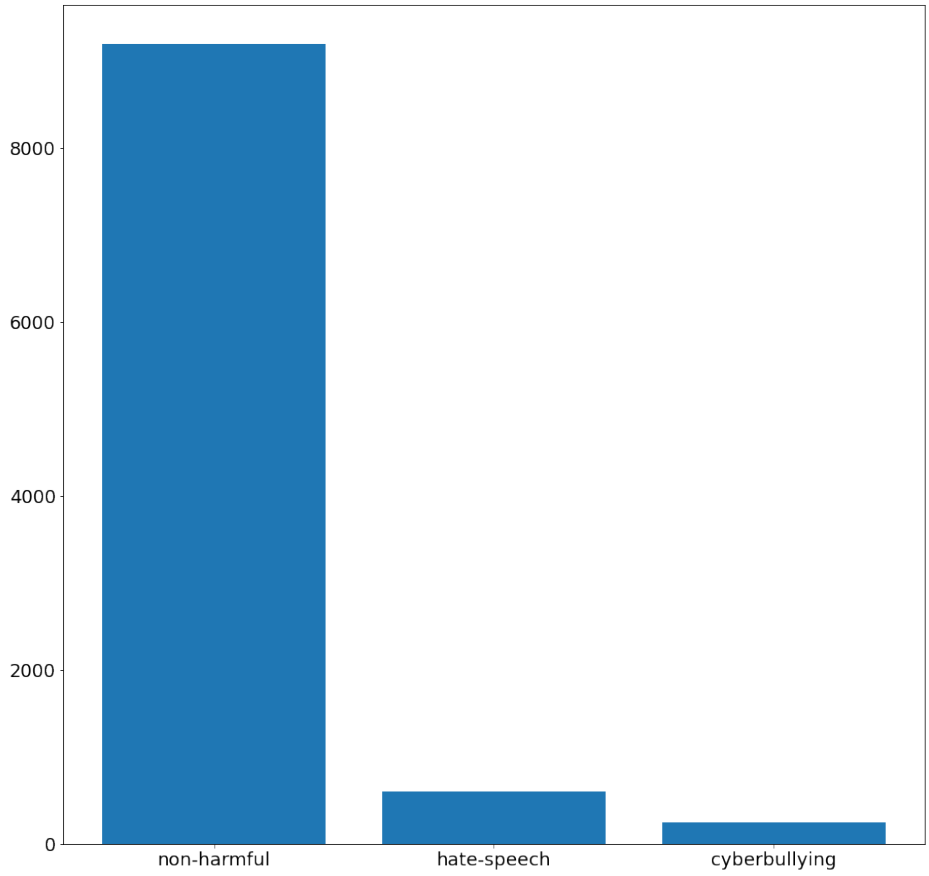

Figure 1: Labels distribution.

In 1 we can see that distribution of classes in the dataset is very unbalanced. We'll have to consider that when creating our algorithm, by e.g.

- shrinking bigger data sets to the smallest one
- applying bigger weights to smaller datasets

3.3 Data attributes

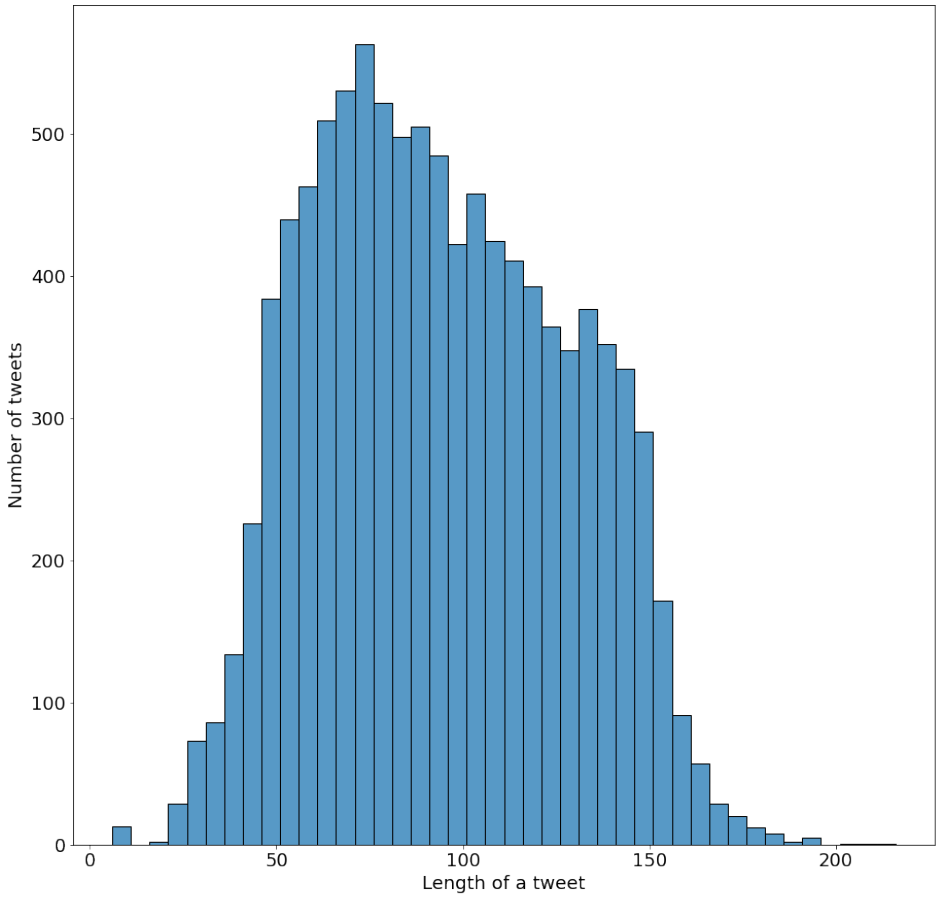

Figure 2: Tweets lengths (i.e number of characters in a tweet).

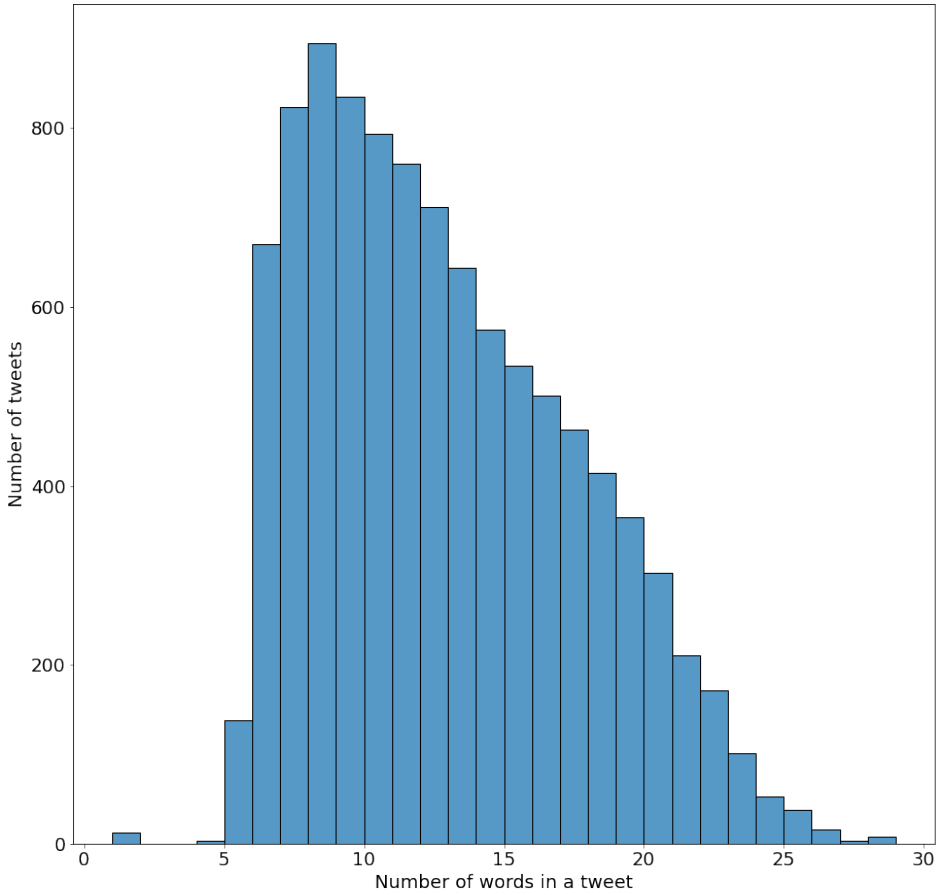

Figure 3: Tweets word counts.

In both 2 and 3 we can see some outliers in the data. In order to simplify our task we should consider removing them.

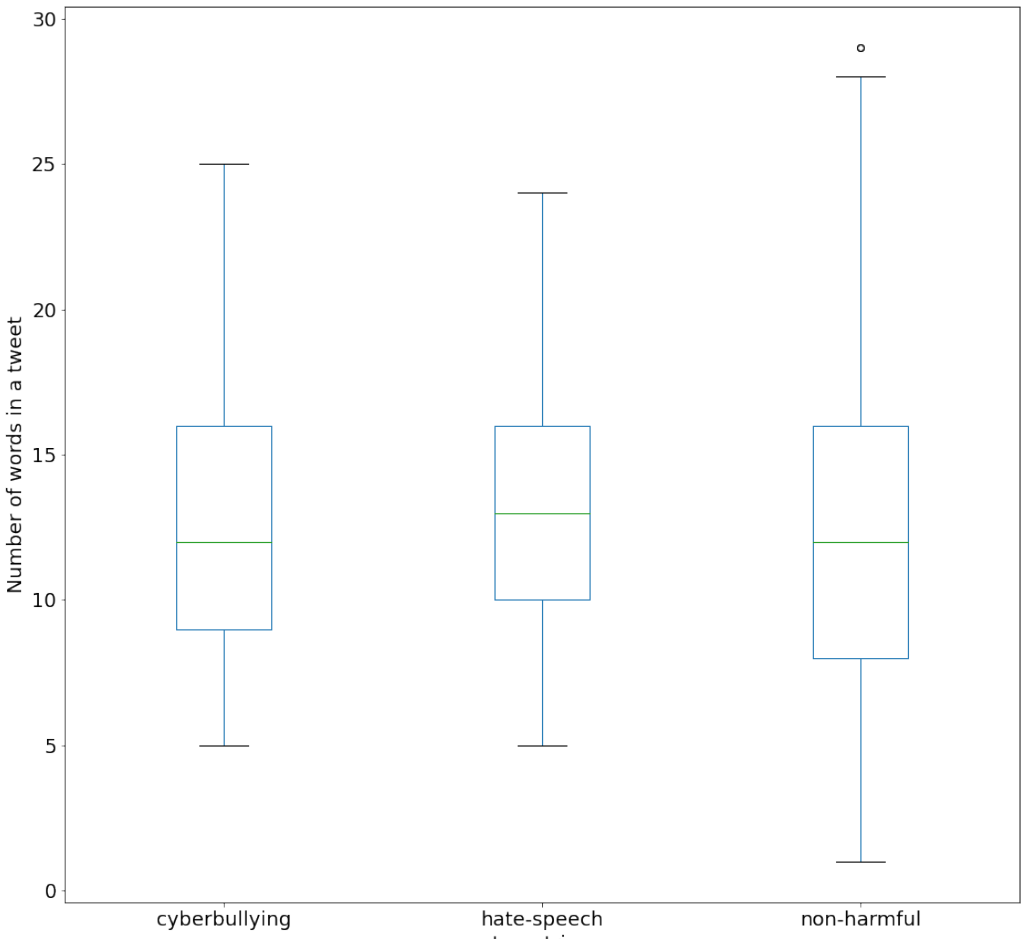

Figure 4: Tweets word counts box plot for different labels.

In 4 different labels have quite similiar expected values of number of words, so this statistic probably won't help us.

3.4 Word cloud

Figure 5: Word cloud.

In 5 we can see that the most important topics in out dataset are football and politics, there are also some curse words which may help distinguish hate speech.

4 Solution concept

Preparing our solution to the problem is to prepare a program that, given an expression written by a human, returns a result of 1 if the statement is hate-filled and 0 otherwise.

In order to perform this task as well as possible, we need to:

1. Properly transfer expressions and sentences into a form understandable by a computer - that is, vectorize them using an appropriate tokenizer.

2. Model the language and its internal dependencies using certain language models.

3. Train the language models for our task, which is hate speech recognition.

The first two steps can be solved by choosing a pretrained model. A collection of such models is available at this link KLEJ-Benchmark. The first two steps are strongly related, since the trained model has been trained for selected embeddings and is unlikely to work for others.

An example model from the list above is HerBERT [4] and XLM-RoBERTa [1]. Both score very high on the CBD task [6], which is the task we solve. For this reason, we decided to select pre-trained models and then retrain them for our purposes. In addition, for the second model, the authors provided scripts that give the ability to train the model to a specific KLEJ-benchmark problem, including the CBD problem. The scripts are available here: scripts.

The computations will be done with google cloud graphics cards, on the google collab site. Verification will be done using Precision, Recalll, F1-score metrics.

5 Solution

To solve the task, we used RoBERTa deep machine learning models along with scripts prepared to train them for the KLEJ benchmark prepared by Allegro. The available scripts are here.

During the development of the solution, we used the Collab Pro environment (Google's computing cloud) and the Tesla graphics card that was assigned to us there.

Solution preparation consisted of:

1. Prepare the training and testing collection for the CBD task format from the scripts described above. Their Internet hate detection task involved distinguishing between neutral and hate speech. In our case, we have 3 classes: hate, cyber-bullying, and neutral. Moreover, the sets for this task appear in two files: tags and text. Obtaining the intended format required combining these files in tag tab text format.

2. KLEJ benchmark script modification. The Task CBD described above had 2 categories and was not set up for post-training testing. Because of this, it required code modification. The modified tests.py file is available in the solution repository.

3. Downloading models on Google Collab.

4. Beginning model training.

5. Generating labels for test data.

6. Running evaluating script.

We tested two models while conducting the experiments. The first one is RoBERTa-large-v2, a model trained on a corpus of about 200GB of data. When training this model, we used the following python script call:

python run_tasks.py –arch roberta_large –model_dir roberta_large –train-epochs 2 –tasks KLEJ-CBD –fp16 True –max-sentences 8 –update-freq 4 –resample 0:1,1:1,2:1

The training process took approximately 320 seconds (2 epochs) on a Tesla P4 16GB.

The parameters mean that we used the roberta_large architecture, the model is in the roberta_large folder, we want to train the model for 2 epochs, the task we are doing is TASK-CBD, we want to do the calculations with less precision (meaning they will be done faster), the maximum number of sentences in the GPU at one time is 8, the update rate is 4, the input resampling is 1 for class 0 (neutral), 1 for class 1 (cyber-bullying), 1 for class 2 (hate).

Next, we tested the samplings as 1:15:10 and 0.75:5:3. In addition, we tested the performance of the RoBERTa-base model trained on a 20GB body, which is 10x smaller than the RoBERTa-large-v2.

	micro-average F-score	macro-average F-score
base-075-5-3	88.10	57.24
large-v2-075-5-3	90.00	58.63
large-v2-1-1-1	90.70	51.48
large-v2-1-15-10	89.50	54.21

Table 1: Results

The results obtained can be found in 1.

Accuracy was about 90% in each case, however, it is not very reliable due to the huge advantage of the class of neutral statements.

The results obtained by the script authors were 66.77% and 74.16% for the base model and the large model, respectively. Note that the task they performed is significantly simpler, because they only had to distinguish between negative and neutral statements. In our case there is a neutral statement and a negative statement in two variants, which is a more difficult task.

We conducted an experiment using the RoBERT-large-v2 model along with 0:0.75, 1:3 resampling (used by the authors) for a class 2 CBD task. The results obtained are: precision 76.64%, recall 61.19%, balanced F-score 68.05%, accuracy 92.3%, so much better than in case of solution for 3 class problem.

References

[1] Sławomir Dadas, Michał Perełkiewicz, and Rafał Poświata. "Pre-training Polish Transformer-Based Language Models at Scale". In: *Artificial Intelligence and Soft Computing*. Springer International Publishing, 2020, pp. 301–314. ISBN: 978-3-030-61534-5.

[2] Thomas Davidson et al. "Automated Hate Speech Detection and the Problem of Offensive Language". In: *ICWSM*. 2017.

[3] Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. "A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media". In: *COMPLEX NETWORKS*. 2019.

[4] Robert Mroczkowski et al. "HerBERT: Efficiently Pretrained Transformer-based Language Model for Polish". In: *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*. Kiyv, Ukraine: Association for Computational Linguistics, Apr. 2021, pp. 1–10. URL: https://www.aclweb.org/anthology/2021.bsnlp-1.1.

[5] Maciej Ogrodniczuk and Łukasz Kobyliński, eds. *Proceedings of the PolEval 2019 Workshop*. Warsaw, Poland: Institute of Computer Science, Polish Academy of Sciences, 2019. URL: http://2019.poleval.pl/files/poleval2019.pdf.

[6] Michal Ptaszynski, Agata Pieciukiewicz, and Paweł Dybała. "Results of the PolEval 2019 Shared Task 6: First Dataset and Open Shared Task for Automatic Cyberbullying Detection in Polish Twitter". In: Institute of Computer Science, Polish Academy of Sciences, 2019, p. 89.

[7] Zeerak Waseem and Dirk Hovy. "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter". In: *NAACL*. 2016.

[8] Zeerak Waseem, James Thorne, and Joachim Bingel. "Bridging the Gaps: Multi Task Learning for Domain Transfer of Hate Speech Detection". In: 2018.