Project Technical Report

Jayveersinh Raj
B20-DS-01
Innopolis University
innopolis, Russia
j.raj@innopolis.university

Makar Shevchenko
B20-DS-01
Innopolis University
innopolis, Russia
m.shevchenko@innopolis.university

Nikolay Pavlenko
B20-DS-01
Innopolis University
innopolis, Russia
n.pavlenko@innopolis.university

I. Introduction

This work explored in practice the use of the zero-shot transfer technique for the task of determining the toxicity of text in the cross-lingual domain.

A. Topic Description

The primary goal of our project was to accomplish **zero-shot transfer** - the ability of an NLP model to accomplish a specific task in a language it was never directly trained on, performing with a reasonable effectiveness without augmentations of the training dataset or additional fine-tuning.

The narrow NLP-relevant topic chosen in our project was a specific type of sentiment and toxicity analysis - **abuse reporting**. The developed model was trained on toxic comments by Jigsaw Google dataset, which contains more than 150,000 English comments, and is meant to detect, classify and flag the presence of toxicity in a specific comment. As mentioned prior, the goal of the project was to achieve multilingual performance for the model, so despite training only on the English dataset, we have set out with an ambitious goal to make the model work for any number of languages, specifically for low-resource ones.

B. Work Methods

1) Data Preparation: In selection of the training dataset we were guided by certain conditions that were determined by the topic at hand:

1) It must contain text data of messages that are to be classified.
2) It must contain labeled data about toxicity for each message.
3) It has to be large enough for the model to be able to train effectively.
4) The dataset has to be diverse enough for the developed model to generalize better.
5) The dataset should ideally be high-quality, devoid of noise, missing data and having many enough samples for each class.

Search was conducted among English datasets, since it is both the language understandable to all members of the team, and multiple datasets satisfying the outlined conditions could be readily found.

During the intermediate evaluation of our project by the teacher assistant we have been adviced to augment the training dataset with Chat-GPT-generated sentences. However, we have not done so, as we have partially used the Chat-GPT in the generation of the testing dataset, and the toxicity of generated toxic statements was rather dubious. Considering that the orginal training dataset already had good examples of toxicity, adding Chat-GPT-generated sentences would reduce the effectiveness of the training process, as our model already understands toxicity better than the GPT.

To test out the zero-shot transfer of our model, we needed a testing dataset. We have manually prepared and annotated 100 sentences from 50 different languages with a 50/50 toxic/non-toxic split for that purpose - it allowed us to evaluate the performance of the model on a wide selection of languages coming from different families.

2) Model Selection: In order to achieve zero shot transfer, the embedding models selected for our pipeline had to be using alignment of the vector space common to all languages as their core idea. Taking that into account, we have implemented and tested several embedding-classifier pipelines: DistilBert embedder with a classifier, and XLM-RoBERTa embedder and classifier.

3) Model Evaluation: In evaluating our model during training the following metrics were used: F1 for XLM-RoBERTa and accuracy for models with DistilBert.

C. Existing Solutions

While working on the project, we have found multiple papers on multilingual toxicity analysis, and were able to find models that we used in our own implementation [1] [2] [3] [9]. However, we were not able to find a model that incorporated zero-shot transfer and was applied to the task of comment toxicity analysis.

D. Motivation

Traditionally, NLP models have been trained on large amounts of language-specific data to perform a particular task, such as sentiment analysis, text classification, or named entity recognition. However, this approach can be time-consuming and resource-intensive, especially when dealing with multiple languages.

One of the aspects of cross-lingual models that interested us the most in the topic is zero-shot transfer. It provides a way for a multilingual model to take the knowledge it learnt in high-resource languages like English, and to apply it on a low-resource language, such as Nepali, achieving much better accuracy than models that were tailored specifically for the latter language and had to contend with significantly smaller datasets.

E. Applications

Our project/model can be used by any platform or software engineer/enthusiast who has to deal with multiple languages to directly flag the toxic behaviour, or identify a valid report by users for a toxic behaviour. The use case for this can be application specific, but the idea is to make the model work with arbitary language by training on a singular language data available.

F. Paper Structure

1) Data Collection and Processing
    a) Data Source
    b) Data Exploration and Visualization
    c) Data Preprocessing
2) Review of Methods and Models
    a) Sentiment Analysis Methods
        i) Existing
        ii) Ours
    b) Embedding Models
        i) Existing
        ii) Ours
    c) Classification
        i) Existing
        ii) Ours
3) Architecture and Implementation
    a) Description of the System Architecture
    b) Implementation Details and Tools
    c) Training and Testing Methodology
    d) Description of the Codebase and Reproducibility
4) Evaluation and Conclusion
    a) Evaluation Metrics
    b) Qualitative Evaluation
    c) Discussion of Results and Limitations
    d) Conclusion and Future Work
5) Team Member Contribution
6) GitHub Link

II. Data Collection and Processing

A. Data Source

1) Training Dataset: Following the guidelines we have outlined prior in **Work Methods** section, we have been able to find several promising training datasets:

1) Personal attacks dataset from Wikipedia's Detox project [4],
2) Jigsaw and Google toxic comment analysis dataset [5],
3) IMDB ratings sentiment dataset [6].

Among them Jigsaw and Google toxic comment analysis dataset was chosen as our training dataset. It contains a large number of Wikipedia comments which have been labeled by human raters for toxic behavior, containing over 150000 English comments.

2) Testing Datasets: As described prior, one of the testing datasets was assembled and labeled by hand. It contains 50 toxic and 50 non-toxic sentences from different languages belonging to very diverse linguistic groups. Some examples from the dataset are illustrated on the next figure.

Another dataset was generated with the help of GPT4,

	Language	Sentences	Labels		Languages
2	English	The weather today is so beautiful.	Non-Toxic		English
3	English	You're so stupid, you can't even understand the weather forecast.	Toxic		French
4	French	Je suis heureux de te voir aujourd'hui.	Non-Toxic		Spanish
5	French	Tu es tellement ennuyeux, je regrette d'être venu ici.	Toxic		German
6	Spanish	Me encanta la música clásica.	Non-Toxic		Italian
7	Spanish	Eres tan malo en la música, deberías dejarlo para siempre.	Toxic		Portuguese
8	German	Das Essen hier ist wirklich lecker.	Non-Toxic		Russian
9	German	Du bist so dumm, du weißt nicht einmal, was gutes Essen ist.	Toxic		Chinese
10	Italian	Questo posto è così bello, mi piace molto.	Non-Toxic		Arabic
11	Italian	Non capisco perché sei così entusiasta di questo posto, è orribile.	Toxic		Japanese
12	Portuguese	Eu adoro viajar e conhecer novos lugares.	Non-Toxic		Korean
13	Portuguese	Você é tão idiota, nunca foi a lugar nenhum interessante.	Toxic		Turkish

but used the same guidelines as the dataset assembled by hand, though it did not include toxic statements, as ChatGPT has filters installed against toxic responses. 100 statements were generated in total. Some samples from the dataset are illustrated on the next figure.

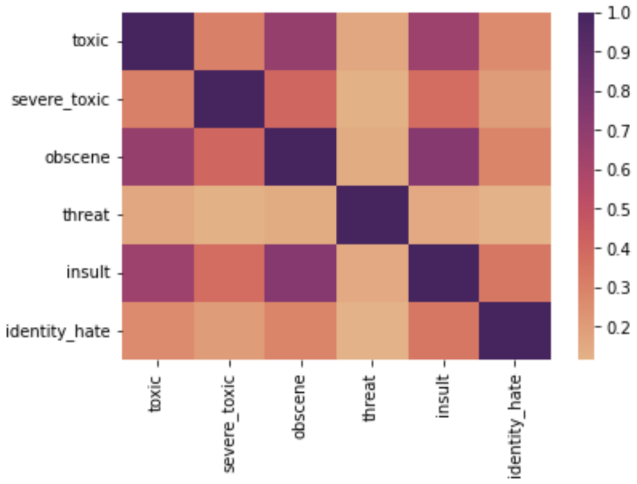
```
1    Me encanta pasar tiempo con mi familia.
2    Je suis ravi de vous rencontrer.
3    Gestern hatten wir einen schönen Tag.
4    Mi piace molto viaggiare.
5    この本はとても面白いです。
```

B. Data Exploration and Visualization

Initially training dataset chosen for our model development contained the following features: **id**: object, **comment_text**: object, **toxic**: int64, **severe_toxic**: int64, **obscene**: int64, **threat**: int64, **insult**: int64, **identity_hate**: int64.

comment_text column is the one we have used to develop the toxicity classification, while columns from **toxic** to **identity_hate** are binary non-exclusive labels describing various toxicity levels present in the comments.

In order to explore our dataset, we have gathered interesting statistical characteristics of all the labels in it, and plotted graphs and a heatmap that highlight correlations between different features:
In order to make sure that the dataset is of a high quality, we have checked if it contained any null values - it did not.



C. Data Preprocessing

During data preprocessing a few important changes were introduced to the dataset:

1) **id** column was dropped, as it was irrelevant in the task our model is set to achieve.
2) Due to extremely sparse nature of the binary labels in the dataset, we have decided to merge them into a single true/false column, that is supposed to represent any level of toxicity found in a comment. It is a temporary measure, and the initial dataset with multiple toxicity labels was preserved for future work.
3) Since various features were found to be heavily imbalanced during data exploration, different solutions were applied to different pipelines: for DistilBERT data was resampled to allow for the same number of toxic and non-toxic comments in the dataset, while for XLM-RoBERTa it was reweighted.

III. Review of Methods and Models

A. Sentiment Analysis Methods

1) Existing: There is a multitude of different sentiment analysis methods in NLP:

1) **Rule-based methods**: Rules are created by hand to identify sentiment in text. For example, certain rules can assign certain scores to words encountered in the text, summing over their values to determine the overall sentiment.
2) **Lexicon-based methods**: These methods use predefined lists of words with associated sentiment scores to analyze text. For example, a lexicon-based approach might assign a sentiment score to each word in a piece of text and then calculate an overall score based on the sum of the individual scores.
3) **ML-based methods**: Usage of ML algorithms to learn patterns between text and sentiment labels.

2) Ours: In our project with have used ML-based methods, using neural networks as toxicity classifiers due to their outstanding performance in recent time.

B. Embedding Models

1) Existing: There are multiple embedding algorithms in NLP, designed to map words from textual to vector representation:

1) **Word2Vec**: This model, developed by Google in 2013, is a neural network-based approach that uses a shallow neural network to predict a target word based on its context. The resulting embeddings are trained to capture semantic and syntactic relationships between words.
2) **GloVe**: The Global Vectors for Word Representation (GloVe) model is a count-based method that uses co-occurrence statistics to generate word embeddings. The model is trained on a global corpus of text and captures both local and global semantic relationships.
3) **FastText**: This model, developed by Facebook in 2016, is an extension of Word2Vec that also considers subword information. This allows the model to generate embeddings for out-of-vocabulary words and capture morphological information.
4) **BERT**: The Bidirectional Encoder Representations from Transformers (BERT) model, developed by Google in 2018, is a transformer-based model that uses a masked language modeling task to generate contextualized word embeddings. BERT is pre-trained on a large corpus of text and can be fine-tuned for various downstream NLP tasks.
5) **XLM-RoBERTa**: XLM-RoBERTa generates contextualized word embeddings - embedding vector for each word depends on the context in which the word appears. This is in contrast to traditional static word embeddings, such as Word2Vec or GloVe, which generate a fixed vector representation for each word.
6) **ELMo**: The Embeddings from Language Models (ELMo) model, developed by Allen Institute for Artificial Intelligence in 2018, is also a contextualized word embedding model. It uses a deep bi-directional language model to generate embeddings that capture both syntactic and semantic information at different layers of the network.

2) Ours: In our project we have decided to use multilingual DistilBERT and XLM-RoBERTa encoding models, as their vector embedding space is the same for all languages.

The reason for choosing DistilBERT was also the inference speed of the model on one Google Colab GPU: about 50 texts per second, while the parent BERT model executed about 0.3 texts per second under the same conditions. Meanwhile, DistilBERT is 40% smaller than the original BERT-base model, is 60% faster than it, and retains 97% of its functionality [7].

We chose the XLM-RoBERTa as the second model as this model give state-of-the-art results in NLP tasks. According to the study by Conneau et al. (2020) [8], a new multilingual language model called XLM-RoBERTa has been developed and outperformed mBERT on various cross-lingual benchmarks, showing a significant improvement of +13.8% in average accuracy on XNLI, +12.3% in average F1 score on MLQA,

and +2.1% in average F1 score on NER. XLM-RoBERTa performed particularly well on low-resource languages such as Swahili and Urdu, with a noticeable improvement of 11.8% and 9.2% in XNLI accuracy, respectively, compared to the previous XLM model.

C. Classification

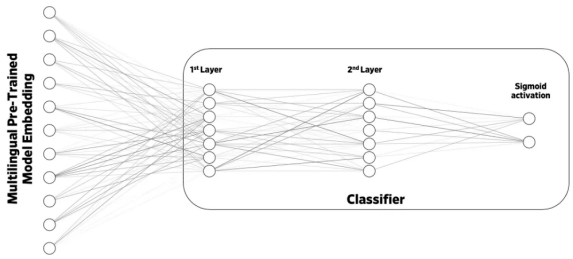
1) Existing: There are a lot of different existing classification algorithms that are used in sentiment analysis tasks. To list a few:

1) **Logistic Regression**: Logistic regression is a simple and efficient classification algorithm that can be used for sentiment analysis. It is well-suited for text classification tasks when combined with vector embeddings of texts.
2) **Support Vector Machines (SVMs)**: SVMs are another popular algorithm for text classification tasks, including sentiment analysis. They work well with high-dimensional vector embeddings and can handle non-linear decision boundaries.
3) **Random Forest**: Random forest is an ensemble-based learning algorithm that can be used for text classification tasks. It works well with high-dimensional vector embeddings and can handle noisy data.
4) **Naive Bayes**: Naive Bayes is a probabilistic classification algorithm that is commonly used in text classification tasks. It is simple and fast, and can work well with high-dimensional vector embeddings.
5) **Neural Networks**: Neural networks, such as feedforward neural networks, convolutional neural networks, and recurrent neural networks, are also commonly used for sentiment analysis with vector embeddings. They can capture complex relationships between words and their contexts, and can achieve high accuracy on sentiment analysis tasks.

2) Ours: In our project we have used the different classifiers. For model with DistilBERT we used the following classifiers: decision tree, Gaussian Naive Bayes, and neural network. The reason why these classifiers were chosen is not high train and test computational complexity, i.e. less than $O(n^2)$, where n is the number of training data points. For model with XLM-RoBERTa we used XLM-RoBERTa classifier as it achieved state-of-the-art performance in a multitude of NLP tasks, including sentiment analysis.

IV. Architecture and Implementation

All our models follow the same structure. They are composed of embedding and classification model. Embedding model transforms input text into vector representation - embedding. Then classification model accepts this embedding to output a decision whether the original text is toxic or not. An exemplary illustration of the described architecture is shown in the figure below.



A. Description of the System Architecture

We developed models with the following embedding-classifier pairs:

- DistilBERT as an embedder and Decision Tree,
- DistilBERT as an embedder and Gaussian Naive Bayes,
- DistilBERT as an embedder and Neural Network,
- XLM-RoBERTa embedder and corresponding XLM-RoBERTa classifier.

For the models with DistilBERT we used multiple versions of embeddings. Each of the versions included downsampling as DistilBERT produce too many features: 768 for each of 128 tokens. Downsampling versions were the following: first, downsampling using feature-wise summing vectors for all the tokens, and second, downsampling using taking feature-wise maximum and minimum values for all the tokens. Also, we experimented with maximum depth of the Decision Tree and number of layers and neurons in them in the Neural Network.

Both pipelines interact with the given dataset, embed the textual column, evaluate it and produce the binary toxic/non-toxic label.

B. Implementation Details and Tools

1) XLM-RoBERTa: XLM-RoBERTa tokenizer was first used. Then these tokens were passed to the XLM-RoBERTa classifier which has a XLM-RoBERTa embedder which aligns the vector space of languages, creates the embeddings, and passes it to the classifier.
   Tools used:
   a) Pytorch
   b) HuggingFace Transformers
   c) XLM-RoBERTa tokenizer, embedder, and classifier
   d) Python
   e) Jupyter notebook
   f) Nvidia 12 GB GPU
2) DistilBERT: for DistilBERT model we utilized distilbert-base-multilingual-cased implementation from transformers Python library. After accepting the input text the model tokenized it and then transormed each of the tokens into embeddings. Then resulting embeddings were downsampled using one of the previously described techniques. Then downsampled embeddings were passed to a classifier. Classifier output the toxicity label for the text. The model was implemented in Python 3.10 programming language.

C. Training and Testing Methodology

1) The model with XLM-RoBERTa has the following training and testing protocols:
   a) Training: Since the dataset is huge, the training is a simple 3 epoch training loop in pytorch. It took approximately 5-6 hours of continous training on the Nvidia 12 GB GPU. F1 metric is to be used
   b) Testing:
      i) **On manually generated data**: **Top-1 accuracy** is to be used. In multilingual models, where the model is trained on multiple languages, it is important to evaluate the performance of the model across all the languages it supports. Top-1 accuracy is a language-agnostic metric that can be used to evaluate the model's performance across all languages. It measures the proportion of cases where the model's top predicted class matches the true label. As described by Zhang, X. et al. [10]

      ii) **On gpt4 generated non-toxic comments**: **Top-1 accuracy** since chatgpt does not generate toxic sentences we could only test it on non toxic sentences.

2) The model with DistilBERT has the following training and testing protocols:
   a) Training was done on the downsampled version of the dataset. We did downsampling to balance the classes. We reduced the number of non-toxic samples to match the number of toxic samples and finally both of the classes had 16 thousands samples.
   The train/evaluation split was 90%/10%.
   We trained only classifiers for the model with DistilBERT while the embedding part was freezed. For neural network we ran 10 epochs, while for Decision Tree and Gaussian Naive Bayes the training configuration was inherited from library without changes.
   b) Testing: the testing was done only on manually generated data. The metric for evaluation is accuracy.

D. Description of the Codebase and Reproducibility

1) **Description of the code base**:
   a) notebooks : contains the prototyping, and experimentation in the jupyter notebooks.
   b) figures: contains the figures of evaluations
   c) test_val_datasets : contains the testiong and validating datasets
   d) progress_reports: the intermediate and final project success reports in latex and pdf formats
   e) helper_functions.py : a python file containing some helper functions for plotting and evaluation that is supposed to stay common across prototyping
   f) deployment : contains deployment files, and the best performing pipeline. In our case the XLM-RoBERTa pipeline.

2) **Reproducibility**: All the code used in our work is available in our GitHub repository. The external data is available online. To reproduce our results a researcher may download the notebooks that were used in our work, and follow the instuctions inside. The software is recommended to be ran in Google Colab with GPU profile.

V. Evaluation and Conclusion

A. Evaluation Metrics

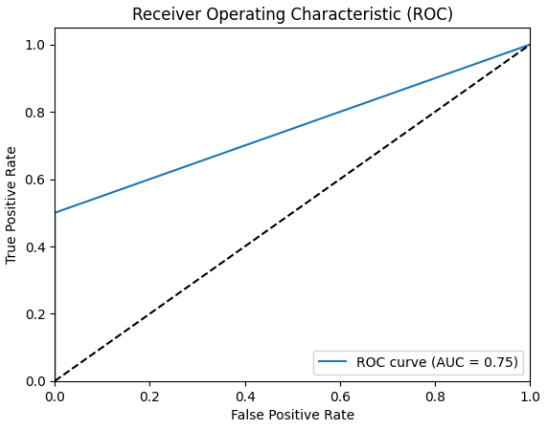
1) **For the English test set** we got the following metrices:
   a) approximately 0.96 F1-Score using XLM-RoBERTa
   b) 54.20% accuracy using DistilBERT + downsampling via sum + Decision Tree with max_depth 10
   c) 55.46% accuracy using DistilBERT + downsampling via sum + Gaussian Naive Bayes
   d) 58.95% accuracy using DistilBERT + downsampling via sum + Neural Network of 3 hidden layers with 768 neurons each
   e) 54.66% accuracy using DistilBERT + downsampling via min and max + Decision Tree with max_depth 10
   f) 54.45% accuracy using DistilBERT + downsampling via min and max + Gaussian Naive Bayes
   g) 57.26% accuracy using DistilBERT + downsampling via min and max + Neural Network of 3 hidden layers with 768 neurons each

2) **For testing samples** we reached 0.75 Top-1 accuracy using XLM-RoBERTa. The models with DistilBERT gave 50% accuracy on the multilingual test set that is similar to random classifier.

- Manually annotated
  Confusion matrix for XLM-RoBERTa

  a) True positive: 25
  b) False positive: 0
  c) True negative: 50
  d) False negative: 25

  AUC-ROC plot



- GPT-4 generated:
  Confusion Matrix
  a) True positive: NOT PRESENT
  b) False positive: NOT PRESENT
  c) True negative: 100
  d) False negative: 0

B. Qualitative Evaluation

The best found pipeline XLM-RoBERTa generalizes well and is suitable for the application that we are targeting. However, the model sometimes struggle against a severe toxic comment on a distant language to English like Gujarati, since Gujarati falls into Family language of Hindi. The patterns are observed that in rare occasions it miss identifies the toxic comment as non toxic.

We reached 0.75 Top-1 accuracy for testing samples because of the nature of the testing set. Toxicity is a subjective term, and out model however only flags severe toxicity, it is expected, we do not want users to be blocked for something that is not severe. In fact, on analysing the non-toxic comments annotated as toxic by manual annotation, is translated by gpt-4 without any issue, which means our model is performing very good.

C. Discussion of Results and Limitations

1) **Discussion**: The study reveals that languages with closer similarity to English, such as German or Dutch, tend to have better zero-shot transfer performance. In contrast, distant languages like Japanese or Korean exhibit relatively lower transferability. However, it is not to be interpret as the model short coming since it still identifies most of severe toxicity of even a distant language. This observation underscores the need for developing models that can better handle the linguistic differences between languages, especially when dealing with low-resource or distant languages.
2) **Results**: The XLM-RoBERTa pipeline shows better results compared to DistilBERT one, and the pipeline is production ready to be deployed for the task of severe toxicity detection over the internet.
3) **Limitations and Challenges**: Despite the promising results, several challenges persist in cross-lingual zero-shot transfer for toxicity analysis. The lack of labeled data for low-resource languages remains a significant hurdle, as it prevents models from being fine-tuned on target languages. Additionally, the models struggle to capture cultural nuances and contextual information that may be critical for accurate toxicity detection. In addition transliteration is to be added to infer on languages written with different scripts. We tried it using open source frameworks, but they are not so effective.

D. Conclusion and Future Work

In conclusion, we have demonstrated the efficacy of a cross-lingual zero-shot transfer pipeline leveraging an effective XLM-RoBERTa model, which represents the cutting-edge of multilingual representation learning. We have learned how with careful experimentation and evaluation we can reveal the promising potential of this innovative methodology to overcome traditional linguistic barriers and deliver strong performance across various languages and tasks.

By combining unsupervised cross-lingual pretraining, masked language modeling, and deep transformer-based architectures, our approach has demonstrated strong generalization capabilities. By exploiting the inherent multilingual semantic structure within XLM-RoBERTa's latent space, we have successfully bypassed the need for resource-intensive parallel data, circumventing the exigencies of task-specific fine-tuning.

Future work in this domain should explore the frontiers of transfer learning, extending our methodology to low-resource and endangered languages to bridge the digital divide and foster linguistic diversity. It would be prudent to investigate the amalgamation of unsupervised and supervised cross-lingual techniques, potentially unraveling novel synergies that could propel performance to new heights. Moreover, addressing the intricacies of domain-specific jargon and tackling linguistic nuances, such as idiomatic expressions and cultural context, are exigent objectives that warrant further exploration.

In addition, the incorporation of advanced model compression techniques, like knowledge distillation and pruning, could engender more efficient models that retain the performance benefits of XLM-RoBERTa while minimizing the computational overhead. Lastly, exploring explainable AI and interpretability techniques can help us understand how the model can transfer knowledge across different languages. This can lead to more reliable, transparent, and ethically aligned multilingual NLP solutions.

VI. Team Member Contribution

- **Jayveersinh Raj**: datasets exploration, development and evaluation of XLM-RoBERTa embedder-classifier pipeline, research into metrics;
- **Makar Shevchenko**: data understanding and preprocessing, development of DistilBERT encoder - BERT classifier pipeline, research into metrics, organizational responsibilities;
- **Nikolay Pavlenko**: progress evaluation, dataset creation, composition of reports.

VII. GitHub Link

Link: https://github.com/SyrexMinus/cross_lingual_nlp

References

[1] Bogoradnikova, D., Makhnytkina, O., Matveev, A., Zakharova, A., & Akulov, A. (2021, May). Multilingual Sentiment Analysis and Toxicity Detection for Text Messages in Russian. In 2021 29th Conference of Open Innovations Association (FRUCT) (pp. 55-64). IEEE.

[2] Leite, J. A., Silva, D. F., Bontcheva, K., & Scarton, C. (2020). Toxic language detection in social media for Brazilian Portuguese: New dataset and multilingual analysis. arXiv preprint arXiv:2010.04543.

[3] Sarvaiya, J. J. (2022). Multilingual Text Analysis using Natural Language Processing and Transfer Learning (Doctoral dissertation, Dublin, National College of Ireland).

[4] https://figshare.com/articles/dataset/Wikipedia_Talk_Labels_Personal_Attacks/4054689

[5] https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data

[6] https://ai.stanford.edu/ amaas/data/sentiment/

[7] https://towardsdatascience.com/everything-you-need-to-know-about-albert-roberta-and-distilbert-11a74334b2da

[8] A. Conneau, K. Khandelwal, N. Goyal et al. (2020) Unsupervised Cross-lingual Representation Learning at Scale. arXiv:1911.02116. https://doi.org/10.48550/arXiv.1911.02116

[9] Göhl, S. A. (2022). Zero-shot cross-lingual transfer learning for sentiment analysis on Swedish chat conversations.

[10] Zhang, X., Yin, J., Zhang, J., & Liu, J. (2021). Multi-lingual text classification with multi-head attention-based transformers. Journal of Information Science, 47(5), 607-625. https://doi.org/10.1177/0165551520969136