Contextual BERT Model for Toxicity Detection in Messaging Platforms

Arthur Buzelin[1], Yan Aquino[1], Pedro Bento[1], Lucas Dayrell[1], Victoria Estanislau[1], Samira Malaquias[1], Pedro Dutenhefner[1], Luisa G. Porfírio[1], Pedro B. Rigueira[1], Caio Souza Grossi[1], Guilherme H. G. Evangelista[1], Gisele L. Pappa[1], Wagner Meira Jr[1]

[1] Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte – MG – Brazil

{arthurbuzelin, yanaquino, pedro.bento, lucasdayrell}@dcc.ufmg.br

{samiramalaquias, victoria.estanislau, luisagontijo}@dcc.ufmg.br

{pedrobacelar.rigueira, guilherme.evangelista}@dcc.ufmg.br

{caio.grossi, glpappa, meira}@dcc.ufmg.br, pedroroblesduten@ufmg.br

Abstract. *The increasing prevalence of messaging platforms has created new challenges in hate speech detection. Traditional classification models designed for social media posts often fall short in these environments due to the lack of contextual information. This paper presents a novel approach to message classification by integrating contextual data from preceding messages, utilizing a fine-tuned BERT model based on PySentimiento. Our results demonstrate that incorporating preceding messages substantially improves the classification task. The average AUC-ROC increased from 0.691 with the PySentimiento base model to 0.784 with standard fine-tuning, and further to an impressive 0.926 with our context-based model.*

1. Introduction

Research on social media has always been a popular area of study, particularly those focused on classifications of some kind. However, it was only recently that the analysis of messaging platforms gained significant attention. Traditionally, classification models such as the Perspective API have been widely used to analyze toxicity in posts on social media platforms such as X (formerly Twitter), Reddit, and Facebook. In contrast, social networks, be they in groups or private messages, are becoming increasingly popular. This scenario highlights the need to evaluate user interactions in these environments, where the detection of toxicity is particularly challenging.

Given the inherent differences between these types of platform, there is an emerging need for classification models that are able to adapt to the unique characteristics of messaging environments, where messages are typically short, lack explicit context, and often depend heavily on the preceding conversation for meaning. These characteristics pose significant challenges for traditional classification models, which may struggle with the brevity, informal language, and rapid topic shifts common in messaging platforms, which are significantly different from tweets, for example, where all the context required

to understand it is contained within the tweet itself. Messaging platforms lack elements of traditional social media posts that enrich context, including hashtags, mentions, and especially lengthy discourse. Consequently, existing tools designed for longer, context-rich content may fail to accurately classify these messages.

To address these challenges, we propose an annotation and classification model that incorporates the contextual information surrounding each message. Specifically, when classifying a target message, our model considers the preceding messages, temporally, sent in the same chat, to provide a more accurate understanding of the content, operating as a sliding window. This approach mitigates the risk of misclassification due to the lack of individual information in short messages, ensuring a more reliable interpretation of the intended meaning. We utilized PySentimiento's [Pérez et al. 2024] model as a base, recognized in the field for its effectiveness in distinguishing hate speech and toxicity across multiple classes, and we extended its capabilities by incorporating our context-sensitive modifications. This enables our model to outperform existing ones in the nuanced environment of messaging platforms, where the context is often fragmented and implicit.

This is accomplished by first distinguishing between the aggregated context messages and the target message to be classified. These two blocks are then concatenated and fed into the transformer model. Although the context is processed through linear transformations, the embedding output is specifically trimmed to focus only on the target message. Consequently, the model classifies only the target message, despite the context being utilized in the transformation process.

In our paper, we describe the modifications made to the attention mechanisms in our model to effectively incorporate preceding messages while keeping the focus on classifying the target message. We also investigate the amount of context needed to enhance the model's performance. Our results show that, by carefully tuning the attention to relevant contextual information, the model achieves a significant improvement in classification accuracy on messaging platforms.

2. Related Work

In this section, we begin by reviewing previous work that explored message platforms, classification models, and what has been done in each area.

2.1. Messaging Platforms

In the last few years, the toxicity-related work on messaging platforms has increased significantly, especially when it comes to WhatsApp. [Melo et al. 2019, Melo et al. 2024] measures fake news and misinformation on the platform, either by developing a platform to monitor widely spread content or analyzing how the forwarding system works. In other ways, [Dahiya et al. 2020] tries to classify messages on WhatsApp into 6 groups based on sentiment analysis. Additionally, more recent studies on this network even started to study digital militias [Kansaon et al. 2024].

For other messaging platforms, many studies started to explore Discord and Telegram. [Wich et al. 2022] Explored abusive language classification on telegram, while other papers such as [Aliaksandr Herasimenka and Howard 2023] talk about conspiracy

theories and fake news on Telegram. Another interesting paper makes a comparison between WhatsApp, Telegram and Discord through the lens of Twitter, characterizing public groups on these platforms. [Hoseini et al. 2020].

2.2. Detecting Hate Speech

Text message classification has always been an extremely popular task. The paper of [Hutto and Gilbert 2014] was a pioneer, introducing VADER, a tool for text sentiment analysis. With the advent of transformers, this task became much easier and was dominated by the Perspective API [Lees et al. 2022], currently one of the most popular tools for text classification. Specifically for Portuguese, the PySentimiento model [Pérez et al. 2024] demonstrates strong performance metrics in hate speech classification, benefiting from training on toxic posts sourced from Twitter.

Recent advances in hate speech detection have focused on leveraging large-scale datasets and domain-specific models to improve accuracy and robustness. The work by [Balayn et al. 2021] presents a comprehensive review of existing resources and benchmark corpora for hate speech detection, highlighting the challenges of dataset variability and annotation inconsistency. Additionally, the use of specialized models, such as HateBERT [Caselli et al. 2020], trained specifically on abusive language corpora, has shown promising results in identifying nuanced forms of hate speech that often elude general-purpose models. These developments underscore the importance of tailored-made approaches to handle the complex and context-dependent nature of hate speech.

2.3. Research Gap

While there is a growing body of work on classification models for social media platforms, the unique challenges presented by messaging platforms remain underexplored. Traditional models for hate speech detection have largely focused on platforms like Twitter and Facebook, where messages are self-contained and rich in context. However, messaging platforms introduce distinct challenges due to the brevity and context-dependent nature of conversations, which require a fundamentally different approach of detection.

The fluid and rapid nature of conversations in these networks can obscure the boundaries of toxic content, making traditional detection methods less effective. In this direction, it is essential to distinguish our approach from the work done in sentiment analysis over the past decades. While much has been done in social media and sentiment analysis, our work addresses the novel challenge of incorporating surrounding context, in this case, the previous messages sent in the same chat, into the classification process. This approach is particularly unique as it applies a contextual deep learning strategy specifically made for detection in Portuguese instant messages, an area that has not been addressed in previous research.

In summary, our work aims to fill these gaps by developing a model that explicitly considers the unique characteristics of messaging platforms, enhancing the reliability of hate speech detection in these rapidly evolving communication networks.

3. Dataset

This section details the data collection process followed by a brief overview of the messages used in the training and evaluation processes.

3.1. Message Collection Process

Existing Portuguese toxic content datasets primarily focus on platforms such as Twitter and Reddit, where individual posts or comments often contain all the context required for classification tasks. However, these datasets lack coverage of messaging platforms, where the interpretation of a message frequently depends on the preceding conversational context.

To fill this gap, we collected messages sent in public Brazilian-Portuguese Discord groups, a prominent social platform serving a diverse array of communities. Similar to services like WhatsApp and Telegram, Discord is particularly known for being popular among teenagers who utilize it as a space for social interaction and virtual engagement. Moreover, the accessibility of Discord's API facilitated the data collection process, enabling the efficient compilation of our dataset.

According to Discords guidelines, public groups are those featured in Discovery [1], an official in-app feature to browse for new groups to join. For this work, we specifically focused on them, since it is an in-app certified public data source.

We collected the unique IDs of all groups available in Discord's Discovery feature. After that, using the official Discord API[2], we access any text messages shared on the groups since the creation of Discord, on May 13th 2015, up to March 1st 2024. To ensure the conversational context was captured, each message was collected along with the preceding 10 messages that were sent immediately before it, which we will refer to as a "set of messages". The data collection process followed established methodologies used in prior research specific to Discord's platform [Bento et al. 2024].

3.2. Data Annotation

The annotators were instructed to label each message with up to five of the available categories, derived from the base model PySentimiento: *Racism*, *Homophobia*, *Sexism*, *Body-Shaming*, *Ideology*. A message was deemed toxic if it fits into at least one of these categories, following the provided classification guidelines. If a message was not assigned any label, it was classified as non toxic.

We then randomly selected a sample of 2,200 Discord sets of messages, divided into two distinct groups. The first, containing 1,100 message sets, was selected by a weighted random selection process, prioritizing messages from larger groups to ensure a representative sample across different user populations. The second, also containing 1,100 message sets, was built using a term search algorithm designed to identify messages containing specific terms, frequently associated with toxic content. This targeted selection aimed to compile approximately 220 sets of messages for each of the 5 PySentimiento categories, deliberately including both hate speech and neutral sets to avoid term-associated bias in our dataset.

We recruited four undergraduate Computer Science students, aged between 19 and 23, consisting of one female and three male individuals, to serve as annotators. The set of messages was evenly divided among annotators, with each student initially receiving 650 messages to classify. After the initial classification, each student was assigned a different

[1]https://discord.com/guild-discovery

[2]https://discord.com/developers/applications

set, previously classified by another annotator, for reclassification. This ensured that every set of messages was classified twice.

We applied the Cohen's Kappa statistic [Cohen 1960], to measure inter-rater agreement. The average Kappa value was 0.89 between annotators, showing a high level of agreement among the raters. This high-level agreement may be attributed to the simplicity and brevity of messages shared in chatting platforms, which differ from the typical datasets labeled for other social media platforms.

4. Contextual BERT

This section describes the process of creating the contextual BERT-based model, which we will call Contextual BERT, for detecting hate speech messages with contextual information, the code is avaiable in [3]. To explain our model, it is essential to understand the central mechanism present in transformer models: the multi-head self-attention [Vaswani et al. 2023].

The self-attention mechanism can be interpreted as a global contextual transformation of embeddings, i.e., a dynamic transformation conditioned by the input data itself. Formally, this mechanism is described by the equation:

$$\text{Attention} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where the matrices Q (Query), K (Key), and V (Value) are different copies of the input embeddings, each undergoing distinct linear transformations using trainable weights. These transformations result in different representations that are used to compute *attention*.

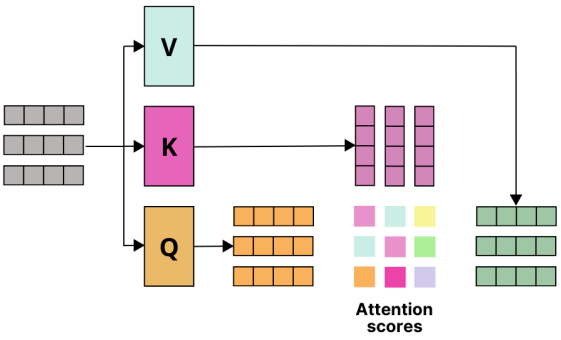

Figure 1. Attention mechanism.

The attention mechanism (Figure 1) explores the similarities between embeddings by using the dot product between matrices Q and K to generate a matrix of dimensions $L \times L$, where L is the length of the sequence of embeddings. Each value in this matrix represents the dot product between pairs of embeddings, capturing the contextual relationships between different parts of the sequence. After this matrix is created, its values are normalized via softmax, resulting in the attention or attention scores matrix.

[3]https://github.com/Buzelin2/Contextual-Bert

The attention matrix is then multiplied by the values matrix V - in the context of self-attention, the values matrix V is a copy of the input embeddings transformed by trainable weights 1. This process can be seen as a transformation applied to the embeddings, where the weights are directly constructed based on the context provided by the embeddings themselves. Technically, while a multi-layer perceptron (MLP) layer with weights W is static concerning the input embeddings, an attention layer with attention scores α is dynamic and conditioned by the input embeddings, as illustrated in Figure 2. Thus, the output embeddings, which pass through multiple transformer blocks, are explicitly transformed and constructed by operations considering the context itself. This property is fundamental to the development of our context-based message classifier.

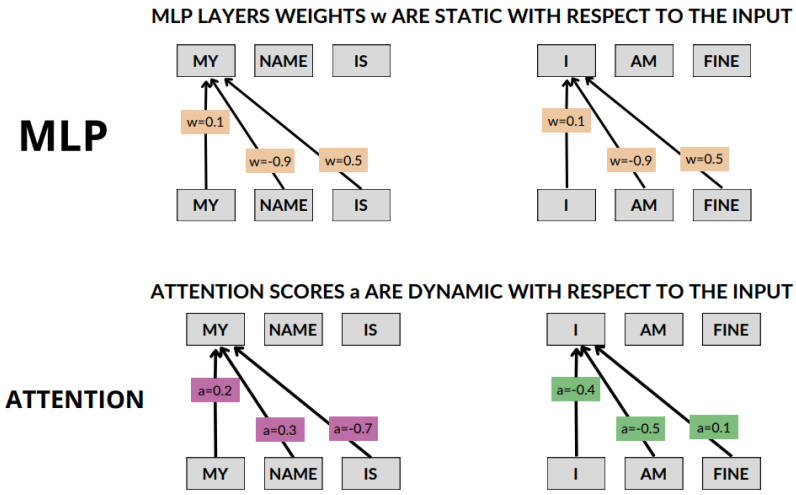

Figure 2. Illustration of how attention weights are conditioned by context.

To develop the context-based model, we build the input text by concatenating the n previous messages to the target message. Specifically, in a continuous social media chat, when classifying message t, our model also uses as input the preceding messages $t-1$, $t-2$, ..., $t-n$, following a sliding window approach of size n. Additionally, a special separation token is inserted between the context and the target message.

The resulting concatenated text is then used as input to the BERT-based transformer model. Following the traditional transformer approach, the concatenated text is tokenized – i.e., converted into discrete numbers, where each number corresponds to an index in the embeddings table. Thus, at the end of the tokenization process, the natural language text is projected into a sequence of numerical vectors, where each vector represents a token. Due to the permutation invariance property of the attention mechanism and transformer blocks, positional information is added to the embeddings through positional encoding vectors, forming the input to a series of transformer blocks.

As mentioned, after passing through multiple transformer blocks, the embeddings of the target message are transformed by the contextual relationships established with the embeddings of the context messages. Finally, since our goal is to classify the target message, we use only the output embeddings corresponding to the target message. We apply a global average pooling process to these embeddings, resulting in a single vector that passes through a final MLP layer to obtain the probabilities for each class.

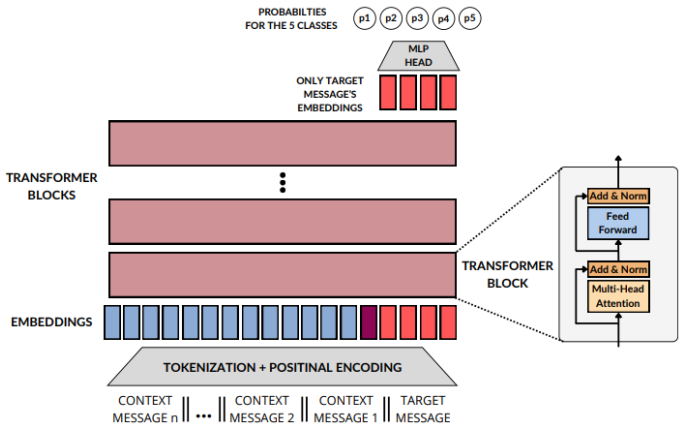

Figure 3. Complete architecture of the model.

Model Training: In this work, we initialized the models using the pre-trained weights from the PySentimiento model, a fine-tuned model for hate speech classification in Portuguese. PySentimiento leverages prior training for multi-class hate speech detection, built upon BERTimbau weights [da Costa et al. 2023], which were derived from self-supervised pre-training on 238 million Portuguese tweets, enabling it to learn rich representations of the language. This extensive pre-training enables PySentimiento to capture nuanced language patterns and contextual meanings specific to Portuguese. Subsequently, the base model was fine-tuned for hate speech classification across five categories – *Racism*, *Homophobia*, *Sexism*, *Body-Shaming*, and *Ideology* – using a dataset of 6,000 Portuguese tweets.

We further fine-tuned the PySentimiento model using a context-aware approach, applying it to our labeled Discord messages dataset. The fine-tuning process preserved the same five hate speech categories. The dataset was split into three non-overlapping subsets: training (80%), validation (10%), and test (10%). We employed the PyTorch library [Paszke et al. 2019] for model implementation, utilizing the AdamW optimizer [Loshchilov and Hutter 2017] with a learning rate of 5×10^{-6}, to minimize the Binary Cross Entropy loss(BCE-loss).

5. Experiments

Our experiments were divided into two phases. First, we show that the fine-tuned version of PySentimiento, even without any context, outperforms the original model. Next, we evaluate how much context is ideal for the model to learn to categorize toxic messages.

In all experiments, the model is run ten times with random validation, varying the test set to ensure statistical significance and minimize the influence of random chance. We then averaged the AUC-ROC scores for each model, as this metric is not dependent on specific thresholds and provides a reliable basis for comparing models with minimal noise. The model was trained on an NVIDIA RTX 4090 GPU, and early stopping was implemented based on validation performance to prevent overfitting.

5.1. Baseline Models

We first compared the performance of the original PySentimiento base model and the PySentimiento model fine-tuned without any contextual information. The AUC-ROC

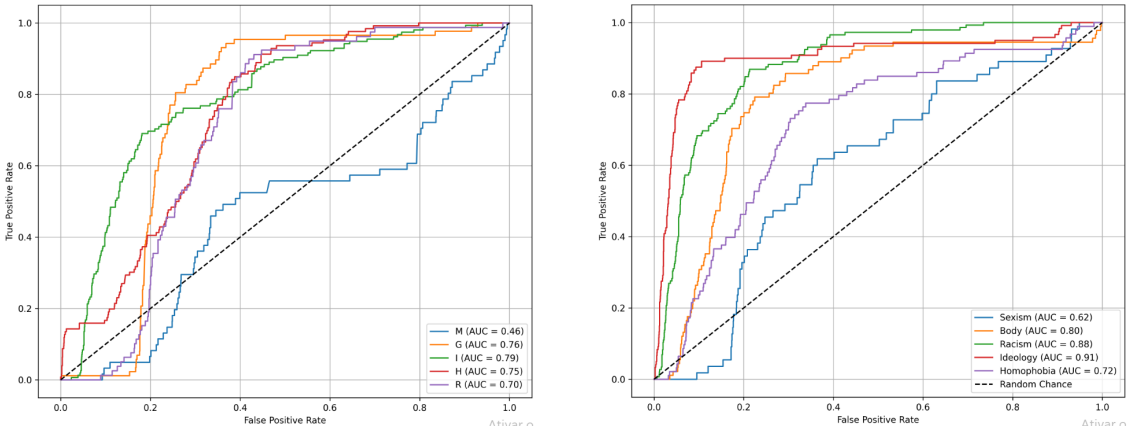

Figure 4. ROC curves of PySentimiento base model(left) and PySentimiento fine-tuned for each category(right).

Table 1. AUC scores of the prediction metrics for both models. Bold indicates the best-performing model.

Category	PySentimiento Base	Pysentimento Fine-tuned
Body shame	0.755	**0.803**
Homophobia	**0.745**	0.717
Ideology	0.791	**0.906**
Racism	0.704	**0.879**
Sexism	0.460	**0.617**
Average	0.691	**0.784**

curves and AUC scores, shown in Figure 4 and Table 1, respectively, provide insights into the models' effectiveness across the five categories.

While some improvements are evident, particularly in categories like *Racism* and *Ideology*, significant challenges remain. The AUC score for the *Sexism* category, in particular, indicates that the model struggles to differentiate sexist content from other types of messages. This issue highlights a fundamental limitation of the current models, suggesting that neither the PySentimiento base model nor the fine-tuned BERT model, without context, is sufficient for achieving robust classification across all categories.

Despite the improvements obtained in certain categories, others, such as homophobia, also underperform relative to expectations for a fine-tuned model. For example, Table 2 shows how the same message can have different classifications in different contexts. If you read context 1 followed by the first message, we have a *sexist* message. If we read after context 2, we have a neutral message. The same applies to the second message, which can represent *homophobia* in context 1 and be neutral in context 2. This underperformance raises the critical question of whether simply fine-tuning a model is adequate, or if a more sophisticated approach – such as incorporating context – is necessary to capture the nuances within these messages.

5.2. Contextual Model

The experiments reported in this section were developed to set the best number of context messages the model needs to effectively classify the target message. Although the model

Table 2. Examples of the same messages appearing in different contexts.

Context 1	Context 2	Message
"Vagabunda"	"pizza da boa mesmo"	**gostosa**
"ele é meio viado"	"Pq a sigla LGBT é: lesbica,"	**gay**

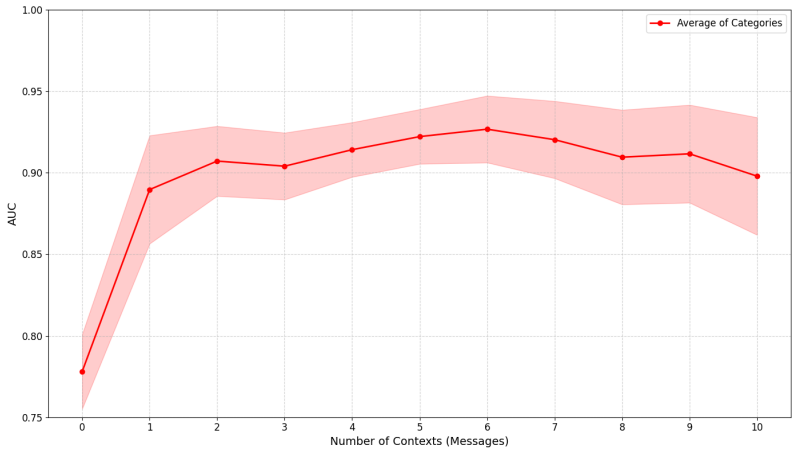

Figure 5. Average AUC scores of the fine-tuned model, representing the mean across all categories, with increasing context. Each configuration was trained and evaluated 10 times. The shaded area represents the variance.

focuses its embedding output solely on the target message, we hypothesized that including too many context messages might confuse the model, making it harder to accurately identify the relevant information.

Hence, we first trained the classifier on the target message with just one context message and gradually increased the number of context messages by one until reaching ten. We also compared it with the model fine-tuned without context, which relies solely on the target message.

Figure 5 shows the results of average AUC scores for the six categories for the model trained with different levels of contextual information, varying from no contextual to up to 10 context messages given to the model. Note that the model with context, regardless of using one or ten messages, always outperformed the model with no context, as indicated by the first point on the graph. On average, the model obtained its best performance with six context messages. However, when examining the categories independently, we observed variability, indicating that different categories might benefit from different amounts of context, as shown in Table 3.

Although there is variance, it is evident that most categories achieve their best performance using around 5 or 6 prior messages as context. Categories like Ideology, where the model performed well even without context, required only two context messages to reach maximum performance. In contrast, categories like *Sexism* and *Homophobia*, which initially performed poorly, needed more context to improve and stabilize their performance.

In general, apart from the AUC scores, the computational cost associated with processing context should be considered when choosing the size of the context that will

Table 3. AUC ROC scores for different context sizes, where "context size" denotes the number of preceding messages used in classifying the target message, followed by mean and standard deviation. Bold indicates the best context sizes for each column.

Context size	Body	Homophobia	Ideology	Racism	Sexism	Mean	SD
N/A	0.803	0.717	0.906	0.879	0.617	0.784	0.106
1	0.862	0.947	0.937	0.835	0.866	0.889	0.044
2	0.810	0.947	**0.958**	0.900	0.936	0.910	0.053
3	0.837	0.915	0.954	0.907	0.905	0.904	0.038
4	0.863	0.948	0.915	0.916	0.926	0.914	0.028
5	0.875	0.913	0.950	0.930	**0.940**	0.922	**0.026**
6	0.863	0.961	0.941	**0.973**	0.894	**0.926**	0.041
7	0.876	0.945	0.949	0.915	0.915	0.920	**0.026**
8	**0.878**	**0.963**	0.918	0.936	0.851	0.908	0.040
9	0.851	0.942	0.919	0.918	0.926	0.911	0.031
10	0.829	0.943	0.903	0.921	0.891	0.897	0.038

Table 4. AUC scores of the PySentimiento base-model, PySentimiento fine-tuned with no context and the best performing contextual model. Bold indicates the best performing model.

Metric	PySentimiento Base	PySentimiento Finetune	Context BERT
Mean AUC-ROC	0.691	0.784	**0.926**

be given to the model. There has to be a trade-off between AUC scores and computational time. While around six messages yielded the best performance in our scenario, choosing one or two contextual messages may also be a viable option, as training with six contextual messages takes nearly five times longer than training with two. This approach would significantly reduce the computational burden while still offering substantial performance gains when compared to using no context at all.

These experiments show that classifying a targeted message without context is challenging for the model, as short texts (which can have a single word) can be ambiguous and context-dependent. With increasing context, the model's performance progressively improves, reaching its optimal point, on average, at six contextual messages. Beyond this point, performance starts to decline. This decline could be attributed to several factors, the most plausible being that an excess of information might overwhelm the model, as the additional context might reference earlier messages irrelevant to the current classification task.

With these results, we demonstrate the substantial improvements achieved by incorporating context, as illustrated in Table 4, which shows that our model significantly outperformed both baseline models. This validates our hypothesis and enhances our understanding of the importance of context in message classification from messaging platforms. Messaging networks exhibit a unique conversational flow, and this new method of contextual classification could be key to accurately analyzing and understanding the dynamics of these media platforms.

6. Conclusion

This work introduces a contextual BERT model for toxicity detection in messaging platforms, addressing the growing need for more accurate content moderation as these platforms continue to gain popularity. By leveraging preceding messages, our model significantly enhances classification accuracy, demonstrating the critical importance of context in understanding and detecting toxic behavior in conversational environments.

Our experiments showed that incorporating up to six preceding messages in the input increases the model performance, significantly outperforming models that do not consider context.

In this manner, this research contributes to the broader field of natural language processing by demonstrating the value of contextual information in improving the accuracy of classification models in messaging environments.

Future work could focus on expanding the dataset to include messages from other widely used messaging platforms like WhatsApp and Telegram, thereby enhancing the model's broadness across different communication environments. Additionally, extending the model's capabilities to support multiple languages, particularly English, would significantly expand its applicability, making it valuable in multilingual settings. By exploring these avenues, the model could become more robust and versatile, effectively moderating content across a diverse array of messaging platforms and languages.

Acknowledgments

This work was partially funded by CNPq, CAPES, FAPEMIG, and IAIA - INCT on AI.

References

Aliaksandr Herasimenka, Jonathan Bright, A. K. and Howard, P. N. (2023). Misinformation and professional news on largely unmoderated platforms: the case of telegram. *Journal of Information Technology & Politics*, 20(2):198–212.

Balayn, A., Yang, J., Szlavik, Z., and Bozzon, A. (2021). Automatic identification of harmful, aggressive, abusive, and offensive language on the web: A survey of technical biases informed by psychology literature. *ACM Transactions on Social Computing (TSC)*, 4(3):1–56.

Bento, P., Buzelin, A., Aquino, Y., Carvalho, I., Dutenhefner, P., Dayrell, L., Santana, C., Estanislau, V., Pappa, G., Miranda, D., Almeida, V., and Jr, W. M. (2024). Impacto da pandemia na discussão sobre saúde mental: O caso do discord no brasil. In *Proceedings of the 30th Brazilian Symposium on Multimedia and the Web*, pages 179–187, Porto Alegre, RS, Brasil. SBC.

Caselli, T., Basile, V., Mitrović, J., and Granitzer, M. (2020). Hatebert: Retraining bert for abusive language detection in english. *arXiv preprint arXiv:2010.12472*.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

da Costa, P., Pavan, M., dos Santos, W., da Silva, S., and Paraboni, I. (2023). Bertabaporu: Assessing a genre-specific language model for portuguese nlp. In *Proceedings of the*

14th International Conference on Recent Advances in Natural Language Processing, page 217–223, Varna, Bulgaria. INCOMA Ltd.

Dahiya, S., Mohta, A., and Jain, A. (2020). Text classification based behavioural analysis of whatsapp chats. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pages 717–724.

Hoseini, M., Melo, P., Júnior, M., Benevenuto, F., Chandrasekaran, B., Feldmann, A., and Zannettou, S. (2020). Demystifying the messaging platforms' ecosystem through the lens of twitter. In *Proceedings of the ACM Internet Measurement Conference*, IMC '20, page 345–359, New York, NY, USA. Association for Computing Machinery.

Hutto, C. and Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225.

Kansaon, D., Melo, P. d. F., Zannettou, S., Feldmann, A., and Benevenuto, F. (2024). Strategies and attacks of digital militias in whatsapp political groups. *Proceedings of the International AAAI Conference on Web and Social Media*, 18(1):813–825.

Lees, A., Tran, V. Q., Tay, Y., Sorensen, J., Gupta, J., Metzler, D., and Vasserman, L. (2022). A new generation of perspective api: Efficient multilingual character-level transformers.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Melo, P., Messias, J., Resende, G., Garimella, K., Almeida, J., and Benevenuto, F. (2019). Whatsapp monitor: A fact-checking system for whatsapp. *Proceedings of the International AAAI Conference on Web and Social Media*, 13(01):676–677.

Melo, P. d. F., Hoseini, M., Zannettou, S., and Benevenuto, F. (2024). Don't break the chain: Measuring message forwarding on whatsapp. *Proceedings of the International AAAI Conference on Web and Social Media*, 18(1):1054–1067.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Pérez, J. M., Rajngewerc, M., Giudici, J. C., Furman, D. A., Luque, F., Alemany, L. A., and Martínez, M. V. (2024). pysentimiento: A python toolkit for opinion mining and social nlp tasks.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.

Wich, M., Gorniak, A., Eder, T., Bartmann, D., Çakici, B. E., and Groh, G. (2022). Introducing an abusive language classification framework for telegram to investigate the german hater community. *Proceedings of the International AAAI Conference on Web and Social Media*, 16(1):1133–1144.