Toxic-Sentiment Analysis for live Chat

Victoria Grosu
881177

Chiara Pesce
882078

Francesco Perencin
880106

Tommaso Talamo
875496

1 Introduction

The problem we intend to address is the *detection of toxic comments in online chats*, with the aim of creating a predictive model capable of evaluating the presence of different kinds of toxic behaviors. With the growing number of users and contributions online, content moderation becomes an increasingly complex challenge and such a model can be a valuable tool for websites administrators, allowing them to promptly identify and address insults, threats, etc. The presence of toxic comments can contribute to an unpleasant online environment and this model can help to grow a more respectful and inclusive online community. Automating the detection of inappropriate behaviors reduces the need for manual processes, enabling faster intervention and optimization of human resources to address the most complex moderation challenges. Such a model can be used in many online contexts, improving the efficiency of moderation and making digital spaces safer and more welcoming. Creating predictive models for recognizing comment toxicity represents an interesting challenge in computational linguistics research, as it requires subtle understanding of natural language and its social aspects. Our project aims to address a relevant and current problem, providing practical solutions to improve the online environment and contribute to the development of advanced natural language analysis models with applications in different digital contexts.

2 Related work

In the domain of sentiment analysis within Natural Language Processing, a lot of methodologies has been explored to solve this problem. One of the earlier approaches was to utilise Convolutional Neural Networks (CNNs). Georgakopoulos et al., 2018)[3] exemplified this in their paper.

However, the landscape of sentiment analysis has witnessed the introduction of innovative architectures, such as recursive neural tensor networks pioneered by Socher et al., (2013) [9]. Socher et al., (2013) [9]'s seminal work marked a shift in the architectural paradigm for sentiment analysis, showcasing the potential of recursive neural tensor networks in this field.

Further studies, exemplified by Colon-Ruiz and Segura-Bedmar, (2020) [1] emphasized the superiority of bi-LSTM architecture over CNN-based structures in sentiment analysis, especially in scenarios like drug review sentiment analysis. In subsequent research, Naseem et al., (2020) [6] proved the effectviness of incorporating diverse embeddings alongside bi-LSTM and attention mechanisms in sentiment analysis tasks. However, these approaches have predominantly focused on binary classification, specifically discerning between positive and negative sentiments.

Additionally, the study presented by Murali et al., (2020) [5] focused on evaluating the impact of employing BERT and bi-LSTM architectures for toxic sentiment analysis. However this architecture has been primarily utilized for binary classification tasks, specifically distinguishing between toxic and non-toxic comments.

In summary, there's a notable scarcity or limited utilization of the BERT and bi-LSTM with attentional mechanism architecture in the field of multilabel toxic comment classification.

3 Your approach

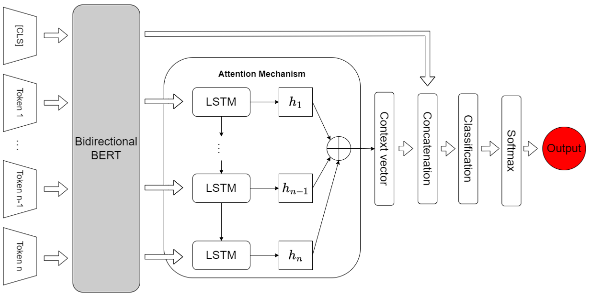
The process begins with taking a sentence as input, to which are then applied various preprocessing techniques, such as tokenization, stemming, lemmatization, or other text cleaning procedures. The pre-processed sequence is fed to a BERT transformer Devlin et al., (2018) [2], which generates token representations for each word in the sentence. These representations from BERT are then used as inputs for a bidirectional LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Unit) network, likely to capture sequential information and dependencies in the text. Afterward, an attention mechanism Vaswani et al., (2017) [10] is performed to obtain a context vector, which is concatenated with the [CLS] token produced by BERT. This combination aims to emphasize important parts of the input sentence for classification. The concatenated vectors are passed through a multilabel classification layer, potentially a fully connected neural network, and as output will return a vector.

The output vector serves as a representation of the sentence's toxicity. If all elements in the output vector are zeros, it means the sentence is to be considered non-toxic, otherwise, each non-zero element in the vector highlightens the type of toxicity present in the comment.

This pipeline is designed for multilabel text classification, where the model not only determines if a sentence is toxic but also identifies the type(s) of toxicity present within it.

Using this model we can analyze the frequency, types, and context of toxic comments within a particular online community can provide valuable insights into the prevailing behavior patterns. Understanding the prevalent toxicity levels, the nature of toxic behavior, and the topics triggering such behavior can help in devising strategies for moderation, fostering healthier discussions, and implementing community-specific interventions. By tracking and analyzing individual user behavior in terms of the toxicity of their comments, you can create profiles that indicate the likelihood of a person engaging in toxic behavior.

Graphical Abstract The main model is proposed by Lee and Lee, (2019) [4]. This model operates by initially receiving a sentence as input and using BERT to generate embeddings for individual words. Subsequently, a bi-LSTM layer scrutinizes each word within the sentence. The outputs from this layer are aggregated via an attention mechanism to form a context vector. This context vector is augmented by concatenating it with the CLS token, thereby enriching the understanding of the sentence's overall structure. Finally, through the classification and softmax steps, the model processes the available results, highlighting the most prominent toxic sentiments expressed within the text.



3.1 Competitors

We selected two distinct papers as competitors for comparison. The first one, Saeed et al., (2018) [8], holds particular relevance as it utilizes the same dataset we propose and focuses on toxic sentiment analysis classification. While they employed CNN models, they also experimented with bi-LSTM and bi-GRU models. However, our architecture differs by incorporating an attention mechanism to enhance performance and achieve superior results.

The second paper, Pham-Hong and Chokshi, (2020) [7], was chosen due to its similarity in approach to ours. They employ BERT for feature extraction, then they feed the word embeddings into an LSTM model, concatenate the results with the [CLS] token, and utilize this concatenation for subsequent classification steps, aligning closely with our proposed methodology.

3.2 Schedule

Using a collaborative approach, our project will leverage the different skills of each team member. The program outlined below will guide the execution of our project.

- **Data preparation (1 week):** Rigorous data cleaning and preprocessing will ensure a high quality training dataset.

- **Defining the objectives of the model (3 days):** The objectives of our model will be precisely outlined, laying the foundations for the subsequent phases.

- **Model creation (2 weeks):** The team will collectively start building the model, selecting an appropriate NLP architecture as the basis for subsequent training.

- **Model Training (1 week):** Collaboratively fine tune hyperparameters, considering learning rates, batch sizes, and optimization algorithms, aiming for optimal model performance.

- **Model Testing (5 days):** As part of a unified effort, we will test the trained model on a separate dataset, validating its effectiveness in real-world scenarios.

- **Documentation (5 days):** We will thoroughly document the process, capturing design choices, model architecture details, and guidelines for future use. This documentation serves as both a record and a resource for knowledge transfer within the team.

It's important to note that the provided time estimates for executing these tasks are approximations. The actual time required may vary based on factors such as dataset size, model training convergence and unexpected challenges encountered during implementation.

4 Experiments

In the context of sentiment analysis for live chat, it's crucial to make the model adapt to the dynamic and real-time nature of conversational interactions. Here are the key settings and evaluation metrics to consider:
Accuracy Metrics: Utilize standard accuracy metrics such as precision, recall, and F1 score to evaluate the model's ability to correctly classify sentiments.
Real-time Performance: Assess the model's real-time performance by measuring the time it takes to analyze and categorize sentiment in live chat messages.
Confidence Level Assessment: Implement confidence level indicators to understand the model's certainty in its predictions, especially in cases where sentiment expression is ambiguous.
Adaptability to New Data: Test the model's adaptability to new data and its ability to maintain accurate sentiment analysis as the live chat evolves.

5 Results

Tackling the challenge of online comment moderation through toxicity analysis requires a clear and realistic understanding of the performance we expect from our model. Let's explore the expectations and the rationale behind them.

First and foremost, we expect accuracy metrics between 85% and 90%, reflecting the overall precision of our model in classifying sentiments in comments. This anticipation is based on the robustness of our NLP architecture and the quality of our dataset, assuming the model's strong capability to capture linguistic nuances and adapt to various online communication styles.

Regarding real-time performance, our goal is an average response time per message of less than 100 milliseconds. This objective is crucial to maintain a seamless user experience during real-time interactions, as seen in online chat scenarios. Lastly, regarding adaptability to new data, we expect a performance loss of less than 10% over time. This assumes a continuous training process with fresh data to maintain the model's relevance in the long run. In conclusion, these expectations will serve as a guide during the implementation and validation phase of the model, but continuous monitoring remains crucial to adapt to any variations in the online context and to sustain high performance in our moderation system.

6 Data

Our project aims to develop a predictive model capable of evaluating the inclination to toxic behaviors and comments on live chats of different streaming platforms. The data used to train and evaluate the model is freely accessible from kaggle **toxic comment classification challenge**.

The dataset is composed by annotated comments sourced from Wikipedia and manually evaluated. Each sentence has a corresponding array which emphasizes the different classes of toxicity (it may have more than one): toxic, severe toxic, obscene, threat, insult and identity hate.

If a sentence is not classified as toxic then the array has all values set to zero.

The dataset is divided in three files:

- train.csv: It contains the training set's comments and their corresponding array (label).

- test.csv: It contains only the comments without any label.

- test label.csv: It contains for each row in the test the corresponding true label.

The selected dataset is highly appropriate for our goal of building a predictive model of comment toxicity. The variety of categories represented in the data allows us to create a sophisticated model that can distinguish between different types of behaviors. Moreover, the considerable size of the training dataset provides a solid foundation for training complex and accurate models.

7 Tools

Our research endeavors in sentiment analysis will be conducted utilizing *Google Colab* as our primary platform. Leveraging the collaborative nature and cloud-based infrastructure of Colab will facilitate seamless collaboration and resource-efficient execution of our tasks.

For deep learning implementations, we will harness the power of *PyTorch*, a versatile and widely-used deep learning framework known for its flexibility and extensive community support. The integration of the *Hugging Face Transformers* library will enable us to easily access and employ state-of-the-art transformer-based models such as BERT, thereby enhancing the sophistication of our sentiment analysis models.

Additionally, the adoption of *NumPy* will be pivotal for efficient numerical operations, while *scikit-learn* will be instrumental in providing robust tools for machine learning, including metrics calculation and model evaluation.

Lastly, the use of the pandas library will streamline data manipulation and preprocessing tasks, ensuring a structured and organized workflow throughout our sentiment analysis pipeline.

References

[1] Cristóbal Colón-Ruiz and Isabel Segura-Bedmar. Comparing deep learning architectures for sentiment analysis on drug reviews. *Journal of Biomedical Informatics*, 110:103539, 2020.

[2] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Spiros V Georgakopoulos, Sotiris K Tasoulis, Aristidis G Vrahatis, and Vassilis P Plagianakos. Convolutional neural networks for toxic comment classification. In *Proceedings of the 10th hellenic conference on artificial intelligence*, pages 1–6, 2018.

[4] L. H. Lee and P. L. Lee. NCUEE at MEDIQA 2019: Medical Text Inference Using Ensemble BERT-BiLSTM-Attention Model. *Proceedings of the 18th BioNLP Workshop and Shared Task*, 2019.

[5] Sourabh Raja Murali, Sanketh Rangreji, Siddhanth Vinay, and Gowri Srinivasa. Automated ner, sentiment analysis and toxic comment classification for a goal-oriented chatbot. In *2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)*, pages 1–7. IEEE, 2020.

[6] Usman Naseem, Imran Razzak, Katarzyna Musial, and Muhammad Imran. Transformer based deep intelligent contextual embedding for twitter sentiment analysis. *Future Generation Computer Systems*, 113:58–69, 2020.

[7] B. T. Pham-Hong and S. Chokshi. BERT-LSTM with Tweets' Pretrained Model and Noisy Student Training Method. *Nome della Rivista*, 2020. Sottomesso per la pubblicazione.

[8] Hafiz Hassaan Saeed, Khurram Shahzad, and S. Chokshi. Overlapping Toxic Sentiment Classification using Deep Neural Architectures. *IEEE computer society*, 2018.

[9] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention Is All You Need. *Advances in Neural Information Processing Systems*, 2017.