Bachelor thesis

Czech
Technical
University
in Prague

Department of Cybernetics

Evaluation of Media News Discussion Posts

Luka Peraica

Supervisor: Ing. Radek Mařík, CSc.
May 2025

Acknowledgements

I would like to wholeheartedly thank my supervisor, Ing. Radek Mařík, CSc., for assigning me this thesis topic and for the opportunity to work under his supervision. I deeply appreciate his guidance, encouragement, and unwavering positivity throughout the entire process. Without his support, this thesis would not have been possible.

Declaration

I declare that this work is all my own work, and I have cited all sources I have used in the bibliography.

Prague, May 15, 2025

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 15. května 2025

Abstract

Online discussions frequently attract disruptive behaviors such as trolling, intended to provoke or mislead users. This bachelor's thesis explores troll detection using Natural Language Processing (NLP), focusing on user comments from the Czech news site Novinky.cz. It employs transformer-based models, specifically multilingual BERT, to assign users a continuous "trolliness" score, rather than only a traditional binary classifications.

Initially trained on multilingual datasets, the model outperformed traditional methods but faced challenges transferring knowledge directly to Czech comments. Further fine-tuning with a small annotated dataset of Czech comments significantly improved its effectiveness.

The thesis concludes by highlighting the method's strengths and acknowledging limitations.

Keywords: Online trolling, Natural Language Processing (NLP), Transformer models, Multiligual BERT, Machine Learning

Supervisor: Ing. Radek Mařík, CSc.

Abstrakt

Internetové diskuse často přitahují rušivé chování, jako je trolling, jehož cílem je provokovat nebo klamat uživatele. Tato bakalářská práce se zabývá detekcí trollů pomocí metod zpracování přirozeného jazyka (NLP), se zaměřením na komentáře uživatelů českého zpravodajského portálu Novinky.cz. Využívá transformer modely, konkrétně vícejazyčný model BERT, k přiřazení uživatelům kontinuálního skóre „trollovitosti“, což lépe vystihuje komplexitu online chování oproti tradičnímu binárnímu rozdělení.

Model původně trénovaný na vícejazyčných datových sadách překonal tradiční metody, ale narazil na problémy při přenosu znalostí přímo na české komentáře. Další doladění pomocí malé anotované sady českých komentářů významně zvýšilo jeho efektivitu.

Práci uzavírá shrnutí silných stránek metody a uznání jejích omezení.

Klíčová slova: online trolling, Zpracování přirozeného jazyka (NLP), Transformer modely, vícejazyčný BERT, Strojové učení

Překlad názvu: Hodnocení diskusních příspěvků mediálních zpráv

Contents

Figures

Tables

Chapter 1

Introduction

1.1 Problem Statement

The way humans communicate and interact has changed dramatically in the age of the internet. Social media sites, forums, and comment sections have become primary spaces for people to share ideas, debate issues, and engage in public discourse. These online discussion platforms allow individuals from different backgrounds to express their opinions and be part of conversations easily than ever before. However, while online discussions create opportunities for connecting people and sharing information, they also come with major challenges like the spread of misinformation, the polarization of society, and the spread of large-scale disruptive behavior.

Understanding how these platforms shape opinion is essential in the modern day. In today's flood of diverse media sources and information, even professional media analysts find it challenging to navigate and filter reliable information. A key aspect of democracy is the ability to express opinions and refine perspectives through discussions. Today, most of such discussion happens online on social media platforms like Twitter, Facebook, and Reddit, which have gained a powerful influence on public opinion, shaping political outcomes [BS12]. The importance and ubiquity of online discussions can also make them targets for individuals whose goal is to disrupt and manipulate conversations for various reasons.

1.2 Defining Online Trolling

To address the negative consequences of disruptive online behavior, it is important to define one of its most prevalent forms: online trolling. Online trolling is a deliberate act intended to provoke, deceive, or disrupt online conversations. According to Coles and West [CW16], trolling involves actions meant to annoy, frustrate, or engage others in pointless disputes. Similarly, Golf-Papez and Veer [GPV17] define trolling as ”deliberate, deceptive, and mischievous attempts to provoke reactions from other users”.

The term ”trolling” itself was originally borrowed from fishing slang, where it referred to dragging a baited line through the water to catch fish. In the online context, the term seems to have first been used in the 1990s on the USET discussion system, where some users would deliberately create posts designed to trigger angry corrections from newbie users who weren't aware of such practices.

Then there is a second dictionary definition for the word ”troll”, which is also quite relevant to the perception of online trolls and perhaps for most people the first connotation that comes to mind. This definition refers to a troll as a large, ugly creature from folklore, often depicted as a brutish ogre. The word ”troll” is derived from the Old Norse word ”troll”, which means ”giant” or ”ogre”. In this context, the term evokes an image of a monstrous being that lurks in the shadows, waiting to pounce on unsuspecting victims. And while the term trolling originated from the early bait posts, related to the fishing term, over time the character and label of the ”troll” developed, which is more closely related to the folklore definition. This shift in meaning reflects the evolution of online trolling from more light-hearted baiting and joking to a label for a malicious character lurking on the internet[DBGJS21].

While some forms of trolling may seem harmless or playful, others can escalate into targeted harassment, misinformation campaigns, and efforts to manipulate public opinion.

People engage in trolling for various reasons, from simply seeking amusement from the activity to pushing political or ideological agendas. Studies indicate that personality traits like psychopathy, narcissism, and Machiavellianism are often linked to trolling behavior [BTP14]. Additionally, research has shown that certain psychological factors also contribute to the online trolling phenomenon, such as the ”online disinhibition effect”. This theory suggests that people act more aggressively online because they feel anonymous and free from real-world consequences [Sul04]. The combination of these factors makes online discussions particularly fertile ground for trolling behavior.

Beyond individual psychology, trolls also exploit the technological factor of the discussions, particularly social media algorithms that focus on engagement above all else. Effectively playing into the algorithm allows them to more easily and effectively spread divisive content and manipulate conversations [GPV17].

1.3 Impacts of Trolling

Trolling negatively affects both honest users and the discussion as a whole. Those targeted by trolls often experience stress, anxiety, and frustration, which can discourage them from engaging in future conversations. Trolling not only harms individual well-being but also degrades the quality of discussions. As user trust is eroded and people become more skeptical of digital interactions, a toxic environment is created where constructive engagement becomes difficult [GPV17].

On a larger scale, trolling can have significant consequences, particularly when used as a tool for political manipulation. State-sponsored troll campaigns have been used to spread propaganda, influence elections, and undermine public trust in media [BH17]. One of the most well-known examples is the Russian *Internet Research Agency* (IRA), which ran large-scale trolling operations during the 2016 U.S. presidential election between Hillary Clinton and Donald Trump. These trolls used fake accounts to post divisive content and manipulate public discourse [LW20]. Similar use of trolling in political campaigns and foreign influence operations has been documented worldwide, demonstrating the severity and importance of addressing the issue.

This thesis aims to identify and analyze the behavior of trolls in online discussions. Specifically, it will explore different NLP techniques for troll detection, including stylometry, topic modeling, deep learning, and transformer models. The goal is to identify harmful contributions and contributors to online discussions and to explore possibilities for further research in this area.

Chapter 2

Natural Language Processing

Given the impact of disruptive trolls on online discourse and society at large, research efforts have focused on developing techniques to better understand, detect, and mitigate their activity. This chapter explores the methods used to analyze and identify trolling behavior, particularly through Natural Language Processing (NLP). It covers key approaches such as stylometry, sentiment analysis, and topic modelling.

2.1 Stylometry

Stylometry is the discipline of analyzing writing style to uncover patterns, identify authors, and extract meaningful details from texts [MW64] [PMMM20]. The term was introduced in 1890 by the Polish philosopher Wincenty Lutosławski, who applied it to analyze Plato's works [Lut98]. In the context of this thesis, stylometry involves the use of automated techniques to analyze linguistic traits that distinguish authors based on their unique writing patterns.

The underlying assumption in stylometry is that an author's choices are influenced by sociological factors, such as age, gender, and education level, as well as psychological factors, like personality and native language proficiency [Dae13]. This assumption can be extended to groups of authors, especially those who may share common objectives or adhere to specific guidelines, such as state-sponsored trolls, or display similar behavioral patterns as seen among ordinary trolls. These individual or collective choices can manifest as identifiable stylistic features within texts, which computational models can analyze to detect trolling behavior. Stylometric analyses typically examine lexical choices like vocabulary richness, syntactic elements including sentence structure and grammatical complexity [SSV18], and semantic dimensions, such as sentiment and thematic consistency [PRKLM18]. Extracting and evaluating these features allows machine learning classifiers to differentiate between regular users and trolls based on their distinctive linguistic signatures.

2.1.1 Stylometry in Literature

An example of stylometry applied to troll detection is presented in the work of Machová *et al.* [MPH21]. The paper examines troll detection in Slovak Facebook discussions on COVID-19 by combining shallow stylometric cues with engagement and affective information. From roughly 2,500 manually labelled comments, they extract length-based metrics (character and word counts, average word length), orthographic signals (capital-letter and digit frequency), and eight sentiment/provocativeness categories, and enrich these with interaction metadata such as the number of "likes". These features were used by classical classifiers-including Support Vector Machines (SVM), Multinomial Naïve Bayes (MNB), and logistic regression trained on bag-of-words and TF-IDF representations.

Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a term-weighting scheme that emphasises words that are frequent in an individual document yet rare in the overall collection. Following Manning, Raghavan, and Schütze's textbook Introduction to Information Retrieval [**?**], the weight of a term t in a document d taken from a corpus of N documents is:

$$\mathrm{tf\text{-}idf}_{t,d} \;=\; \mathrm{tf}_{t,d} \times \log\!\Big(\frac{N}{df_t}\Big),$$

where $\mathrm{tf}_{t,d}$ is the (raw or normalised) term frequency of t in d and df_t is the number of documents in which t appears. This product down-weights ubiquitous stop-words (e.g., "the", "and") while retaining terms that are stylistically distinctive for an author, making TF-IDF a staple feature in authorship-attribution and troll-detection pipelines.

Other Stylometric Approahces in Lietrature

In another paper, an example of stylometry applied to fake news detection is presented in the work of Pérez-Rosas et al. [PRKLM18]. They used a variety of stylometric features, including n-grams, punctuation frequency, readability metrics, and syntactic features. They also incorporated psycholinguistic features extracted from the LIWC lexicon, which categorize words into various psychological categories. LIWC features capture psychological aspects of a text, such as emotional tone or cognitive processes, potentially revealing underlying psychological differences between fake and legitimate news writers. A linear SVM classifier was trained on these features to differentiate between fake and legitimate news articles. Their results showed that stylometric features can be effective for the task, achieving accuracies of up to 76% which outperformed two human annotators. The analysis uncovered distinct linguistic patterns in fake news, such as increased use of social and positive words, a focus on present and future actions, and a higher prevalence of adverbs, verbs, and punctuation marks.

Though stylometry has proven useful for text classification, recent advancements in large language models and their potential for misuse might pose a substantial challenge to its efficacy. A language model (LM) is a probabilistic system that assigns likelihoods to sequences of words and can generate human-like text. As demonstrated by Schuster et al. [SSSB20], stylometry may struggle to differentiate between human-written and machine-generated text. In their study, they find that while a state-of-the-art stylometry-based classifier could effectively detect the presence of machine-generated text within human-written content, it struggled to discern the truthfulness of the generated text. For instance, even a single auto-generated sentence within a longer human-written text was easily detectable, but the veracity of that sentence remained largely undecidable. Additionally, even a relatively weak LM could produce statement inversions that evaded detection by the stylometry-based model.

These findings collectively highlight stylometry's potential for detecting hidden manipulation in online text, although recent advancements in language generation models present new challenges. It is also important to note that to achieve good results, stylometric features weren't used on their own but along with others like metadata (like counts, followers) or sentiment analysis.

2.2 Sentiment Analysis

Sentiment analysis is a method of determining the emotional tone of text, which can be achieved using lexicon-based methods, machine learning, or deep learning approaches. It identifies positive, negative, neutral, or ambivalent tones in text.

2.2.1 Sentiment Analysis in Troll Detection

The use of sentiment analysis has been explored as one of the methods used for troll detection. Jiang et al. [JTS21], for example, explored the use of sentiment analysis for troll detection on the Chinese social media platform Weibo. They employed a Word2Vec model trained on a dataset of Weibo comments to generate word embeddings. These embeddings were then used to calculate sentiment scores, incorporating features such as happiness, anger, disgust, and fear. The sentiment was used along with meta features such as the location of a comment in a thread or its like count to train XGBoost and SVM models for the troll detection task. The approach proved effective with the XGBoost model achieving an accuracy of up to 89% and SVM up to 87%.

In a related study, Machová et al. [MMV22] investigated the detection of suspicious reviewers in online discussions, focusing on trolls. Their lexicon-based approach analyzed the polarity of comments to identify trolls. It was based on the tendency of trolls to express extreme opinions that oppose the general sentiment of the discussion. They compared this approach with a Convolutional Neural Network (CNN) model, finding that both performed similarly on text data, achieving accuracies of 0.95 and 0.959, respectively.

The study also employed machine learning methods, such as Support Vector Machines (SVM), using non-textual features like comment karma, likes, and dislikes. With the SVM model, they achieved an accuracy of 0.986.

In recent years, Transformer models have emerged as the state-of-the-art approach for a wide range of NLP tasks. First introduced by Vaswani et al. in 2017 [VSP+17] in the pioneering paper *Attention Is All You Need*, Transformers offer a new way for machines to understand and generate language. Unlike earlier methods, they are designed to efficiently capture relationships and patterns within text, even over long passages. Their ability to handle large amounts of data and to adapt to complex language structures has made them a key tool in modern NLP applications. Another important reason for the success of Transformer models is that they are typically very large models pre-trained on massive datasets. Many of these datasets are multilingual, meaning that the models learn from several languages at once. As a result, we can often use the same embedding space for different languages, and the knowledge gained in one language can transfer to some extent to others. This makes pre-trained Transformers especially valuable when dealing with multilingual or cross-lingual data.

In this section, we will introduce the basic ideas behind Transformer models.

2.2.2 Self-Attention

The key innovation of Transformer models is the use of self-attention mechanisms, which allows the model to dynamically focus on different parts of the text sequence, regardless of position. Instead of processing text word by word like earlier models, Transformers can consider all words in a sentence at once, deciding how much attention each word should pay to others. This allows the model to capture complex patterns and dependencies across long texts, making it very powerful for understanding both the context and meaning of language.

Each word (token) creates three vectors, which are used by the self-attention mechanism[VSP+17]:

Query (Q) - a vector that represents the active word in the input sequence ("What am I looking for?").

Key (K) - a vector that represents other words from the input sequence, which are compared to the query ("What do I have?").

Value (V) - a vector that represents the information we want to extract from the compared words ("What do I want to know?")

The model uses a dot product to calculate the similarity between the query and the key vectors. The result is then scaled and passed through a softmax function to create a probability distribution. This distribution is then used to weight the value vectors, allowing the model to focus on the most relevant information in the input sequence. The model does all of the above steps in parallel for sets of queries, keys, and values - it works with matrices instead of separate vectors. The equation for the self-attention mechanism as described is then:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

2.2.3 Multi-Head Attention

In addition to the self-attention mechanism, Transformer models also use multi-head attention. This means that instead of having a single set of query, key, and value vectors, the model has multiple sets (or heads) that can learn different aspects of the input data. Each head performs its own self-attention calculation, and the results are then concatenated and linearly transformed to create the final output. This allows the model to capture a wider range of relationships and patterns in the data.

The multi-head attention mechanism is defined as follows[VSP+17]:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^O \tag{2.2}$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.3}$$

The output is finally passed through a feed-forward neural network (FFN), which consists of two linear transformations with a ReLU activation function in between. The FFN is applied to each position separately and identically.

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2 \tag{2.4}$$

These mechanisms allow the Transformer to capture relationships and dependencies between words in a text, regardless of their position. The use of multi-head attention further enhances these abilities by allowing the model to learn different aspects of the text simultaneously. As a result, Transformer models can efficiently process long sequences, recognize contextual nuance, and achieve state-of-the-art performance in various NLP tasks.

2.2.4 BERT

One of the most influential Transformer-based models for natural language understanding is BERT (Bidirectional Encoder Representations from Transformers). Introduced by Devlin et al. in 2018 [DCLT18], BERT builds on the Transformer encoder architecture and is pre-trained on large text corpora using masked language modeling and next sentence prediction tasks. Unlike traditional language models that read text from left to right, BERT is deeply bidirectional, meaning it learns information from both the left and right context of a token simultaneously during pre-training.

An important aspect of the BERT architecture, particularly for tasks requiring a summary representation of an entire input sequence, is the use of a special token called [CLS] (classification token). This token is added at the beginning of the input sequence and is used to aggregate information from all tokens in the sequence. The final hidden state corresponding to the [CLS] token can then be used, for example, as the input for classification tasks.

This bidirectional nature and the [CLS] token convention, among other features, allow BERT to capture richer contextual information about language, making it highly effective for a wide range of downstream tasks such as text classification, question answering, and sentiment analysis, and a promising fit for our troll detection task.

2.2.5 Transformer Models in Troll Detection

The paper MetaTroll[TZL23] proposed a few-shot troll detection framework designed to adapt quickly to new state-sponsored influence campaigns using minimal labeled data. Their approach is based on a meta-learning framework and incorporates campaign-specific transformer adapters to tackle catastrophic forgetting, a common problem where models lose the ability to detect trolls from older campaigns after continual updates. MetaTroll outperformed traditional n-gram SVM baselines, achieving a 92.3% F1-score in 5-shot scenarios and demonstrating strong cross-lingual capabilities.

Embeddings generated by transformer models have been shown to outperform static methods due to their ability to capture the contextual meanings of words. BERT-based word embeddings outperformed static GloVe vectors reaching AUC scores of 0.924. The authors of the paper [YZ23] highlight how transformers can better capture contextual nuances in language, which can be important for the task of troll classification, where language can be manipulative and deliberately deceptive.

In this thesis, we will attempt to leverage pre-trained Transformer models such as BERT and try to fine-tune them for the troll detection task. The reasons for the choice of using Transformers will be outlined in the next chapters.

Chapter 3
Dataset

Before outlining the proposed method, I will first describe the dataset that forms the basis of our work. The properties of the dataset are important as they directly shape the choice of methods and define the limitations and the goals of the work.

3.1 Main Dataset

The dataset used in this thesis consists of user-generated comments collected from the discussion sections under news articles published on Novinky.cz, one of the largest Czech news portals. Each article on Novinky.cz includes a public comment section where users participate in discussions about the content presented. These discussions are often extensive, with some articles attracting hundreds of user comments.

In the Czech online media environment, it is generally recognized that the comment sections on major news sites, particularly on Novinky.cz, for example, frequently serve as hotbeds for controversy and emotionally charged discourse. They are often perceived by the public as spaces where individuals express grievances, frustrations, and polarizing viewpoints, sometimes in ways that border on or cross into what could be described as abusive, manipulative, or troll-like behavior. This cultural context makes Novinky.cz a relevant and interesting setting for exploring how online discussions develop, especially where conversations become heated or emotionally charged.

For the purposes of this thesis, a large-scale dataset comprising approximately 350,000 comments posted by around 48,000 users was provided by Newton Media, a prominent media intelligence organization.

Each data entry includes the following attributes:

- **Comment content** - the full textual body of the comment.
- **Article metadata** - including the title and link to the article under which the comment was posted
- **Timestamp** - the date and time when the comment was published.
- **Author name** - full author name as collected from the discussion.

- **Sentiment label** - a sentiment category assigned by Newton Media, labeled as one of the following: *Neutral*, *Positive*, *Negative*, or *Ambivalent*.

A consideration for the work with this dataset is the nature of the comments. The comments in general seem to be mostly negative in tone and often emotionally charged, as can be seen by the distribution of sentiment.

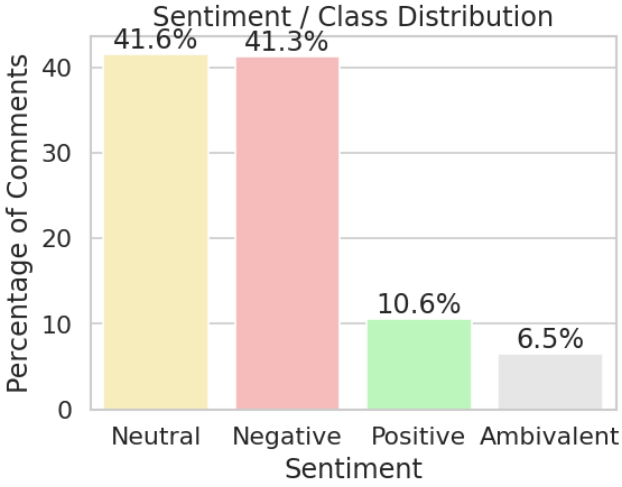


Figure 3.1: Sentiment distribution of collected novinky.cz comments

A key challenge posed by this dataset is the absence of explicit troll/non-troll labels. Since there is no ground-truth annotation for trolling behavior, we cannot directly apply supervised classification methods. Because of this, we have to rely on unsupervised or semi-supervised techniques, such as clustering, topic modeling, or anomaly detection, to try to find patterns that may point to trolling based on how the comments are written and their sentiment. The lack of labeled data also makes it difficult to verify the accuracy or effectiveness of any classifications or patterns identified during experimentation. Without labeled data, we cannot easily measure how accurate our models are and have to instead rely on manual checks and interpretation of the results.

3.2 Additionall Datasets

In addition to the primary dataset collected from Novinky.cz, several publicly available labeled datasets were also used in this thesis. They were collected from different platforms such as Twitter or Reddit, and the majority are in English, although some include other languages as well. While they differ in context and language, they offer labeled examples that we will attempt to use for pre-training of models for the later analysis of Czech online discussions.

3.2.1 IRA Troll Tweets

One of the external datasets used in this thesis is a collection of tweets linked to the Russian *Internet Research Agency* (IRA), which was mentioned in the introductory chapter of the thesis. The dataset was published by FiveThirtyEight in connection with their article *Why We're Sharing 3 Million Russian Troll Tweets*, and was originally collected by researchers from Clemson University. It contains nearly 3 million tweets posted between February 2012 and May 2018 by accounts identified by Twitter as being linked to the IRA, which were provided to the US Congress for investigation into 2016 presidential election interference. In total, the dataset includes 2,973,371 tweets from 2,848 Twitter handles. Most of the tweets are in English, but some are in other languages, including Russian or German. The dataset is publicly available and can be accessed through the FiveThirtyEight GitHub repository.[1]

3.2.2 Information Operations Dataset

Another external resource used in this thesis is a collection of labeled datasets for research on information operations (IOs), introduced by Seckin et al. [SPN$^+$24]. The full collection contains over 13 million posts from approximately 303,000 accounts. The dataset includes both verified IO posts and control data from legitimate accounts, covering 26 distinct manipulation campaigns originating from different countries. The data is organized first by one of 16 identified state actors, such as Russia, China, or even Catalonia, and then further subdivided into distinct operations. The IO posts were identified and released by major social media platforms, including Twitter, Facebook, and Reddit, while the control data captures organic user discussions on similar topics within the same time frames.

3.2.3 Non-troll Datasets

In addition to the troll datasets, several non-troll datasets were also collected to ensure the availability of apolitical and organic user discussions.

The first non-troll dataset is the Civil Comments dataset, obtained from Hugging Face. It consists of public comments posted between 2015 and 2017 on approximately 50 English-language news sites. Each comment is labeled with values for toxicity, obscenity, and other attributes. For the purposes of this thesis, only comments labeled as non-toxic (toxicity score between 0 and 0.1) were used, which represent the majority of the dataset.[2]

Additionally, a small dataset of celebrity tweets was obtained from Kaggle. This dataset consists of posts by well-known public figures providing examples of casual and generally non-political online communication.[3].

Finally, a custom dataset was manually created by scraping tweets from Czech public figures and politicians. A selected list of Twitter accounts was compiled, and 20 tweets were collected from each account.

[1] https://github.com/fivethirtyeight/russian-troll-tweets/

[2] https://huggingface.co/datasets/google/civil_comments

[3] https://www.kaggle.com/datasets/abaghyangor/celebrity-tweets

Chapter 4
Proposed Method

In this chapter, we will outline the proposed method for detecting troll-like behavior in online discussions. The core idea behind the method is the use of transformer-based models, specifically multilingual BERT-based models, in a regression task designed to quantify a user's troll-like behavior. Instead of a binary classification task, the approach is to assign a user with a continuous ”trolliness” score, measured from 0 to 1.

4.1 Motivation

As the backbone of the method, I decided to use multilingual BERT-based models, as they are trained across dozens of languages at once, which makes them a natural choice when trying to transfer knowledge from English or Russian troll datasets to Czech. Beyond their multilingual capabilities, BERT models are also able to capture and represent both syntactic and semantic relationships and dependencies within a text sequence. Instead of manually designing and extracting individual features like syntax counts, stylometric traits, sentiment scores, in theory, BERT should be able to learn and encode much of this information into its embeddings and attention mechanism[RKR20].

A classical machine learning approach using manually selected stylometric and other features is not suitable for this task, due to the limitations of the datasets we are working with, which were mentioned above. However, BERT should be able to capture similar semantic and syntactic knowledge while also being able to be used in our specific task with limited labeled data and a multilingual dataset.

The motivation to use a regression task instead of a binary classification task is twofold. First, the main dataset of Czech comments lacks troll/non-troll labels, so standard supervised classification methods cannot be applied. Second, troll behavior isn't a straightforward binary state, but rather a spectrum of behavior, with users displaying varying degrees and different types of disruptive behavior. For those reasons, we focus on getting a *trolliness score* rather than a troll classification.

4.2 Data Collection and Preprocessing

The first step of the method is the collection and preprocessing of the data. To ensure consistency of the embeddings generated by BERT, I implemented a preprocessing pipeline for handling text. The pipeline includes the following steps:

- **URL and Media Removal** - URLs and images hosted on Twitter are removed.
- **Twitter-specific Artifacts Removal** - Twitter-specific elements, such as hashtags, mentions, are removed in training as they do not appear in the target Czech dataset.
- **Emoji and Special Character Handling** - Emoji and non-standard Unicode characters are removed from the text.
- **Whitespace Normalization** - Consecutive whitespace characters are replaced with a single space.

An intentional choice was made to retain all stop words during preprocessing, as BERT models use context from all words, including stop words. The removal of stop words is common in classical NLP tasks, but in the case of BERT, it is not necessary and can even diminish the performance.

A key design decision in this thesis was to rate the trolliness at the user level, rather than at an individual comment level. This decision was based on the analysis and observations from the labeled troll datasets. A recurring pattern was that many troll accounts not only engaged in disruptive and manipulative behavior all the time. Instead, in many cases, trolls posted mostly ”normal” content, perhaps to blend in with regular users, pushing their agenda more subtly in some posts and then only occasionally posting more overtly troll-like comments.

For this thesis, we will exclude all users with fewer than 5 comments, as our aim is to try to find broader patterns of troll-like behavior not only one-off examples of offensive or provocative comments. We do this both for the initial training as well as when working with the target Czech dataset. While this discards about half of the users in the dataset, it is only a small fraction of the comments, about ten percent. The distribution of authors by number of comments can be seen in Figure 4.3.

4.3 Model Architecture

The model architecture is designed in two levels: the comment level and the user level.

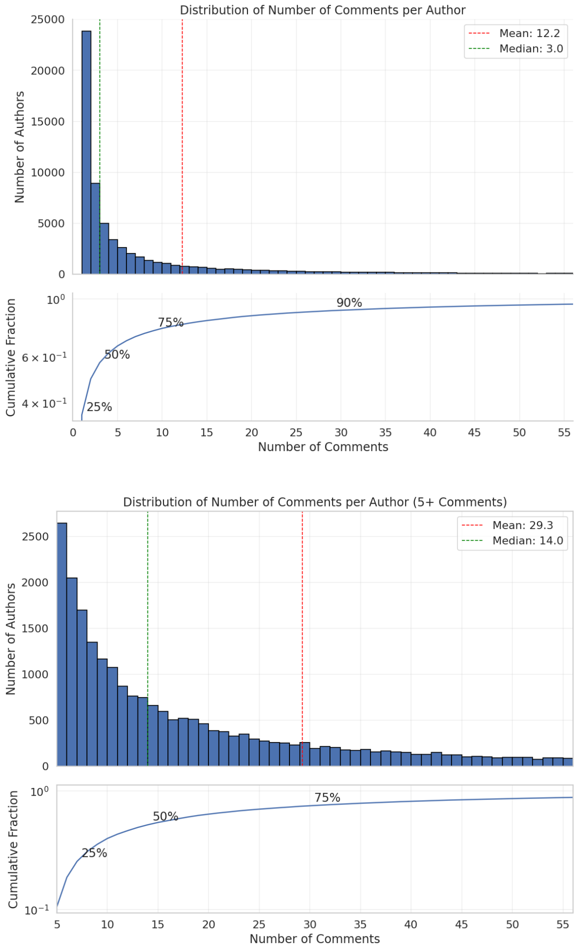


Figure 4.3: Distribution of comments per author before and after filtering for 5+ comments

4.3.1 Comment Attention Mechanism

At the comment level, each individual comment is first encoded using a pretrained multilingual BERT model. As introduced in our discussion of BERT (Section 2.3.3), we utilize the final hidden state corresponding to the [CLS] token. This provides a fixed-length vector representing an entire comment, suitable for further processing. The embedding vector denoted as $\mathbf{E}_{\text{CLS}}$ resides in $\mathbb{R}^{768}$, which corresponds to the hidden size of DistilBERT.

$$\mathbf{E}_{\text{CLS}} = \texttt{BERT}_{\text{embeddings}}(\texttt{comment})_{[\text{CLS}]} \tag{4.1}$$

To aggregate the information from all comments belonging to a single user, we take inspiration from the concept of the attention mechanism, which have become fundamental in recent breakthroughs in deep learning, particularly due to their sucess in Transformer models introduced by Vaswani et al. [VSP+17]. These models use complex multi-head attention and self-attention mechanisms.

However, for our base implementation, we propose a more ligthweight and simplified form of attention.

Attention mechanisms are proposed as a way to allow machine learning models to learn how to assign different importance to different parts of the input data. Various types of attention mechanisms have been used for different machine learning tasks, ranging from natural language processing to computer vision, including tasks like image classification, machine translation or speech recogniction.[NZY21]

In our implementation the concept of attention is implemented in a simplified manner using PyTorch neural networks.[1]

In this simplified attention mechanism, the user's comment embeddings, denoted as $\mathbf{E}_{\text{CLS},i}$ for the i-th comment, are processed using a dropout layer (`nn.Dropout`) followed by a single linear layer (`nn.Linear`), which outputs scalar attention scores. These attention scores are normalized into weights using a softmax function (`F.Softmax`). Finally, a weighted sum of the comment embeddings is calculated using these attention weights, producing the final user-level representation.

$$s_i = \mathbf{w}_{\text{att}} \cdot \mathbf{E}_{\text{CLS},i} + \mathbf{b}_{\text{att}} \qquad \text{(Linear Layer)} \tag{4.2}$$

$$\alpha_i = \frac{\exp(s_i)}{\sum_j \exp(s_j)} \qquad \text{(Softmax Normalization)} \tag{4.3}$$

$$\mathbf{V} = \sum_i \alpha_i \mathbf{E}_{\text{CLS},i} \qquad \text{(Weighted Sum)} \tag{4.4}$$

Here, $\mathbf{w}_{\text{att}}$ is a learnable weight vector and $\mathbf{b}_{\text{att}}$ is a learnable scalar bias of the linear layer. The learnable weight vector $\mathbf{w}_{\text{att}}$ resides in $\mathbb{R}^{768}$, matching the size of the embeddings.

The s_i value represents the attention score for the i-th comment, and α_i is the normalized attention weight for the i-th comment. The final user-level representation $\mathbf{V}$ is the weighted sum of the individual comment embeddings.

Although this is a simplified approach, it serves as a baseline for our approach. To further extend on the idea, we also epxerimeneted with an enhanced version where we extend the architecture by intorducing a neutral network transofmration for the comment embeddings before calculating the attetion scores.

4.3.2 Enhanced Attention Mechanism

In this enhanced attention mechanism, the user's comment embeddings, denoted as $\mathbf{E}_{\text{CLS},i}$ for the i-th comment, are first processed through a feedforward neural network with two linear layers (`nn.Linear`) with ReLU activation functions (`nn.ReLU`) after each linear layer. This is followed by an the same attention mechanism as before, using a dropout layer (`nn.Dropout`) and a linear layer (`nn.Linear`) that outputs scalar attention scores, which are normalized into weights using a softmax function (`F.Softmax`).

[1] https://docs.pytorch.org/docs/stable/index.html

$$\mathbf{H}_i = \text{ReLU}(\mathbf{W}_1 \mathbf{E}_{\text{CLS},i} + \mathbf{b}_1) \quad \text{(First Linear Layer with ReLU)} \tag{4.5}$$

$$\mathbf{E}_{\text{MLP2},i} = \text{ReLU}(\mathbf{W}_2 \mathbf{H}_i + \mathbf{b}_2) \quad \text{(Second Linear Layer with ReLU)} \tag{4.6}$$

$$s_i = \mathbf{w}_{\text{att}} \cdot \mathbf{E}_{\text{MLP2},i} + \mathbf{b}_{\text{att}} \quad \text{(Attention Linear Layer)} \tag{4.7}$$

$$\alpha_i = \frac{\exp(s_i)}{\sum_j \exp(s_j)} \quad \text{(Softmax Normalization)} \tag{4.8}$$

$$\mathbf{V} = \sum_i \alpha_i \mathbf{E}_{\text{MLP2},i} \quad \text{(Weighted Sum)} \tag{4.9}$$

In this case $\mathbf{W}_1$ and $\mathbf{W}_2$ are the learnable weight matrices of the two linear layers, and $\mathbf{b}_1$ and $\mathbf{b}_2$ are the corresponding biases. Where $\mathbf{W}_1 \in \mathbb{R}^{h \times 768}$, $\mathbf{b}_1 \in \mathbb{R}^h$ and $\mathbf{W}_2 \in \mathbb{R}^{768 \times h}$ and $\mathbf{b}_2 \in \mathbb{R}^{768}$. The first linear layer transforms the comment embeddings into a hidden dimension h, while the second linear layer transforms them back to the original embedding size of 768. The attention mechanism is then applied to the transformed embeddings, similar to the previous approach.

This enhanced version tries to leverage the high-dimension representation of the comment embeddings, which encodes various semantic feautres of the comments. By introducing non-linear transformations, the model can potential capture richer features, such as sentiment, language patterns or other semantic aspects.

4.3.3 Regression Head

As the final step of the calculation a regression head is applied on top of the aggregated user-level representation vector. The regression head is implemented using PyTorch and as a a feed-forward neural network (`nn.Sequential`) composed of two fully connected linear layers (`nn.Linear`) with a ReLU activation function (`nn.ReLU`) and a dropout layer (`nn.Dropout`) between them. This network takes the aggregated user-level representation $\mathbf{V}$ as input and produces a single scalar output representing the predicted trolliness score for the user.

$$\mathbf{h} = \text{ReLU}(\mathbf{W}reg, 1\mathbf{V} + \mathbf{b}reg, 1) \quad \text{(Linear Layer with ReLU)} \tag{4.10}$$

$$\mathbf{h}' = \text{Dropout}(\mathbf{h}) \quad \text{(Dropout Layer)} \tag{4.11}$$

$$\hat{y} = \mathbf{W}reg, 2\mathbf{h}' + \mathbf{b}reg, 2 \quad \text{(Final Linear Layer)} \tag{4.12}$$

Here $\mathbf{W}_{\text{reg},1} \in \mathbb{R}^{h \times d}$ and $\mathbf{b}_{\text{reg},1} \in \mathbb{R}^h$ are the learnable weights and biases of the first fully connected linear layer, where h is the hidden dimension of the regression head d is the size of the aggregated user representation $\mathbf{V}$. Similarly, $\mathbf{W}_{\text{reg},2} \in \mathbb{R}^{1 \times h}$ and $\mathbf{b}_{\text{reg},2} \in \mathbb{R}$ are the learnable weights and bias of the second fully connected layer.

The scalar output $\hat{y}$ represents the predicted trolliness score for the user. Notably, $\hat{y}$ is not directly bound between 0 and 1 because it is the raw output of a linear layer. However since the model is trained using Binary Cross-Entropy with Logits Loss (BCE Loss), which internally applies a sigmoid function to $\hat{y}$, the model learns to produce vlaues that correspond to the legit of a probability. At inference time, a sigmoid function is explicitly applied to $\hat{y}$, transforming it into a probability value between 0 and 1.

4.3.4 Loss Function

The model is trained using Binary Cross-Entropy with Logits Loss (BCE Loss) also provided by PyTorch (`nn.BCEWithLogitsLoss`). This loss function combines a Sigmoid layer and the Binary Cross-Entropy Loss in a single class, which is numerically more stable than using them separately. The formula of the underlying Binary Cross-Entropy Loss is defined as follows:

$$L_{\mathrm{BCE}}(y, \hat{y}) = -\left[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})\right] \tag{4.13}$$

Here, y is the ground-truth label, and $\hat{y}$ is the predicted score for the user, which, when working with BCE Loss, is interpreted as the probability of the user having the troll label.

The decision to use BCE Loss, despite framing the problem as a regression task, was motivated by the fact that in the training data we work with binary labels and not continuous target values. Unlike standard regression losses (like Mean Squared Error or Huber Loss), BCE Loss is specifically designed to handle cases where the target values are binary, but the model's predictions are continuous probabilities. This setup helps the model avoid becoming overly biased towards low values, which was a problem with a test run with HuberLoss.

4.4 Training

The training of the model is done in two steps. Larger training on the large labeled troll datasets from foreign domains, and a smaller fine-tune on manually annotated Czech comments from the target dataset.

The first training step includes the Russian IRA troll tweets, information operations datasets, and non-troll datasets like Civil Comments. The training is done using a regression objective, where the model is trained to predict the trolliness of the users instead of their binary class.

Since the labeled training data comes from different domains and languages than our target Czech dataset, a second small fine-tuning step is performed.

After the initial training, the model is fine-tuned on a small set of manually annotated Czech user comments from our target dataset. I created this data by exploring the users classified with high or low trolliness scores and high confidence during preliminary runs. This few-shot tuning step helps the model better adapt to our specific domain.

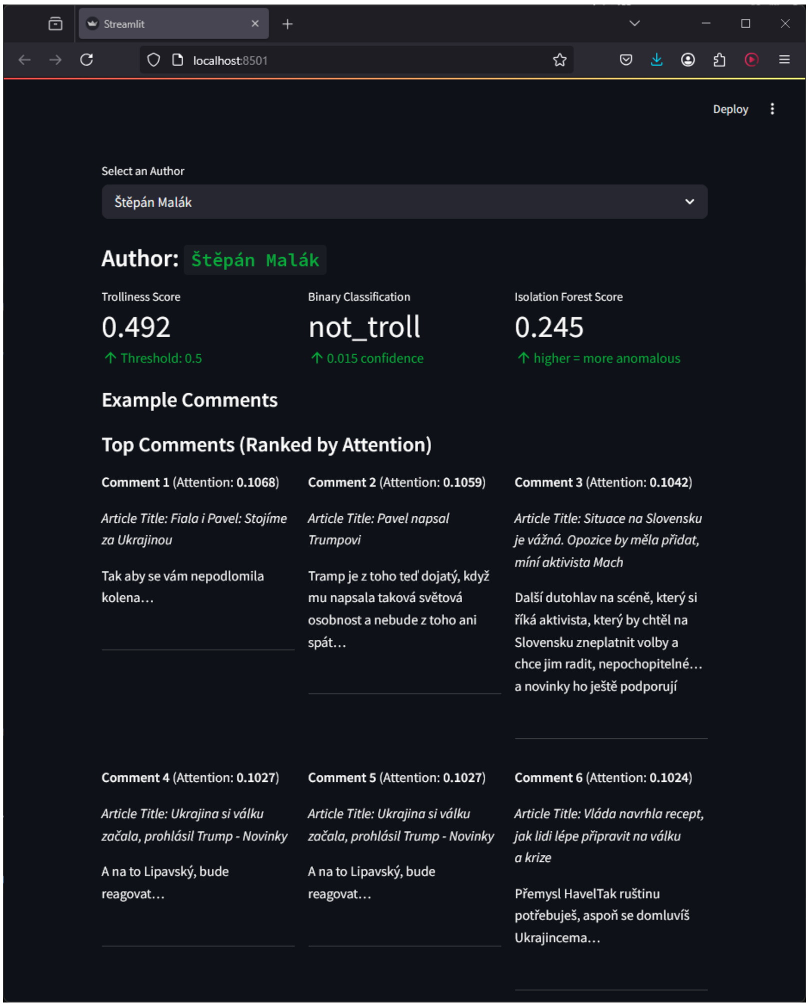
4.5 Lightweight Annotation and Evaluation App

We also created a simple annotation application to help with working with the model and dataset. The main goal of the tool is to allow labeling of users from the Czech dataset and searching for their comments. The app also shows the model's predicted trolliness scores and attention weights. The annotations can then be used both for few-shot fine-tuning and for manual exploration of the model's predictions.

The app loads a saved model checkpoint and available Czech comments from the dataset. It then allows a user to search for an author by name and displays the predicted trolliness score of the author and all of their comments, along with their attention weights. Finally, the app allows the user to label the author as a troll, a non-troll, or uncertain and saves the labels to a file. The app is implemented in Python using the Streamlit library, which allows for easy creation of interactive web applications.

We used the app to manually label a small set of users from the Czech dataset. We focused on users with high or low trolliness scores, as well as those with uncertain scores. The task proved to be quite challenging. Most users tended to be quite negative and angry in general, as mentioned in the previous section, but it was still difficult to rate them as trolls.

The manual labelling highlighted the inherent difficulty of the task of distinguishing between toxic behaviour and genuine disagreement in online discussions. The challenge of recognizing trolls from other disruptive or even genuine but negative users could be challenging even for political science or psychology experts, and goes beyond this thesis's scope. Despite the annotation's complexity and time-consuming nature, we created a small labeled dataset for further use.



Select an Author

Štěpán Malák

Author: `Štěpán Malák`

Trolliness Score
0.492
↑ Threshold: 0.5

Binary Classification
not_troll
↑ 0.015 confidence

Isolation Forest Score
0.245
↑ higher = more anomalous

Example Comments

Top Comments (Ranked by Attention)

Comment 1 (Attention: **0.1068**)

Article Title: Fiala i Pavel: Stojíme za Ukrajinou

Tak aby se vám nepodlomila kolena…

Comment 2 (Attention: **0.1059**)

Article Title: Pavel napsal Trumpovi

Tramp je z toho teď dojatý, když mu napsala taková světová osobnost a nebude z toho ani spát…

Comment 3 (Attention: **0.1042**)

Article Title: Situace na Slovensku je vážná. Opozice by měla přidat, míní aktivista Mach

Další dutohlav na scéně, který si říká aktivista, který by chtěl na Slovensku zneplatnit volby a chce jim radit, nepochopitelné… a novinky ho ještě podporují

Comment 4 (Attention: **0.1027**)

Article Title: Ukrajina si válku začala, prohlásil Trump - Novinky

A na to Lipavský, bude reagovat…

Comment 5 (Attention: **0.1027**)

Article Title: Ukrajina si válku začala, prohlásil Trump - Novinky

A na to Lipavský, bude reagovat…

Comment 6 (Attention: **0.1024**)

Article Title: Vláda navrhla recept, jak lidi lépe připravit na válku a krize

Přemysl HavelTak ruštinu potřebuješ, aspoň se domluvíš Ukrajincema…

Figure 4.4: Annotation app interface

Chapter 5
Experiments

The experiments conducted to evaluate the proposed method are presented in this chapter. The main objective of these experiments is to validate the performance of the proposed BERT-based model and to explore its ability to transfer knowledge to a new language.

The experiments are divided into two main stages. The first stage will compare the proposed method with a baseline model. The goal is to evaluate the performance of the BERT-based model against a traditional machine learning approach using stylometric features.

Given the multilingual nature of the BERT model used (DistillBERT-Multilingual-Cased), the second stage will investigate the model's ability to transfer knowledge across languages. Specifically, the model trained on the international troll dataset will be directly applied to the Czech dataset. This approach aims to leverage the inherent multilingual capabilities of multilingual BERT models, as demonstrated by Pires et al. [PSG19], who showed that M-BERT can achieve surprising levels of cross-lingual generalization, even across languages with different scripts. This characteristic is essential for our task, as we try to have the model learn from English and other confirmed troll data and to carry this knowledge into our target Czech domain.

Additionally, we experiment further with fine-tuning approaches to adapt the model to the Czech language and domain. As we will recognize, the direct application of the multilingual model may not achieve optimal results.

5.1 Baseline Comparison

To evaluate the reasonability and effectiveness of the proposed method, we first established a set of baseline models using traditional machine learning techniques. The primary goal of this comparison is to validate the feasibility of a BERT-based approach for a troll detection task.

The baseline model consists of two main types:

- **Stylometric-Only Models** - These models rely on a set of manually selected stylistic features, specifically chosen based on their simplicity and use in related work. The features are:
  - Character count

  - Word count
  - Average word length
  - Capital-letter ratio
  - Digit ratio

  These features we used to train two machine learning models:

  - Linear Support Vector Regressor (SVR)
  - Gradient Boosting Regressor (GBR)

- **TF-IDF + Ridge Regression Model** - Furthermore, a more advanced baseline model was created using Term Frequency - Inverse Document Frequency (TF-IDF) to represent the comments. This model transforms the text into a high-dimensional vector space, where each dimension corresponds to a unique word or n-gram in the corpus. The TF-IDF vectors were then used to train a Ridge Regression model. Overall, this model was trained on 50,000 unigrams and bigrams extracted from the comments.

The models and feature choices were inspired by the work of Machová et al. [MPH21].

Finally, for comparison, the BERT-based model was also trained on the same train/test/validation splits. This model uses a regression head with a Sigmoid activation function and Binary Cross-Entropy Loss (BCE Loss) as the loss function.

5.1.1 Baseline Evaluation Metrics

To compare the performance of the models, we choose to compare two metrics: the Mean Squared Error (MSE) and the Coefficient of Determination (R^2 score). The MSE is a measure of the average squared difference between the predicted and actual values. It is defined as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5.1}$$

where y_i is the true value, $\hat{y}_i$ is the predicted value, and n is the number of samples. The MSE is a common loss function used in regression tasks and measures how well the model's predictions align with the actual values. A lower MSE indicates better performance.

The R^2 score is a statistical measure of what share of the original variance of the data the model explains. To break down how it works, we first define two equations.

The Total Sum of Squares (TSS):

$$\text{TSS} = \sum_{i=1}^{n}(y_i - \bar{y})^2, \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{5.2}$$

And the Residual Sum of Squares (RSS):

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{5.3}$$

The R^2 score is then:

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}} \tag{5.4}$$

The equation measures the proportion of total variance (TSS) left unexplained by the model's residuals (RSS). A value of 1 indicates a perfect fit, and zero indicates that the model does not explain any of the variance in the data. A negative R^2 score indicates that the model performs worse than simply guessing the mean of the target variable.

5.1.2 Baseline Results

The Support Vector Regressor (SVR) and the Gradient Boosting Regressor performed poorly with the simple stylometric features. They did not beat guessing the mean and achieved a negative R^2 score and an MSE of about 0.26. This result was not entirely surprising, given that the models relied only on five features. Still, it shows that a few handpicked stylometric features are insufficient for a complex task like finding troll behavior patterns.

When we extend the feature set to 50000 (uni- and bi-gram counts) through TF-IDF, performance improves significantly. The model achieves an R^2 score of 0.57 with a mean square error of 0.09. Despite this, the TF-IDF model remains fundamentally constrained to the English language and the specific domain it was trained on.

In contrast, the BERT model, which was trained on the same dataset, achieved an R^2 score of 0.61 and an MSE of 0.08. The model outperforms the two base models by significant margin. And while the TF-IDF model's performance was comparable to BERT, it is limited to the vocabulary of the training data, while a multilingual BERT transformer has the potential to carry its learned knowledge over to a new language.

The complete model training and evaluation results are shown in the table below.

Model	**Features**	**Val MSE**	**Val R^2**	**Test MSE**	**Test R^2**
Linear SVR	5 stylometric	0.260	−0.211	0.260	−0.211
Ridge TF–IDF	50 000 1-2-grams	0.090	0.579	0.097	0.547
DistilBERT	contextual	**0.087**	**0.592**	**0.082**	**0.618**

Table 5.1: Author-level results on English train/test/val data.

These results show that BERT achieves similar or better results than a traditional stylometric model, and we validate our decision to use it as our main approach.

5.2 Further Training of BERT Model

For further experiments, we trained the BERT model on a mix of all available troll comments, rather than being limited to English-language comments as in the baseline evaluation. This aims to leverage the strong cross-lingual capabilities of the multilingual BERT referenced earlier [PSG19]. Building on the findings of Piers et al. (2019), who noted that cross-lingual transfer capabilities are often enhanced between typologically similar languages, we also decided to investigate a more targeted approach. Specifically, given the typological similarities between Russian and Czech, we will present an additional training run utilizing only Russian-language data. The hypothesis here is that pre-training on a more closely related Slavic language might facilitate a more effective transfer of knowledge by the model.

The distribution of languages in the annotated datasets can be seen in Figure 5.1.

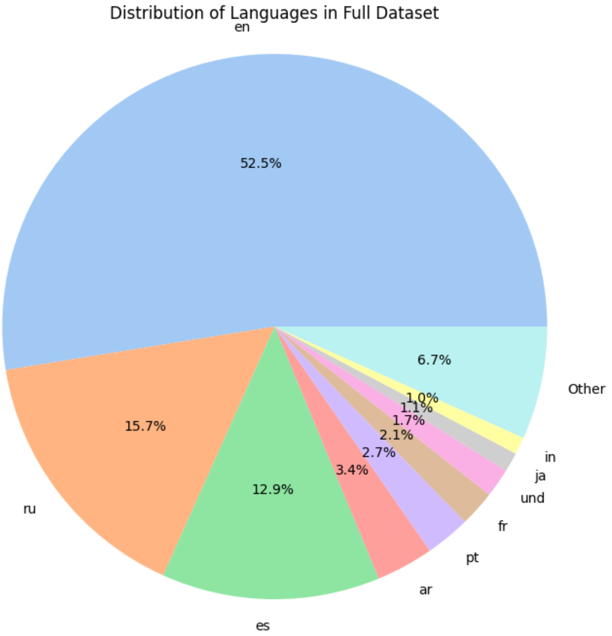


Figure 5.1: Language Distribution

To manage the size of the multilingual training dataset, two limitations were made. First, each author was limited to a maximum of 50 comments. Second, because the full dataset was heavily imbalanced towards the nontroll class with 40000 accounts as opposed to only 4000 troll accounts, and a ratio of 90% to 10% troll to non-troll comments, the dataset was balanced to achieve a 50% to 50% comment ratio and about a 65% to 35% user ratio. These limitations were not used for the Russian dataset. The statistics for the training datasets can be seen in Table 5.2 and 5.3

Multilingual Tweet Distribution			Multilingual Author Statistics		
Category	**Count**	**Percentage**	**Category**	**Count**	**Percentage**
Troll tweets	144,563	51.1%	Troll authors	4,555	35.6%
Non-troll tweets	138,340	48.9%	Non-troll authors	8,236	64.4%
Total tweets	282,903	100%	Total authors	12,791	100%

Table 5.2: Tweet distribution and author statistics in the dataset.

Russian Tweet Distribution			Russian Author Statistics		
Category	**Count**	**Percentage**	**Category**	**Count**	**Statistics**
Troll tweets	63,318	49.0%	Troll authors	2,096	30.1%
Non-troll tweets	65,987	51.0%	Non-troll authors	4,875	69.9%
Total tweets	129,305	100%	Total authors	6,971	100%

Table 5.3: Russian Tweet Distribution and Author Statistics in the Dataset.

5.2.1 Training Results

The model was trained for five epochs on a single NVIDIA RTX 3060 Ti GPU (8GB memory) using a batch size of 16 and the AdamW optimizer with a learning rate of 1e-5 and weight decay of 0.01. A linear learning rate schedule was employed. The training, utilizing PyTorch and the HuggingFace Transformers library for the BERT model, took approximately 4 hours for the multilingual run and about 2 hours for the Russian-only run. The table 5.4 and figure 5.2 below show the training results, including training/validation loss and R^2 scores over epochs.

Best Epoch Multilingual: 2			Best Epoch Russian: 3		
Metric	**Training**	**Validation**	**Metric**	**Training**	**Validation**
Loss	0.5437	0.5557	Loss	0.2740	0.3405
MSE	0.0734	0.0746	MSE	0.0653	0.0913
R^2 Score	0.6814	0.6641	R^2 Score	0.7328	0.6288
Binary Accuracy	0.9134	0.9165	Binary Accuracy	0.9091	0.8742

Table 5.4: Training and validation metrics at the best epochs.

5.3 Predictions on Czech Dataset

In this section, we evaluate the performance of the two multilingual/Russian models on the Czech dataset. Both models were trained using the methods described in the previous section.

We will assess the models in two scenarios:

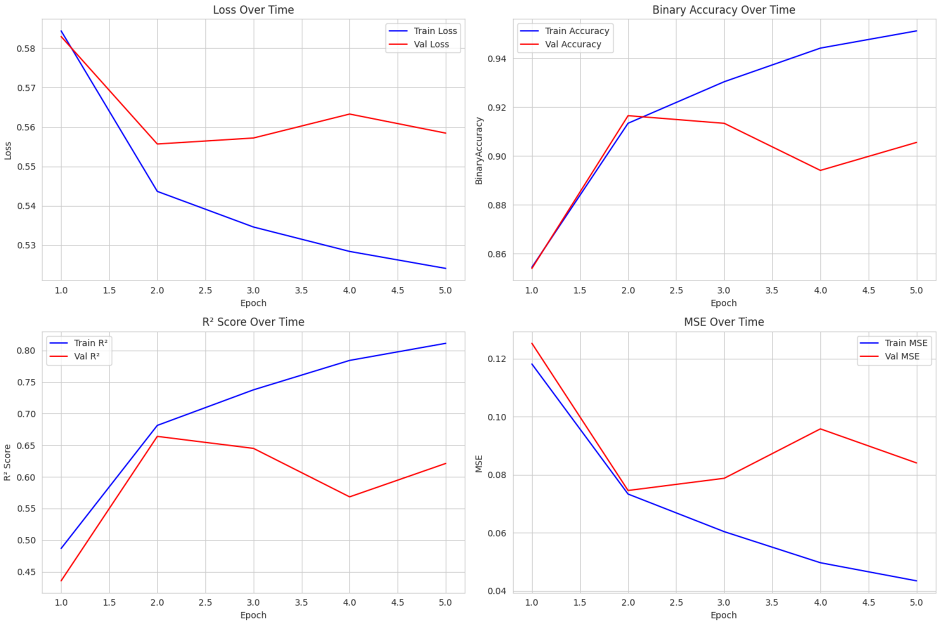


Figure 5.2: Training and validation loss over epochs - Multilingual

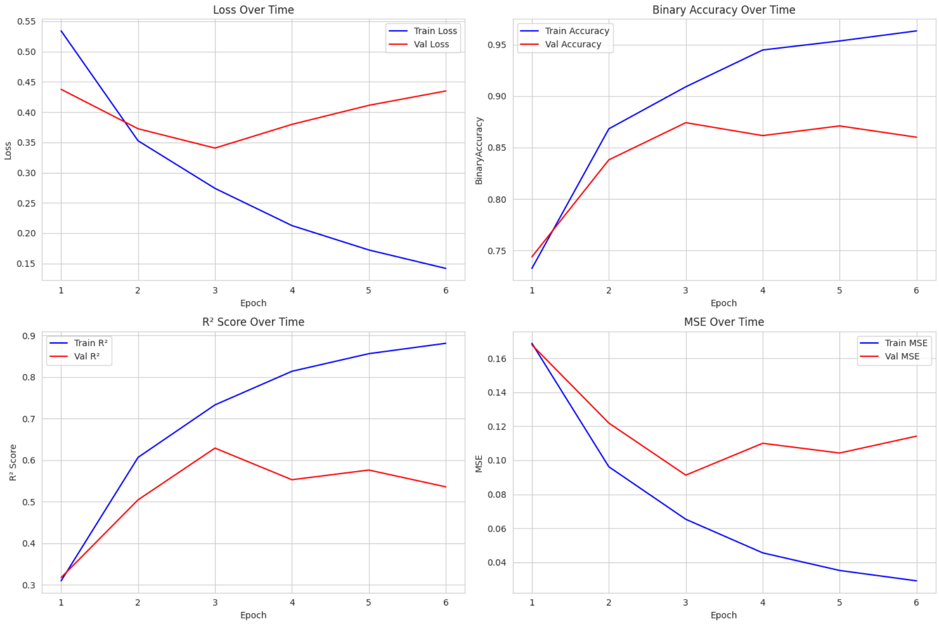


Figure 5.3: Training and validation loss over epochs - Russian only

- **Zero-Shot Predictions** - The multilingual model is applied directly to the Czech dataset without any further training or fine-tuning.

- **Fine-Tuned** - The multilingual model is further fine-tuned on a small set of manually annotated Czech users.

Each model is evaluated based on the distribution of predicted trolliness scores across users, manual review of selected user cases and performace on a small hand annotated test set.

5.3.1 Zero-Shot Experiment

We first applied both the Multilingual and Russian-only trained models directly on the Czech dataset without any further fine-tuning for the Czech news comments domain. The goal was to test their zero-shot ability to generalize to a new language and domain.

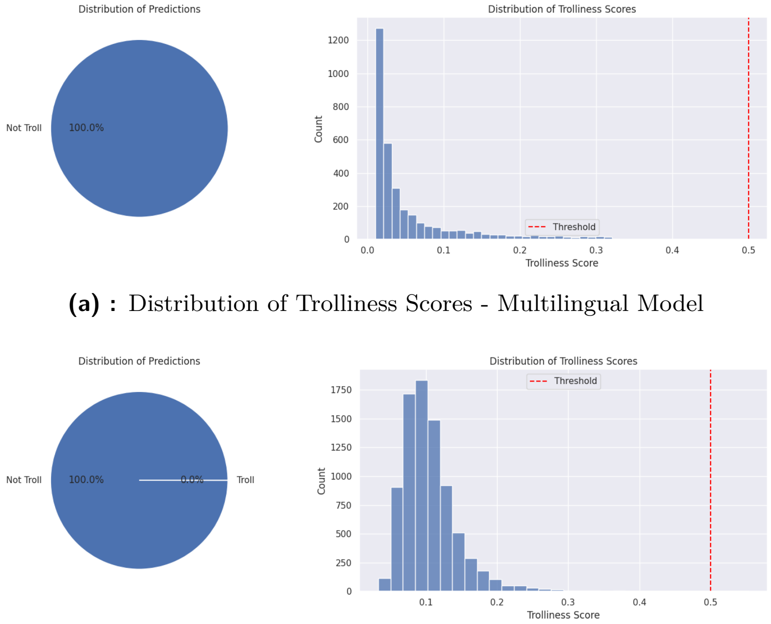
5.3.2 Results

The distribution of predicted troliness scores for both models is shown below. We observe that the Multilingual model struggles to classify trolls, as seen in Figure 5.4a, where the vast majority of predictions are close to zero. The russian model, while also having most predictions at low values, shows a slightly higher mean and a higher median and maximum score.

Model	Mean Trolliness	Std. Deviation	Median	Max
Multilingual	0.0561	0.0644	0.0270	0.3316
Russian-only	0.1056	0.0411	0.0979	0.5534

Table 5.5: Zero-Shot Trolliness Score Distribution

Figure 5.4 shows the difference in the distribution of trolliness scores between the two models. The multilingual model's distribution is sharply concentrated at near-zero values, effectively classifying most users as non-trolls. The Russian model, in contrast, has a higher median and maximum score, suggesting it is slightly better at identifying a range of troll-like behavior and classifying some users as trolls. However, the overall distributions are both heavily skewed towards low trolliness scores, indicating that both models struggle to identify troll behavior.



(a) : Distribution of Trolliness Scores - Multilingual Model

(b) : Distribution of Trolliness Scores - Russian-Only Model

Figure 5.4: Comparison of Trolliness Score Distributions between Multilingual and Russian-Only Models (Zero-Shot)

These results show that the model struggles to meaningfully capture variation of trolliness in the Czech comments and cannot generalize well to the Czech domain. There could be several factors contributing to this issue. It could be language differences, as although the model is based on a multilingual BERT model, the nuances of the Czech language may not be well represented. They may be important for the troll detection task. Additionally, the domain might be too widely misaligned, as the training data was collected from various platforms, contexts, and time frames. The missalignment of the training data could make it quite difficult for the model to use its learned knowledge on the specific Czech comments domain.

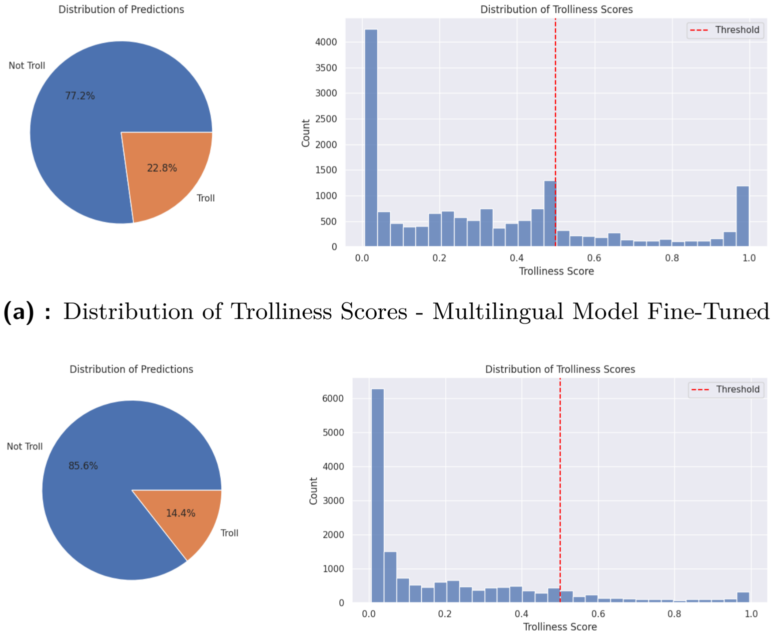
Recognizing the limitations, we explored a strategy to overcome this problem by further fine-tuning the model on a small manually annotated dataset, which was mentioned previously when describing the annotation app.

The next step, described below, demonstrated more promising results.

5.3.3 Fine-Tuned Experiment

After seeing the relatively poor performance of the zero-shot approach, we decided to try to explore the fine-tuning of the model through a few-shot training approach. The goal was to see if the model could learn to better adapt to the Czech language and domain by training on a small set of manually annotated users.

The model was trained for five epochs with a manually annotated dataset of 50 users. The distribution of trolliness scores shifted noticeably for both models, as shown in Figures 5.5a and 5.5b.



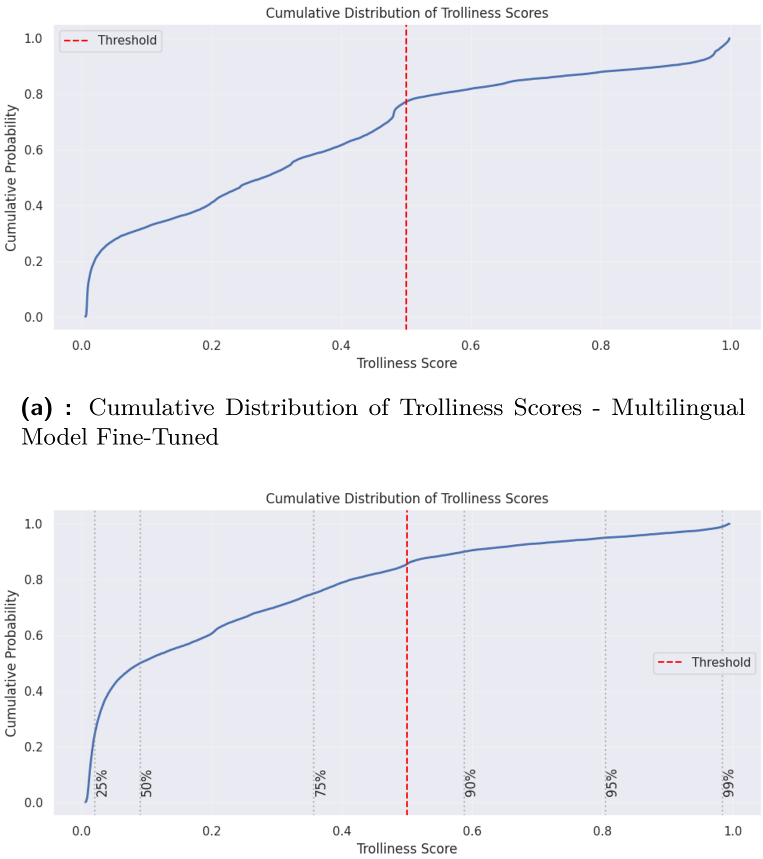
(a) : Distribution of Trolliness Scores - Multilingual Model Fine-Tuned

(b) : Distribution of Trolliness Scores - Russian-Only Model Fine-Tuned

Figure 5.5: Comparison of Trolliness Score Distributions between Multilingual and Russian-Only Models (Zero-Shot)

The results indicate improvement in the multilingually trained models' abilities to identify various levels of troll behavior and shift the distributions of scores significantly. The multilingual model now has a mean trolliness score of 0.33 with a standard deviation of 0.3, while the Russian-only model has a mean of 0.21 and a standard deviation of 0.25. Both models use the full score range from 0 to 1, unlike the zero-shot models, which had maximum scores of 0.33 and 0.55, respectively.

Both models have a large concentration of users with a trolliness score of near zero, as can be seen in the histograms of Figure 5.5a and 5.5b. Both models also tend to predict assigned values more frequently than higher ones, which can be seen in the cumulative distribution of the scores, particularly with the Russian-trained model in Figure 5.6b, where we can see that the 80th percentile of scores is less than 0.4. This conservativness might not necessarily be a negative trait, as it reduces the risk of false positives and might also align with a realistic expectation that only a minority of users are genuinely trolls. Despite the clustering around zero, both models can still assign a wide range of scores, indicating their ability to differentiate between varying levels of troll behavior.



(a) : Cumulative Distribution of Trolliness Scores - Multilingual Model Fine-Tuned

(b) : Cumulative Distribution of Trolliness Scores - Russian-Only Model Fine-Tuned

Figure 5.6: Comparison of Trolliness Score Distributions between Multilingual and Russian-Only Models (Zero-Shot)

The choice of the 0.5 classification threshold is somewhat arbitrary and was set for training purposes. It directly influences the referenced percentages of users classified as trolls, but it does not inherently define troll behavior.

It is important to note that the similarity of the distributions of the two models after fine-tuning may be a result of overfitting. Given the small size of the fine-tuning dataset, the model may have learned to reflect the specific examples it was exposed to closely, leading to very similar behavior.

5.3.4 Manual Review of Predictions

To further understand the model's predictions, we will manually review a selection of users with high, low, and uncertain trolliness scores. The goal was to see if the model's predictions aligned with our expectations and to identify any potential issues or biases in the model.

We will present a specific user example, including a selection of the user's comments, the model's assigned trolliness score and the attention weights for each comment.

To preserve the anonymity and privacy of users, we will not disclose the actual names as they are not relevant to the analysis. Instead, we will refer to the users as User A, User B, etc.

User A - Trolliness Score: 0.268 (Multilingual Model) / 0.960 (Russian Model)

- **Comment 1:**
  Ukrajince v Čechách nechceme!
  Attention Weight: 0.091 / 0.681

- **Comment 2:**
  Proč se u cizinců v zahraničí uvádí původ u takových článků a u nás v ČR jsou to pouze „cizinci"? (Všichni víme jakého pUAvodu)
  Attention Weight: 0.438 / 0.233

- **Comment 3:**
  Paráda, ospalý Joe už bude jen špatná vzpomínka. MAKE AMERICA GREAT AGAIN!
  Attention Weight: 0.006 / 0.035

User B - Trolliness Score: 0.014 (Multilingual Model) / 0.013 (Russian Model)

- **Comment 4:**
  ANO, které v Evropském parlamentu hlasuje proti Ukrajině kdykoliv mohlo? ANO jsou fakt jen klauni populističtí
  Attention Weight: 0.076 / 0.010

- **Comment 5:**
  Rusko 2008 - napadá Gruzii i když s nimi měli mírovou dohodu - Ukrajina žádá NATO o teoretickou možnost obrany - NATO to odmítá. Rusko 2014 - útočí na Ukrajinu - Ukrajina se snaží bránit (o kterých dnes víme, že byli ruští vojáci) a nemá sílu bránit ještě Krym. - tak aspoň žádá NATO o tréning - to je jim povoleno. Rusko 2022 - útočí na Ukrajinu, neb si dovolila požádat NATO o trénink, když je Rusko napadlo. TRUMP: uKrAjInA To zAčAlA. blebt blebt
  Attention Weight: 0.146 / 0.227

- **Comment 6:**
  Dan Svatek. I kdyby byl plyn z Ruska - tak Azerbajdžán nakupuje Ruský silně pod cenou... (takže Rusko na tom nevydělává a to je důležité) a kdyby ho přestali odkupovat - tak Rusko nebude mít komu ho prodávat - a to technologicky je složitější a peněžně náročnější než nakupování pod cenou...
  Attention Weight: 0.162 / 0.164

User C - Trolliness Score: 0.478 (Multilingual Model) / 0.434 (Russian Model)

- **Comment 7:**
  To bylo kecu jak jsme konečně nezávislí na tom ruském plynu, při zavření tranzitu přes UK Ukrajinskou stranou. Teď jsou ukačka bez peněz z tranzitu a celá Evropa krásně za větší cenu zasponzoruje Putinovy cestu k vítězství. Hlavně nesmí EU zapomenout udělat sankční balíček číslo 34 komici. Peníze si cestu vždycky najdou.
  Attention Weight: 0.320 / 0.538

- **Comment 8:**
  Harrisová, nepochopila že jediná věc na kterou se muže spoléhat a je jistota a to i v americe je daně a smrt jak se říkává.
  Attention Weight: 0.040 / 0.026

- **Comment 9:**
  To je blázen
  Attention Weight: 0.004 / 0.009

User D - Trolliness Score: 0.159 (Multilingual Model) / 0.024 (Russian Model)

- **Comment 10:**
  No, bylo by dobré sledovat to celorepublikově. Pokud ANO nebude cosi lidem dávat (a nevím, kde na to vezme, protože republiku ožebračili už tak, že víc nelze), znamená to tedy, že Fialova vláda lidem nic nevzala. Jasně nevzala. Jen přestala rozdávat. Neměla z čeho. Měl by se přihlásit každý, kdo od ANO něco dostane. Tedy pozitivního. Šikany rozdali dost.
  Attention Weight: 0.196 / 0.430

- **Comment 11:**
  Báťuška Hitler také nechtěl válku. Tedy alespoň na tyhle řeči nachytal spousty příznivců. Oni jsou si tak podobní, Hitler a Putin. Oba v podstatě lidumilové. Jediné, co tenhle typ (včetně Trumpa) potřebuje je situace, kdy se celý svět bude válet na zádech, hrabat v podřízené póze nožičkama do větru...
  Attention Weight: 0.234 / 0.175

- **Comment 12:**
  Ta poruchovost ruských oken začíná být opravdu fatální. Nevím, proč už tahle vypadávací dávno někdo nenamontoval do Kremlu. Byl by klid.
  Attention Weight: 0.089 / 0.033

For the manual review, we selected from a randomly sampled set of users. We selected one user with a high trolliness score, one with an uncertain score, and two with a low trolliness score. We then manually selected three comments for each of them, ensuring that each set included at least one with a high attention and one with a low attention weight. This was done to showcase the comment-level attention mechanism. Comments were chosen to try to provide explanatory value for the model's decision. Notably, each user references Ukraine in different contexts, which provides a useful basis for comparison.

5.3.5 Example User Analysis

The first user, User A, is classified as a troll by the Russian model, and his first comment receives a high attention weight. This comment is an aggressive and troll-like statement with a strong anti-Ukrainian sentiment. Interestingly, the Multilingual model assigned this comment a low attention weight, which is quite surprising given its hostile content. In the second comment, the user again expresses negative sentiment towards foreigners, and this time, the comment receives a high attention weight from both models. The third comment, which is less inflammatory, is given a relatively low attention weight by both models. However, it is still an interesting example because it features the "MAGA" slogan in all caps. The difference in attention weights here suggests that the Russian model may be more sensitive not only to topics related to Russia and Ukraine but also to American political rhetoric.

In contrast to User A, we have User B, who is consistently classified as a non-troll by both models. User C's comments are critical, focusing primarily on Czech politics, but they also mention Russia, Ukraine, and even Trump in various contexts. Despite the presence of politically charged language, including the use of all caps in some instances, the models do not classify this user as a troll. Instead, they assign him a low trolliness score.

This user provides an interesting case study because, despite the use of words like *Ukraine, Trump, Russia* and even direct criticism of the ANO political party (calling them ”clowns”), the models recognize that these comments are not inherently troll-like. Instead, they reflect the perspective of a politically engaged citizen who may be emotionally expressive but is not exhibiting troll-like behavior. The model seems to demonstrate an ability to understand the context in which charged words are used and to distinguish between genuine political criticism and hostile trolling.

5.3.6 Reflection on User Analysis

While the manual review provides some insights into how the model assigns troliness scores, it is a very limited analysis. The review is based on a very small, randomly selected sample of users and comments, of only four users out of a total of 16500. Given this limited scope and limited ability to generalize, it would be inappropriate to draw conclusions about the model's overall performance or its capabilities. For the sake of clarity and not to draw many

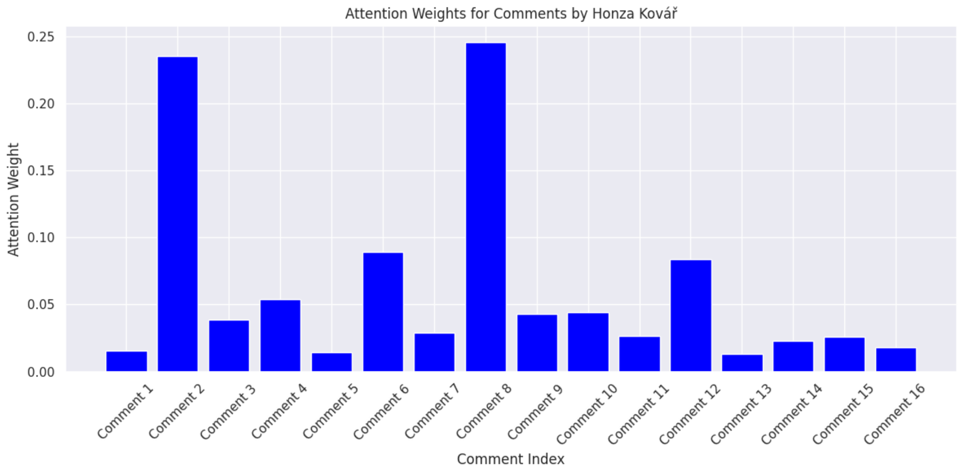


Figure 5.7: Attention weights assigned to 16 comments written by an example user. The model focuses on a handful of comments with disproportionately high weights.

conclusions, we have limited the user analysis to the comparison of the first two users, A and B. However, we have also included additional user examples, allowing readers to examine the comments, attention weights, and trolliness scores for themselves and form their own opinions.

Moreover, determining whether a user is genuinely a troll or not is a complex and subjective task that goes beyond the scope of this work. This section should therefore be understood as an illustration of the model's behavior rather than a definitive assessment of its performance.

5.3.7 Analysis of Attention Mechanism

Figure 5.7 shows the distribution of comment-level attention weights for one illustrative user. The head of the distribution is highly skewed: two comments together receive almost half of the total mass, while the long tail attracts only marginal weight. This suggests that the encoder learns to concentrate on the remarks it deems most "trolly" when constructing the user representation.

Table 5.6 juxtaposes the three comments with the highest attention weights against the three with the lowest. The *individual troll scores* shown in the table were obtained by running the same model on each comment in isolation. Because the architecture was trained to ingest multiple comments per user, these per-comment scores are only indicative and are not directly comparable to the aggregated *user* troll scores produced in the main experiments. Nonetheless, comments that attract maximal attention are strongly more trolly and their isolated troll scores approach 1.0, whereas relatively less trolly comments draw little attention and are rated as non-trolly. These examples therefore suggest that the attention mechanism is functioning as intended-highlighting the utterances that drive the final trolliness estimate.

#	Weight	Troll score	Comment text
Top 3 highest attention comments			
1	0.246	0.997	Nevidím žádné komentáře od Ssssýkory, Uncajtigové, Brože, Pelce a podobných žlutomodrých pacientů…
2	0.235	0.997	Komunistický prokurátor Stříž jako nejvyšší státní zástupce. Ukázka našeho Absurdistánu. Ještě že ani on to už nemohl vydržet…
3	0.089	0.993	Jedna banda lhářů přeskočila druhou bandu ještě větších lhářů. Na tom není nic divného.
Bottom 3 lowest attention comments			
1	0.013	0.008	Konečné slovo do pranice! Co na to naši žlutomodráci?
2	0.014	0.140	Ale jdi ty, PePo Rozvědčíku.
3	0.016	0.007	To bude první služba pro lidi, kterou by udělala.

Table 5.6: Comments with the highest and lowest attention weights together with their individual trolliness scores.

5.3.8 Enhanced Attention Model Experiments

In addition to the baseline models, we also trained a model with the enhanced version of the attention mechanism, as described in Section 4.3.2. The model with the enhanced attention mechanism was trained following the same pipeline as the baseline and was initially trained using the Russian-language only data. It was also fine-tuned exactly as the base line models on the small manually labeled Czech dataset. The training results, including training and validation loss curves, as well as the distribution of predicted trolliness scores, can be seen in the figures below.

At first glance, the richer two-layer attention block behaves more plausibly than the lightweight variant. In the zero-shot experiment, the enhanced model already produces a noticeably flatter histogram (Figure 5.9a), whereas the simplified-attention baseline all but collapsed to values below 0.1 (Figure 5.4b). This suggests that the intermediate non-linear projection helps the network harvest additional semantic cues and prevents and lets it perform better even in the zero-shot case. Although a thorough evaluation is still outstanding, these early signs are encouraging for the enhanced attention mechanism.

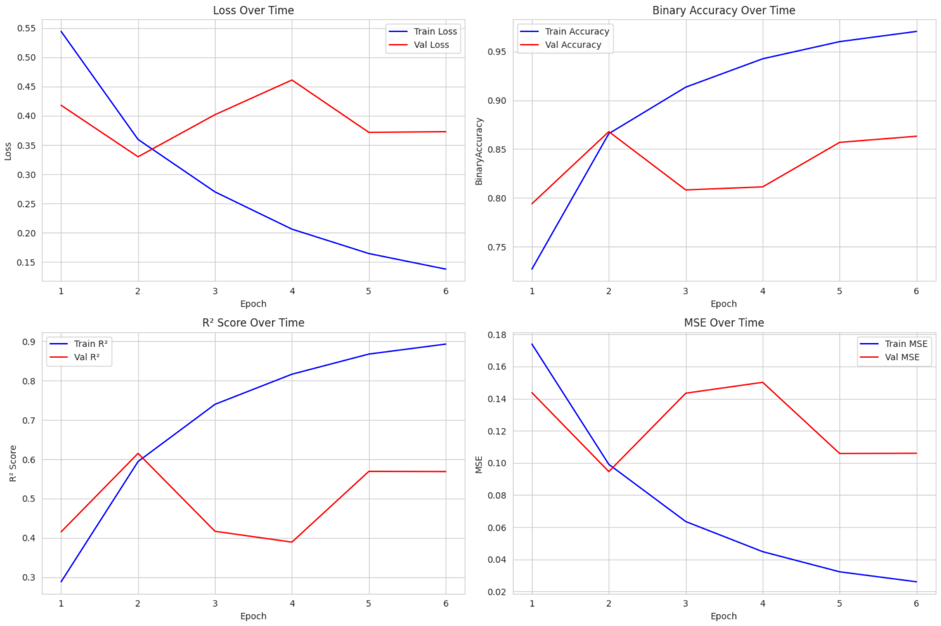
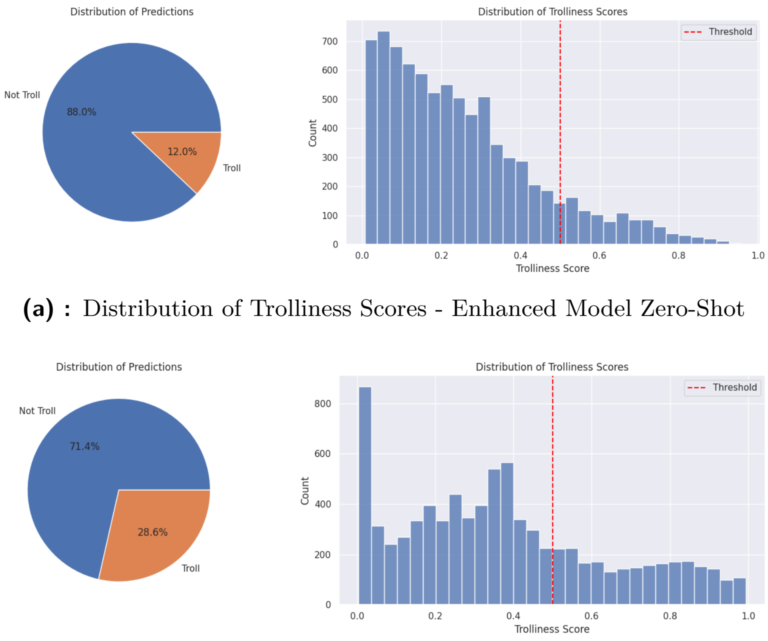


Figure 5.8: Training and validation loss over epochs - Enhanced Attention



(a) : Distribution of Trolliness Scores - Enhanced Model Zero-Shot

(b) : Distribution of Trolliness Scores - Enhanced Model Fine-Tuned

Figure 5.9: Comparison of Trolliness Score Distributions before and after Fine-Tuning

5.4 Language Adapter Experiment

As a failed experiment, we also attempted to further adapt the BERT model to the Czech language by training a language adapter. Adapters are lightweight modules that can be inserted into transformer models, allowing them to efficiently adapt to new domains with minimal changes to the main model [HGJ+19]. The approach has been shown to be effective in many contexts, including multilingual tasks. Moreover, the MetaTrolls model, previously cited in Related Methods [TZL23], successfully used adapters to specialize its model for different trolling campaigns, which inspired us to explore this method.

We trained an adapter using Masked Language Modeling (MLM) on the Czech comments dataset. The adapter was trained to predict masked tokens in the input text, with the aim to have it learn Czech-specific linguistic patterns, while retaining the multilingual knowledge and training of the BERT model.

Unfortunately, this approach did not work as expected. Instead of improving the model's performance, it disrupted it. This problem occurred because the adapter altered the outputs of the model in a way that the final regression head was not trained to handle, effectively breaking the model.

Chapter 6

Conclusion

This thesis investigated the challenge of identifying online trolling using Natural Language Processing (NLP) techniques,, focusing on discussions on the Czech news website comments. The core of the proposed methodology was using a multilingual BERT model to assign a *trolliness* score to each user, based on their comments. Initial training utilized diverse multilingual datasets of known troll and non-troll activity. This training was followed by a crucial fine-tuning stage using a small, manually annotated set of Czech news discussion comments.

The experiments showed promise for the potential of transformer-based models for this complex task. While baseline comparison demonstrated that the BERT model's performance was competitive against traditional stylometric and TF-IDF approaches, the multilingual BERT model's cross-lingual capabilities were insufficient for effective troll detection in the Czech domain without further fine-tuning. Zero-shot predictions proved insufficient, but fine-tuning, even with limited annotated data, led to noticeable improvements in the model's ability to discern varying degrees of troll-like behavior. The model's performance was further enhanced by training on a typologically similar language, in this case, Russian, which provided a more effective knowledge transfer. However, the limited size of the Czech annotated data remains a constraint, suggesting that future work would benefit from more extensive annotation, and alternatives could also be explored by utilizing different and larger transformer models like M-BERT or XLM-R.

Finally the thesis shows that our single-layer attention head can pick out the few comments that matter most, giving a path from comment-level evidence to a user-level score. The proposed enhanced two-layer version behaves similarly but spreads its scores more evenly, already flagging a few likely trolls even before fine-tuning. Both mechanisms therefore look useful for turning raw predictions into a practical, graded "user rating," though more data and testing would be needed before firm claims can be made.

Appendix A

Bibliography

[BH17] S Bradshaw and P Howard. Troops, trolls and troublemakers: A global inventory of organized social media manipulation. Technical report, 2017.

[BS12] Lance Bennett and Alexandra Segerberg. The logic of connective action: Digital media and the personalization of contentious politics. *The Logic of Connective Action: Digital Media and the Personalization of Contentious Politics*, 15:1–240, 01 2012.

[BTP14] Erin E. Buckels, Paul D. Trapnell, and Delroy L. Paulhus. Trolls just want to have fun. *Personality and Individual Differences*, 67:97–102, 2014. The Dark Triad of Personality.

[CW16] Bryn Alexander Coles and Melanie West. Trolling the trolls: Online forum users constructions of the nature and properties of trolling. *Computers in Human Behavior*, 60:233–244, 2016.

[Dae13] Walter Daelemans. Explanation in computational stylometry. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, pages 451–462, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[DBGJS21] Vlad Demsar, Jan Brace-Govan, Gavin Jack, and Sean Sands. The social phenomenon of trolling: understanding the discourse and social practices of online provocation. *Journal of Marketing Management*, 37:1–33, 03 2021.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[GPV17] Maja Golf-Papez and Ekant Veer. Don't feed the trolling: rethinking how online trolling is being defined and combated. *Journal of Marketing Management*, 33(15/16):1336–1354, November 2017.

[HGJ$^+$19] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. *CoRR*, abs/1902.00751, 2019.

[JTS21] Zidong Jiang, Fabio Di Troia, and Mark Stamp. Sentiment Analysis for Troll Detection on Weibo. *CoRR*, page 0, March 2021. arXiv:2103.09054 [cs].

[Lut98] Wincenty Lutoslawski. Principes de stylométrie appliqués à la chronologie des œuvres de Platon. *Revue des Études Grecques*, 11(41):61–81, 1898.

[LW20] D. Linvill and Patrick Warren. Troll factories: Manufacturing specialized disinformation on twitter. *Political Communication*, 37:1–21, 02 2020.

[MMV22] Kristina Machova, Marian Mach, and Matej Vasilko. Comparison of machine learning and sentiment analysis in detection of suspicious online reviewers on different type of data. *Sensors*, 22(1), 2022.

[MPH21] Kristína Machová, Michal Porezaný, and Miroslava Hreškova. Algorithms of machine learning in recognition of trolls in online space. In *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 000349–000354, 2021.

[MW64] Frederick Mosteller and David L. (David Lee) Wallace. *Inference and disputed authorship: The Federalist.* Reading, Mass., Addison-Wesley, 1964.

[NZY21] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, September 2021.

[PMMM20] Antonio Pascucci, Raffaele Manna, Vincenzo Masucci, and Johanna Monti. The role of computational stylometry in identifying (misogynistic) aggression in english social media text. In *Second Workshop on Trolling, Aggression and Cyberbullying*, 2020.

[PRKLM18] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3391–3401, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[PSG19] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert? *CoRR*, abs/1906.01502, 2019.

[RKR20] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how BERT works. *CoRR*, abs/2002.12327, 2020.

[SPN⁺24] Özgür Can Seçkin, Manita Pote, Alexander Nwala, Lake Yin, Luca Luceri, alessandro flammini, and Filippo Menczer. Labeled datasets for research on information operations, November 2024.

[SSSB20] Tal Schuster, Roei Schuster, Darsh J. Shah, and Regina Barzilay. The limitations of stylometry for detecting machine-generated fake news. In *Booktitle*, 2020.

[SSV18] Yunita Sari, Mark Stevenson, and Andreas Vlachos. Topic or Style? Exploring the Most Useful Features for Authorship Attribution. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 343–353, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[Sul04] John Suler. The Online Disinhibition Effect. *CyberPsychology & Behavior*, 7(3):321–326, June 2004.

[TZL23] Lin Tian, Xiuzhen Zhang, and Jey Han Lau. Metatroll: Few-shot detection of state-sponsored trolls with transformer adapters. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 1743–1753. ACM, April 2023.

[VSP⁺17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[YZ23] Seyhmus Yilmaz and Sultan Zavrak. A context-sensitive word embedding approach for the detection of troll tweets, 2023.