LITHUANIAN HATE SPEECH CLASSIFICATION USING DEEP LEARNING METHODS

Eglė Kankevičiūtė1, Milita Songailaitė2, Bohdan Zhyhun3, Justina Mandravickaitė4
1,2,3,4Vytautas Magnus University, Kaunas, Lithuania
ORCID: https://orcid.org/0000-0002-9352-97091, https://orcid.org/0000-0003-4315-24612, https://orcid.org/0000-0002-1159-29753, https://orcid.org/0000-0001-9426-61654
E-mail: egle.kankeviciute@vdu.lt1, milita.songailaite@vdu.lt2, bohdan.zhyhun@vdu.lt3, justina.mandravickaite@vdu.lt4

Copyright © 2021 by author and the journal “Automation of technological and business – processes”.
This work is licensed under the Creative Commons Attribution International License (CC BY).
http://creativecommons.org/licenses/by/4.0

DOI: 10.15673/atbp.v15i3.2621

Abstract. The ever-increasing amount of online content and the opportunity for everyone to express their opinions online leads to frequent encounters with social problems: bullying, insults, and hate speech. Some online portals are taking steps to stop this, such as no longer allowing user-generated comments to be made anonymously, removing the possibility to comment under the articles, and some portals employ moderators who identify and eliminate hate speech. However, given the large number of people, an appropriately large number of people are required to do this work. The rapid development of artificial intelligence in the language technology area may be the solution to this problem. Automated hate speech detection would allow to manage the ever-increasing amount of online content, therefore we report hate speech classification for Lithuanian language by application of deep learning.

Keywords: Deep Learning, Transformers, Hate Speech, Text Classification

Abstract. Online hate speech (HS) has become a threat because of violence based on racial, ethnic, sexual, social, and religious distinctions which occur due to the ever-growing volume of online content and the capacity of every user to voice their opinions through comments or other means of expression. Controlling the spread of HS, especially on social networks, is notoriously challenging. It is particularly troublesome since there is debate over how to monitor the content that contains HS, including whether it should be flagged, removed, or otherwise managed [1]. However, it has been demonstrated that HS offenses can result in hate crimes [2], therefore limiting HS should be a priority. Automatic hate speech recognition tools could facilitate this task. There are many distinct hate targets, therefore different languages offer conditions for comparison, revealing variations in linguistic and communicative acts against these targets, allowing HS detection systems to be used in a number of settings [3].

In this paper we present hate speech detection experiments for Lithuanian, where we fine-tune and compare Multilingual BERT, LitLat BERT and Electra models. Section 2 briefly describes relevant literature, definitions and existing challenges. Section 3 presents data, research methods and experimental setup. Section 4 shows the results of fine-tuning and comparison of selected pre-trained models. Section 5 ends this paper with conclusions.

1. Introduction

Online hate speech (HS) has become a threat because of violence based on racial, ethnic, sexual, social, and religious distinctions which occur due to the ever-growing volume of online content and the capacity of every user to voice their opinions through comments or other means of expression. Controlling the spread of HS, especially on social networks, is notoriously challenging. It is particularly troublesome since there is debate over how to monitor the content that contains HS, including whether it should be flagged, removed, or otherwise managed [1]. However, it has been demonstrated that HS offenses can result in hate crimes [2], therefore limiting HS should be a priority. Automatic hate speech recognition tools could facilitate this task. There are many distinct hate targets, therefore different languages offer conditions for comparison, revealing variations in linguistic and communicative acts against these targets, allowing HS detection systems to be used in a number of settings [3].

In this paper we present hate speech detection experiments for Lithuanian, where we fine-tune and compare Multilingual BERT, LitLat BERT and Electra models. Section 2 briefly describes relevant literature, definitions and existing challenges. Section 3 presents data, research methods and experimental setup. Section 4 shows the results of fine-tuning and comparison of selected pre-trained models. Section 5 ends this paper with conclusions.

2. Literature review on HS detection

As HS is a complex and non-trivial phenomenon, it is difficult to detect. Researchers contribute to the detection of this phenomenon by designing frameworks, annotating corpora, extracting meaningful features, and testing automatic classifiers. Also, a few evaluation tasks for HS detection in different languages resulted in released benchmark corpora to encourage further developments in HS detection as it may aid in confronting escalation of online violence and hatred or the spread of fake news [4]. Online HS is suspected to be an important factor in political and ethnic violence such as the Rohingya crisis in Myanmar [5], [6]. Therefore, media platforms are pressured to timely detection and elimination of HS as well as related phenomena, such as cyber-bullying and offensive content [7]. Researchers contributed to a considerable amount of work on HS detection, e.g. as in [8]–[13]. However, most of it is based on hand-crafted features, user information and/or a variety of
metadata which is usually platform-specific [5] and that limits HS detection generalization in terms of new data sets as well as new data sources.

To contribute to HS detection, a number of shared tasks have been organized [14]–[17]. Each of them concentrated on different aspects of HS. For example, [9] introduced a typology on the abusive nature of HS, distinguishing it into generalized, explicit, and implicit abuse. Meanwhile, [17] studied hateful and aggressive messages which targeted women and immigrants. Furthermore, [14] explored the identification of targeted and untargeted insults, therefore proposing the classification of HS into hateful, offensive and profane. Moreover, [15] examined aggression and misogynistic content identification in terms of trolling and cyberbullying. Finally, [18] concluded that most of the shared tasks cover individual-directed abuse, identity-directed abuse, and concept-directed abuse.

As for language coverage in HS detection, English is by far the most investigated as initial work in this area started with English [19]. However, with the advances in natural language processing (NLP) and deep learning, non-English HS detection solutions steadily increase. Therefore, there exist HS detection models for Arabic (e.g., [20]–[24]), Turkish (e.g., [24], [25]), Greek (e.g., [24], [26], [27], Danish (e.g., [24], [28]), Hindi (e.g., [29]–[31]), German (e.g., [15], [32]), Malayalam (e.g., [33], [34]), Tamil (e.g., [34], [35]), Chinese (e.g., [36]–[38]), Italian (e.g., [39]), Urdu (e.g., [40]–[42]), Bengali (e.g., [43]–[45]), Korean (e.g., [46]), French (e.g., [47]–[49]), Indonesian and Portugese (e.g., [50]), Spanish (e.g., [51]), Polish (e.g., [52]) and some others as well, not mentioned due to limits of scope of this article.

Regarding challenges in HS detection, one of them is dataset accessibility and availability. A common problem in this is that datasets, available publicly, become unavailable after some time due to a variety of reasons. To this problem data degradation issue comes hand in hand, i.e., when a dataset, published in some encrypted format, needs to be regenerated on-demand, and after such generation, this dataset does not produce the same volume of data as reported on the publication [53]. This can happen when, e.g., Twitter data is released in the form of tweetIDs and the user deletes the original account or tweets.

Another challenge is class imbalance issue as the hate class in most cases makes less than 12% for the multi-class datasets and for the binary datasets – much less than the preferable half of the total dataset [53]. This dataset structure is one of the causes of lesser accuracy in HS detection. Also, a variety of HS definitions may raise a challenge. HS belongs to a set of related concepts, such as abusiveness, aggressiveness, racism, etc. See the detailed representation of these related concepts in Fig. 1.

![Fig. 1 – Relations between HS and related concepts][4]

However, even in the variety of HS definitions, there were consistencies among them. For example, [54] have analysed available definitions of HS and identified the following similarities:

- HS has a target;
- HS induces violence or hate;
- HS attacks or demeans;
- HS can have some types of humour, such as sarcasm.

A variety of definitions assumes that there are differences in perception of HS. Therefore, existing datasets are affected by these varied definitions because they are annotated based on them, and similar instances can be assigned to different annotation categories based on these differences. For example, [55] investigated the effects of definition on annotation reliability. They concluded that HS requires a stronger, more uniform definition. Similarly, [54] found that most of the publicly available datasets are incompatible due to different definitions attributed to similar concepts.

Also, HS datasets occasionally have very similar labels and some studies merge some of them together into one class (usually the purpose is to reduce class imbalance) [53]. However, this practice could have a negative impact on research as the distinction between classes is necessary. For example, this happened for [12] dataset with offensive and hate classes and [56] – with racist and sexist classes. Classes in the former dataset were merged in [7] and [57] where they merged hate and offensive classes into one class. Meanwhile, [58] merged the offensive and neither class into a non-hate class. Similarly, [59] and [60] merged classes in the dataset introduced in [56]. In HS studies abusive language or toxic comments can surround several paradigms [53], therefore it was proposed to use the terms strictly following the available definitions. Similarly, it was suggested that offensive language is not the same as HS and should not be merged [12].

Finally, methods for detecting HS and related abusive behaviour have become popular and getting better in terms of...
performance and generalization [61], [62]. However, current state-of-the-art solutions still have their limitations in accuracy and therefore their practical real-time applications are restricted [63]. HS detection is still an extremely difficult task, especially when the expression of hate is implicit [64].

3. Data, methods and experimental setup

3.1. Annotated Lithuanian Hate Speech Corpus

About 60,000 user-generated comments from various news portals (15min.lt, alkas.lt, delfi.lt) were collected to create a solution for recognizing HS in Lithuanian. They were also supplemented with 226,776 user-generated comments from the news portal lrytas.lt and thousands of manually collected HS comments from various social media pages and news portals. The latter comments have been collected by a focused search. A total of 25,219 comments were annotated by four annotators. The annotation scheme consisted of 3 classes (tags): neutral language, offensive language and hate speech.

The data in the collected dataset was not evenly distributed. Most user-generated comments were non-hate (neutral), which accounts for 60.7 percent of the total dataset. The lesser part is the category offensive (offensive language), which is 31 percent, and the smallest part is the category hate (hate speech), which accounts for only 8.26 percent of all annotated data (see Fig. 2).

![Fig. 2 – Distribution of classes in annotated text](http://www.atbp.onu.edu.ua/)

There were exceptional cases when marking comments with manifestations of HS, for example, the content of the comment was racist, insulting and inciting hatred, but there were no clearly expressed words, there was a lack of context, for example, the comment: "on fire them!", taken from a topic about women of another nationality, but without context, it can simply be assigned to a neutral class. Another exceptional case was where the comment has a racist, hateful meaning, but is expressed in a figurative sense. To determine the class of such a user-generated comment requires a background of social knowledge. All four annotators decided together what category to assign this type of comment to.

There were also comments which did not correspond to any of the three annotated categories, for example, when the content of the comment consisted only of an internet link, name or nickname, various symbols and emoticons. Such comments were skipped by assigning them the skip category. This category also included user-generated comments which the annotators could not agree on their decisions.

3.2. HS detection methodology

For HS detection we used three popular deep learning models: Multilingual BERT, LitLat BERT and Electra. These artificial intelligence models were trained to work with Lithuanian language data. All three models were further trained to classify Lithuanian user-generated comments, i.e., detect those comments that may contain HS. The selected models have been briefly introduced in the following subsections.

3.3.1. BERT architecture models

Transformer BERT (Bidirectional Encoder Representations from Transformers) is a deep learning model based on the attention mechanism [65], which is usually applied to solve various language technology problems [66]. This model works on the principles of transfer learning [67]. A neural network is trained to generate word embeddings, which are then used as input functions for models that solve mainstream language technology tasks. One of the most significant advantages of the BERT architecture models over other neural network models is understanding the context between words in the text. The model learns the context using the attention mechanism characteristic of transformer models, which consists of encoding and decoding mechanisms [65]. In our case, two BERT models were used in developing the hate speech detection solution: Multilingual BERT and LitLat BERT.

Multilingual BERT model uses the architecture of the BERT model and is trained by the Google team using 104 different languages, including Lithuanian [68]. The model was trained with various texts from Wikipedia. These texts were not annotated, or otherwise processed [69] for the training process.

LitLat BERT is a trilingual model which was built using the XLM-RoBERTa-base (Robustly optimized multilingual model architecture [70]) and trained on Lithuanian, Latvian, and English data. According to the scientific literature, LitLat performs better than Multilingual BERT [71], since in this case, BERT focuses on only three languages.

3.3.2. Electra transformer

Electra is a transformer model that uses a pre-training method that trains two neural network models: a generator and a discriminator. The purpose of the generator is to replace lexemes in a sequence, so it is trained as a masked language
model. Meanwhile, the discriminator tries to determine which lexemes have been replaced by the generator \cite{72} (see Fig. 3). The generator can be any language model that produces an output that is a distribution of lexemes. However, the most common choice is a Masked Learning model trained along with a discriminator. After initial training, the generator is discarded, and only the discriminator (Electra model) is refined in subsequent tasks \cite{73}. This proposed training method is significantly more efficient than the masked training method used in BERT models. This is why the Electra model requires less data and computer resources for training.

**Fig. 3 – An example of detecting a modified lexeme when using Electra model** \cite{73}

The main disadvantage of this model, compared to the previously reviewed models Multilingual and LitLat BERT, is that there has been no pre-trained Electra model for the Lithuanian language. This means that we needed to use as many collected Lithuanian texts as possible in order to pre-train the model ourselves. Nevertheless, for training this model from scratch we needed fewer computer resources than we would have needed for training BERT models.

3.3.3. Fine-tuning the models for the classification task

All three embedding models were additionally trained (fine-tuned) on an annotated dataset of Lithuanian user-generated comments. This dataset consisted of 25 219 comments, annotated into the three classes mentioned earlier:
1. **Hate speech** (2 082 comments);
2. **Neutral language** (15 316 comments);
3. **Offensive language** (7 821 comments).

The datasets were divided into training, validation, and testing sets by the ratio 0.6:0.2:0.2. Comments for the hateful and abusive language classes were replicated (duplicated) in each of the subsets to compare the number of user-generated comments in each class. Since the generated embeddings were vectors of length 512 or 128 symbols, any comments longer than 512 characters were discarded. The BERT models were trained for ten epochs and the Electra model – for 30 epochs. The learning rate of the models was 1e-3, and the Adam optimizer \cite{74} was used to optimize the network weights.

3.4. Implementation of HS detection methodology

The HS detection methodology was implemented by creating a prototype that detects HS in user-generated comments in Lithuanian news and social media. The model we selected for developing a prototype was the LitLat transformer, which classifies the comments into three classes: hate, neutral or offensive language. Upon detecting user-generated comments marked with hateful or offensive language, a model built into a particular system will be able to alert the administrators and let them know that the corresponding user-generated comment is inappropriate. Then the system administrators will have the opportunity to remove the comment or to deal with it differently in order to monitor that such user-generated comments would not exist in the system.

The workflow of the prototype is presented in Fig. 4. In the first stage of the prototype, data extraction takes place, during which user-generated comments are collected from various Lithuanian news portals and social networking sites. Then, the collected comment texts are processed in the prototype’s second stage. At this stage, the text is processed in the exact same way as when preparing user-generated comments for model training. The third stage is the classification of processed comments. Here, the model integrated into the prototype classifies user-generated comments into three classes described earlier. Finally, in the fourth step, the results generated by the prototype are obtained. The result is a list of classified comments, where one can see user-generated comments that may contain HS.

**Fig. 4 – Prototype workflow for detecting HS in Lithuanian in user-generated comments published in news portals and social media**

Since the prototype is adapted for integration into news portals and social networking sites, no graphical user interface was developed. The integrated prototype is expected to work at the code level on the servers. The results of the prototype are meant to be reviewed by the admins of these sites and further dealt with according to their needs.

4. Results

During the experiments, the models were trained to classify annotated texts. The performance of these models was evaluated using accuracy, precision, recall, and F1-score metrics. Each classification model was trained three times with
different random number generation parameters (Seed). Once the models have been trained, the aim was to select the model with the highest accuracy estimate to be used for the next stage of testing. This method of selecting trained models avoids randomly obtaining sub-optimal initial network weights, which can severely compromise the classification results during training [75].

First, the results of the three LitLat BERT test runs are reviewed. For training this model, 10 epochs were used, which was sufficient to train the models for the classification task. In the graph below (see Fig. 5) we can see that the best performance of the LitLat BERT model was achieved in the third trial, where F1-score at epochs 2 and 3 reached almost 71%. However, the completeness result shows the opposite. In comparison to the other curves, the completeness curve is the lowest in the third trial, which means a worse estimate, but still an estimate higher than 0.7. Since F1-score combines precision and recall, the third test run of LitLat base_986 was chosen for further testing based on the result of this metric.

Fig. 5 – Classification accuracy (top left), precision (top right), recall (bottom left) and F1 score (bottom right) values for all LitLat BERT model tests

Meanwhile, when training the Multilingual BERT model, the best F1-score is seen in the first test run, which reached almost 0.66 at epochs 5-6 (see Fig. 6). The first test run resulted in a higher score in either accuracy or recall. As far as the precision is concerned, the best result was achieved in the second test run, which reached an estimate of 0.69. However, considering the overall results, the first test run, Multilingual_base_7458, was chosen for further testing.

Fig. 6 – Classification accuracy (top left), precision (top right), recall (bottom left) and F1 score (bottom right) values for all Multilingual BERT model tests

The Electra model showed a slightly different trend in the curves (see Fig. 7). BERT models obtain high metric estimates from the very beginning, i.e., at epoch 1, in some cases even the best model estimate is reached at epoch 1 or 2 (for example, in the second test run LitLat BERT reached a peak of precision at epoch 1 and then the result only got
worse). In contrast, the Electra model started from the lowest estimate and with each epoch the result improved rapidly up to a certain threshold. For this reason, the Electra model needed more epochs for training (30 epochs were chosen) to reach a similar level as the BERT models. However, it is important to mention that the training time of the Electra model was shorter, so the increased number of epochs did not affect the time.

From the line graphs in Fig. 7 it is observed that the Electra model performed best in the first test. The first test achieved the highest scores in both accuracy, precision, recall and F1-score. However, we can see that neither accuracy nor F1-score curves for the first test reached an estimate of 0.54, which means that this model performed worse as compared to the BERT models.

Fig. 7 – Classification accuracy (top left), precision (top right), recall (bottom left) and F1 score (bottom right) values for all Electra model tests.

Summary of all the model tests is presented in see Fig. 8. After plotting the first 10 epochs, we can see that the Electra model is still well behind the two BERT models in all test estimates. The peak of the Electra model estimates occurred between epochs 10 and 20.

Fig. 8 – Classification accuracy (top left), precision (top right), recall (bottom left) and F1 score (bottom right) values for all models across all tests.

Accuracy for all three models is shown in Fig. 9. We can see that the best performing model was the LitLat BERT, and the worst one was the Electra transformer. The reason why Electra model’s accuracy is so low is that it was trained on only 70 million Lithuanian words. In comparison, LitLat BERT model was trained on 1.21 billion Lithuanian words. So, even though the structure of the Electra transformer allows for the model to be trained with smaller amounts of data,
the amount of data available was still insufficient to train the model accurately.

Since accuracy testing of the models showed that the best HS detection model is LitLat BERT, we chose to use this model for the prototype of HS detection in user-generated comments published in Lithuanian news portals and social media.

5. Conclusions

1. An annotated corpus consisting of 25,219 user-generated comments has been developed for training and testing the hate speech recognition model. In this dataset, 8.26% of the comments contain hate speech, 31% of the comments contain offensive speech and 60.7% of the comments contain neutral speech.

2. Based on precision, recall, accuracy and F1-score, we found that the LitLat BERT model performed the best, with an accuracy of 72%, which is why it was chosen for prototype implementation.

6. References


References


