IntelliLeksika at HOMO-MEX 2024: Detection of Homophobic Content in Spanish Lyrics with Machine Learning

Luis Ramos [*†], Carolina Palma-Preciado[*†], Olga Kolesnikova, Magdalena Saldana-Perez, Grigori Sidorov, and Moein Shahiki-Tash

Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC), Mexico City, Mexico

Abstract

Hate speech analysis in texts is important, and the development of models for its detection presents a challenge that demands the consideration of various approaches, particularly methods based on natural language processing. The identification of homophobic terms in songs, as proposed in Track 3 of the HOMO-Mex 2024 shared task, is of interest since these events create new knowledge in the area. This paper proposes the utilization of both traditional machine learning and deep learning algorithms to compare their performance. Among the submitted runs, the team achieved the best results using a Decision Tree with the NNLM embedding, attaining a macro F1 score of 0.482, and with a Bert-like model (BETO), which obtained a macro F1 score of 0.486. This represents a non-significant difference, indicating that there is no substantial distinction in the behavior of the models for this problem, and that further investigation is needed since the overall scores were low.

Keywords

Hate Speech Detection, LGBT+ Phobia, Lyrics, BERT, Deep Learning, Natural Language Processing

1. Introduction

The study of texts on hate speech is a topic of great interest that has been approached from different perspectives, as this area can encompass many things [13]. In this case, Homo-MEX 24 focuses on detecting hate speech targeted at the LGBT+ community in texts written in Spanish, specifically from Mexico [2].

[*] Corresponding author.

[†] These authors contributed equally.

✉ lramos2020@cic.ipn.mx (L. Ramos); cpalmap2020@cic.ipn.mx (C. Palma-Preciado); kolesnikova@cic.ipn.mx (O. Kolesnikova); amagdasaldana@cic.ipn.mx (M. Saldana-Perez); sidorov@cic.ipn.mx (G. Sidorov); mshahikit2022@cic.ipn.mx (M. Shahiki-Tash)

iD - 0009-0008-5586-6668 (L. Ramos); 0000-0003-3253-4464 (C. Palma-Preciado); 000-0002-1307-1647 (O. Kolesnikova); 0000-0002-2475-1621 (M. Saldana-Perez); 0000–0003–3901–3522 (G. Sidorov); 0009-0003-5767-0566 (M. Shahiki-Tash)

Text analysis, achieved through natural language processing (NLP), aims to identify hateful sentences. Now, it is focused on homophobia in social content. This seeks to serve as a solution for moderating content with the goal of creating a safe environment for users.

Hate speech can occur in different ways, either directly or indirectly, primarily through texts containing explicit expressions such as insults, profanity, scorn, and derogatory words or, less directly, through insinuations.

In track 3, the study of songs with homophobic lyrics is proposed. This is a compilation of songs from different genres, and the creators of the HOMO-MEX 2024 workshop suggest using it to automatically detect this type of discrimination [16].

As classification tasks have become more complex, new models based on transformers and neural network language models (NNLM), have been developed that, in most cases, perform better than traditional machine learning algorithms such as support vector machine, logistic regression, and decision tree, among others.

The models based on transformers use a self-attention mechanism to evaluate the importance of words. They also use positional encoding, which helps identify information

about words by managing their positioning. Thus, they provide context by allowing the identification of a word's position in a sequence [18].

While model selection is essential, another equally important aspect to consider in NLP is data preprocessing. This helps clean the data or texts of uninformative information, thereby maintaining the most significant features. When using a representation method such as Term Frequency–Inverse Document Frequency (TF-IDF) or embeddings, vector representations will be obtained and used to train the models.

This work mentions the process carried out in the participation of track 3 for HOMO-MEX 2024, where both approaches, machine learning algorithms, and deep learning, are utilized, considering different levels of preprocessing.

2. Literature Review

The presence of offensive language in music lyrics has become a considerable concern in today's society, promoting vast research into its detection and impact. Offensive language, hate speech, and dismissive terms, can influence listeners' perceptiveness and behaviors, often perpetuating negative stereotypes and contributing to societal issues.

Given the problems described above, offensive language detection models have been developed in lyrics. These methods range from traditional to deep learning algorithms capable of understanding context and subtleties in language.

This review aims to provide a comprehensive overview of the existing methodologies for detecting offensive language in lyrics and the detection of offensive language or hate speech against LGBTQ+ community.

2.1. Explicit Language Detection in Song Lyrics

In [5] the authors compare different methods to detect explicit or inappropriate language in Korean song lyrics. Different preprocessing methods were used, such as dictionaries, Bag of Words (BOW), and TF-IDF.

In addition to the use of Convolutional Neural Networks (CNN) and transformers. The proposed method found that complex models do not necessarily surpass simple methods but that combining different techniques can offer a more robust approach.

In conclusion, detecting explicit or inappropriate language is a challenging task due to its inherent subjectivity, which is further complicated by cultural influences.

On the other hand, in [4], the Random Forest algorithm was utilized to classify songs containing explicit language. For this purpose, TF-IDF was used as a method of vectorization over preprocessed text.

This method demonstrates that combining RF with TF-IDF is effective in classifying songs with explicit language. Additionally, it was observed that the Hip-Hop music genre tends to be the most frequent producer of songs with explicit content.

2.2. Hate Speech Detection Against LGBT+ Community

There are no applications focused on hate speech or explicit language detection against LGBT+ community in song lyrics, but there are some proposals for hate speech detection.

In [8], a model for detecting homophobia and transphobia in social media comments was developed for languages with limited resources, specifically in Malayalam and Hindi, using data obtained from YouTube. Traditional methods such as Naive Bayes and Random Forest, as well as transformer models including BERT, RoBERTa and XLM-RoBERTa, were employed. The features utilized comprised TF-IDF, fastText, and BERT embeddings.

Furthermore, the transfer of knowledge between different languages was explored through cross-learning, evaluating the ability of models trained in one language to predict homophobic and transphobic content in another language. This approach demonstrated potential and feasibility for detecting discriminatory content.

Another approach to detecting hate speech targeting the Spanish-speaking LGBT+ population in Mexico using BERT-based models for analyzing tweets was investigated in [13]. The authors emphasize the importance of preprocessing, since the text must be cleaned of lexical noise and apply lemmatization to improve the effectiveness of self-attention mechanisms in transformers.

Moreover, the researchers identify two key factors that influence the results: the lack of preprocessing before tokenization and the quality of the labeling of the dataset. To conclude, the authors suggest improvements in dataset labeling and classification, as well as exploring new approaches to improve detection and mitigation of LGBT+ phobia in online spaces.

Finally, in [6], models for hate speech detection in tweets were developed using traditional machine learning models and BERT-based transformers for Spanish (BERT and RoBERTa) and multilingual (mDeBERTaV3). As a conclusion, both methods proved to be effective, emphasizing the importance of developing automated tools to help protect vulnerable communities.

3. Methodology

The process of classifying homophobic songs includes analyzing the data set, preprocessing the data, and training and testing models. In Figure 1, each stage describes the steps taken into account. However, it should be noted that combinations were made between data preprocessing, representation, and the models, as the goal was to find the best combination for optimal performance. While internal evaluation of the models' performance was conducted during training, the final test score was directly calculated by the organizers of the shared task HOMO-Mex 2024.

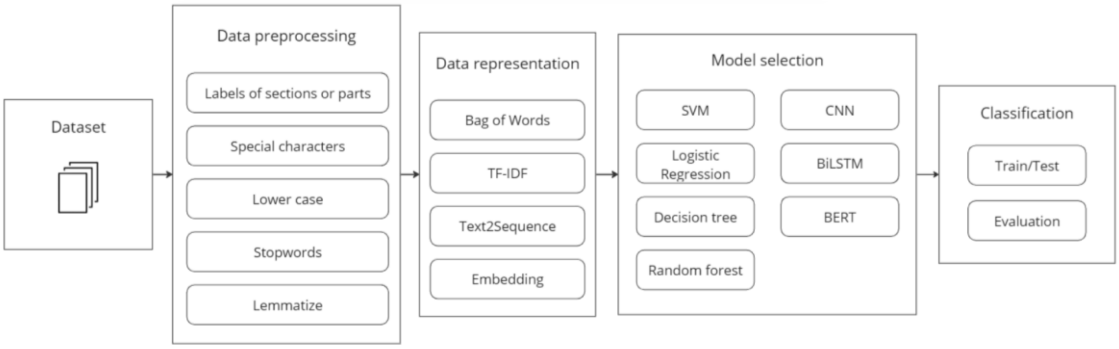

Figure 1: Methodology for hate speech detection in lyrics using traditional machine learning algorithms and transformer models.

3.1. Dataset

The composition of a dataset is important because knowing how it is structured and how many samples there are for each class label helps identify its strengths and weaknesses and thus address these in the models to be trained.

The dataset of songs for task 3 is for binary classification, as it has two classes: the first labeled "P" which refers to LGBT+ phobic songs, and the second labeled "NP" which denotes songs unrelated to LGBT+ phobia [8].

For the training phase, a sub dataset of 984 samples was provided, with 945 labeled as "NP" and 39 as "P". This indicates that the dataset is highly imbalanced, with an imbalance ratio of $\frac{945}{39} = 24.23$ (see Figure 2).

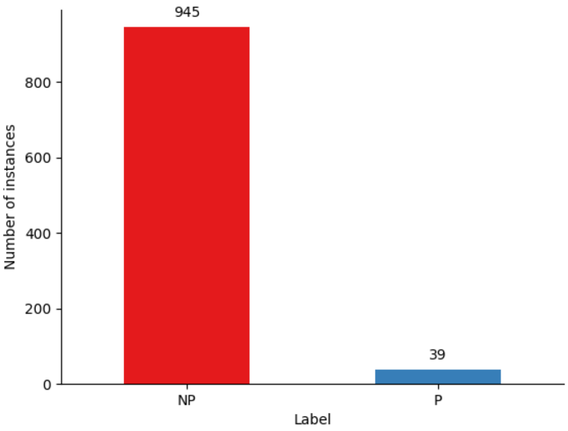

Figure 2: Training dataset distribution between class labels.

On the other hand, the test subset contains only 246 samples, which is 20% of all the data, contemplating both training and test since there is no information about the proportion that belongs to the positive class, i.e., the class label “P” that we want to predict correctly.

3.2. Data preprocessing

Data cleaning is an integral part of preprocessing because it helps deleting the less representative parts of the text. In this case, five steps were considered apart from tokenization, where each sentence is separated into tokens (words). It is worth noting that different levels of processing were used during the training phase; these combinations will be explained in the classification phase.

Labels of sections or parts: Any tags referring to the song's phase were removed, such as tags indicating the intro [Intro], verse [Verso 1], chorus or refrain [Coro], bridge [Puente], interlude [Interludio], outro [Outro], among others.

Special characters: Characters like punctuation, exclamation marks, and others like @, %, $, were considered less relevant and removed from the text. Although accents may not be regarded as unique characters in Spanish, they were also changed so that accented vowels were represented by their unaccented counterparts, i.e., á was changed to a.

Lowercase: Converting the text to lowercase is important in reducing the number of unique words when creating the vector space using text representation methods like Bag of Words or TF-IDF. This ensures that words like "song" and "Song" are considered the same instead of two different words.

Stopwords: These are repetitive words in the language that contribute little to the context. Removing them does not make the text lose meaning, so they are commonly eliminated; these include articles, prepositions, pronouns, and auxiliary verbs.

Lemmatize: It's the process of converting a word to its root form, known as a lemma. Lemmatization considers the meaning and grammar of the word, using dictionaries to accurately obtain them [9].

3.3. Data representation

Data representation is an essential step since algorithms cannot understand text; therefore, it is necessary to create a vector space that represents what is written. In NLP, different methods exist, from the simplest form using word or token frequency to more robust methods such as embeddings.

Bag of Words: Text is characterized as a bag of words, where the order of the words is not maintained, and only the frequency of each token is considered.

TF-IDF: This type of representations addresses the issue of word bags by normalizing term counts so that repeated words do not receive the most weight, is one of the most used methods for its simplicity and good performance [17].

For TF-IDF, not only word frequency was considered, but n-grams of both words and characters were also used to assess their impact on the model behavior.

Text2Sequence: Text is converted into integer sequences. This process captures the word order by preserving its sequence in the integer representation. Additionally, padding is applied to ensure all samples have the same length.

Embedding: They capture the meaning of text using pre-trained models to generate their vector encodings. These models leverage pre-existing sentences to handle the values. The dimensionality of embeddings can vary depending on the method used.

In this work, we tested three different embeddings: the one from the BERT model and the NNLM embedding with 128 in dimension. Both were trained on Spanish datasets and can be directly applied to text using the TensorFlow Hub library [12, 18].

The last one was AffectiveSpace, it was built for sentiment analysis tasks, these embeddings are associated with a dictionary of 49,825 elements, which was created with combinations of words from one to seven and the size of the embedding is 100 [12]. For each lyric, it was divided each lyric in (1-7) N-Grams and it was obtained the embedding that matched with each N-Gram, and it was computed the mean to obtain an embedding to represent each song in the vector space.

3.4. Model selection

A comprehensive approach was employed for model selection, considering machine learning and deep learning algorithms as they have become the most used techniques in artificial intelligence [14], Besides, the algorithms were chosen because of their specific characteristics and adaptability to vector space and offers multiple setups, such as kernels in Support Vector Machine or trees' number Random Forest. Nevertheless, more details about the chosen algorithms are given in the results section. Within machine learning, algorithms were evaluated based on their approach:

- Statistical methods: Logistic regression
- Decision tree methods: Decision tree and random forest
- Support vector machine (SVM)
- Neural network methods: Multilayer Perceptron (MLP)

Deep learning approaches utilizing more complex neural network architectures were also explored. Initial testing included Convolutional Neural Networks (CNNs) and Bidirectional Long-Short-Term Memory (BiLSTM) networks. Finally, transformer-based models, including a Spanish variant of the BERT model known as BETO [1], were evaluated.

3.5. Classification

During the classification stage, all testing was conducted using the training set. Different combinations of the aforementioned techniques were experimented with to optimize the model's performance. This included evaluating no preprocessing, lightweight and complete preprocessing of the data and its vector representation. Lightweight preprocessing includes only the elimination of the song label and complete include every process mentioned in data preprocessing section.

Additionally, an internal analysis of the models' performance was conducted to identify their weaknesses and improve them in this stage. The final goal was to obtain the best-trained model for predicting the labels of the test set.

The Sci-kit-learn, Keras, and TensorFlow libraries, along with libraries like Transformers, were employed for this stage. The testing was conducted in the Google Colab environment, where the entire development process occurred.

4. Results

This section presents the preliminary results obtained during training, the analysis of the dataset, and the considerations for selecting the best models. These models were then used to predict the labels of the test set provided and were uploaded to the Codabench platform for evaluation.

To assess the model's ability to learn and capture patterns in the dataset, using the same dataset for training and testing can be helpful in identifying difficult patterns. This is because if the model struggles to correctly classify certain samples even when it has seen them during training, it indicates that these are challenging for the model to learn.

After conducting an evaluation, MLP was achieved a perfect score and the same result was obtained (1.0 in precision and 0.99 for recall and f1-score over macro avg) in Random Forest, SVM, Logistic Regression, and Decision Tree algorithms, only one instance misclassified. The confusion matrix in Figure 3 shows that only one pattern is identified as a false negative, being the sample with the index **803**.

	Patterns classified as Positive	Patterns classified as Negative
Positive class	38	1
Negative class	0	945

Figure 3: Confusion matrix of the training dataset experimentation.

Due to the song's length, only the first verse is showed in the following extract from the song that refers to the sample 803:

Original version:

> *[Verso: Raymix] Oye mujer, Lo que has provocado en mí, No tengo explicación, Me hundo en la emoción, Qué sucede, Oye mujer, Tú me has conquistado y yo, Ni como decir lo que yo haría por ti, Yo te amo.*

Translate version:

> [Verse: Raymix] Hey woman, What you have provoked in me, I have no explanation, I sink into emotion, What's happening, Hey woman, You have conquered me and I, I can't even say what I would do for you, I love you.

This review did not reveal any reference to LGBT+ phobic language, although in this dataset it is labeled as such. This indicates that more information should be given about how organizers produced their dataset and what labeling guidelines were used for it. The tag assigned has nothing to do with this because there is no homophobic content in the song, which affects model performance. On the other hand, further analysis was not done on labelling of this data set because it serves a different purpose for this task.

The team submitted four runs for evaluation for the official results in shared task 3. These results are shown in Table 1, where the performance measures were calculated using Macro Score. While all the algorithms mentioned in the methodology were tested, only the runs that classified some patterns as “P” in test phase were submitted. Most of these algorithms detected all patterns as the class “NP” and were therefore not considered.

Additionally, the limit on run submission influenced the model selection process. The team had to decide which models to submitted based on observations and the criterion of label results from patterns classified as "P". As a result, the models that classified the most patterns as "P" were submitted, but this does not guarantee their correctness.

As shown by Table 1, a model based on BERT (BETO) had the best results from the send runs, with a F1-score of 48.64%, precision of 47.94%, and a recall of 49.36%. Only three patterns were classified as “P”, indices **204, 231,** and **237**. This process underwent minimal preprocessing, removing only the song section labels and repeated spaces and using the model's embedding. The pre-trained model was obtained from the "dccuchile/bert-base-spanish-wwm-uncased" model on Hugging Face [3]. The configuration used for this experiment included a learning rate of 2e-5, a batch size of 180, and was run for 3 epochs.

Table 1
Model Performance on test data, team submission evaluation.

Model	F1-Score	Precision	Recall
BETO[2]	0.4864	0.4794	0.4936
Decision Tree[3]	0.4821	0.4790	0.4851
Decision Tree[3]	0.4788	0.4788	0.4788
Decision Tree[1]	0.4755	0.4785	0.4725

[1] No data preprocessing
[2] Lightweight data preprocessing
[3] Complete data preprocessing

On the other hand, decision trees were tested with different combinations: one using full preprocessing with the NNLM embedding, another without preprocessing with the same embedding. For both configurations, the hyperparameters were set to the default values provided by scikit-learn. However, an exception was made for the criterion parameter, which was explicitly set to 'entropy'.

Also, another configuration with different features was used for a Decision Tree. The feature vector was built using two vectorization methods with the CountVectorizer function from the scikit-learn library. The best configuration was (2-4) N-Grams of words with max_features=55, concatenated with (3-15) N-Grams of characters with max_features=770 and analyzer='char_wb'. In this instance, the same criterion was used, but the splitter value was changed to 'random'.

For both cases in the Decision Tree models, applying comprehensive data preprocessing resulted in better outcomes than without preprocessing. No lightweight processing was performed using these algorithms. However, for the BETO, lightweight processing was applied, as transformer models do not require extensive text processing due to the robustness of the used model.

5. Discussion

The best results were obtained using a lightweight and full preprocessing step before training a model. BETO achieved the highest F1-score among others with 0.4864, but the performance difference between BETO and other models was not statistically significant, including a comparison with the highest F1-Score achieved in the task 3 overview [8].

The error analysis highlighted concerns regarding the quality of the data and its labeling, as these directly influence the performance of the models and the conclusiveness of the evaluation, suggesting that these factors impact the performance of the models and the reliability of the evaluation outcomes.

6. Conclusions

To identify homophobic lyrics, various classification and text representation models were tried in this proposal. During the procedure, it was noticed that the quality of dataset along with its labeling may impact on how well these models work which stresses their importance in such tasks.

Overall, BETO showed the best performance, although the difference in F1-score was minimal, with variations in hundredths and thousandths considering macro avg precision and recall results. This improvement, while noticeable for the competition, is not statistically significant respect to the other models and could be related to other factors such as model hyperparameters or data preprocessing.

The low performance of the algorithms in this task suggests that the complexity lies in data quality, semantic relationships, and the cultural context of linguistic expressions. Despite the results obtained, this proposal has certain limitations, such as the unknown distribution of musical genres, song popularity, among others. Therefore, considering additional details about the songs may lead the creation of more complex methods and the performance of the algorithms may improve.

This study demonstrates that classifying homophobic lyrics is a challenging task, and the achieved performance highlights the necessity to collect more data and explore alternative methodologies. Researchers working on hate speech or explicit language detection should prioritize the origin, quality, and quantity of their data. Additionally, performing various analyses and employing diverse methods is recommended to enhance results in such tasks.

Acknowledgments

The work was done with partial support from the Mexican Government through the grant A1-S-47854 of CONACYT, Mexico, grants 20241816, 20241819, and 20240951 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de

Supercómputo of the INAOE, Mexico and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

References

[1] Cañete, J., Chaperon, G., Fuentes, R., Ho, J., Kang, H., & Pérez, J. (2023). Spanish pre-trained BERT model and evaluation data. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2308.02976

[2] Chiruzzo, Luis, Jiménez-Zafra, Salud María, & Rangel, Francisco. (2024). *Overview of IberLEF 2024: Natural Language Processing Challenges for Spanish and other Iberian Languages*. In Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2024), co-located with the 40th Conference of the Spanish Society for Natural Language Processing (SEPLN 2024), CEUR-WS.org.

[3] *dccuchile/bert-base-spanish-wwm-uncased. Hugging Face*. (n.d.). https://huggingface.co/dccuchile/bert-base-spanish-wwm-uncased

[4] Dwiyani, L. K. D., Suarjaya, I. M. a. D., & Rusjayanthi, N. K. D. (2023). Classification of explicit songs based on lyrics using random Forest algorithm. *Journal of Information Systems and Informatics*, *5*(2), 550–567. https://doi.org/10.51519/journalisi.v5i2.491

[5] Fell, M., Cabrio, E., Corazza, M., & Gandon, F. (2019, September 2). *Comparing automated methods to detect explicit content in song lyrics*. https://hal.science/hal-02281137

[6] Fernández Rosauro, C., & Cuadros, M. (2023, September). Hate speech detection against the Mexican Spanish LGBTQ+ community using BERT-based transformers. *CEUR Workshop Proceedings*. https://ceur-ws.org/Vol-3496/homomex-paper7.pdf

[7] Gemma Bel-Enguix, Helena Gomez-Adorno, Gerardo Sierra, Juan Vásquez, Scott-Thomas Andersen, & Sergio Ojeda-Trueba (2023). Overview of HOMO-MEX at Iberlef 2023: Hate speech detection in Online Messages directed toOwards the MEXican Spanish speaking LGBTQ+ population. *Natural Language Processing*, 71. 1989-7553.

[8] Kumaresan, P. K., Ponnusamy, R., Priyadharshini, R., Buitelaar, P., & Chakravarthi, B. R. (2023). Homophobia and transphobia detection for low-resourced languages in social media comments. *Natural Language Processing Journal*, *5*, 100041. https://doi.org/10.1016/j.nlp.2023.100041

[9] Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. https://doi.org/10.1017/cbo9780511809071

[10] *nnlm*. (n.d.). Kaggle. https://www.kaggle.com/models/google/nnlm/tensorFlow1/de-dim128-with-normalization/1?tfhub-redirect=true

[11] *SenticNet*. (n.d.). https://sentic.net/downloads/

[12] Paz, M. A., Montero-Díaz, J., & Moreno-Delgado, A. (2020). Hate Speech: A Systematized review. *SAGE Open*, *10*(4), 215824402097302. https://doi.org/10.1177/2158244020973022

[13] Shahiki-Tash, M., Armenta-Segura, J., Ahani, Z., Kolesnikova, O., Sidorov, G., & Gelbukh, A. (2023, September). LIDOMA at HOMO-MEX2023@IberLEF: Hate speech detection towards the Mexican Spanish-Speaking LGBT+ population. The importance of

preprocessing before using BERT-Based models. *CEUR Workshop Proceedings*. https://ceur-ws.org/Vol-3496/homomex-paper1.pdf

[14] Sharifani, Koosha & Amini, Mahyar. (2023). Machine Learning and Deep Learning: A Review of Methods and Applications. 10. 3897-3904.

[15] *universal-sentence-encoder*. (n.d.). Kaggle. https://www.kaggle.com/models/google/universal-sentence-encoder/tensorFlow2/multilingual/2?tfhub-redirect=true

[16] Helena Gómez-Adorno, Gemma Bel-Enguix, Hiram Calvo, Juan Vásquez, Scott Thomas Andersen, Sergio Ojeda-Trueba, Tania Alcántara, Miguel Soto & Cesar Macias. (2024). Overview of HOMO-MEX at Iberlef 2024: Hate Speech Detection Towards the Mexican Spanish Speaking LGBT+ Population. *Natural Language Processing*, *73*. 1989-7553.

[17] Vajjala, S., Majumder, B., Surana, H., & Gupta, A. (2020). Practical natural language processing: A Pragmatic Approach to Processing and Analyzing Language Data. O' Reilly Media.

[18] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1706.03762