ITC's Participation at DIMEMEX: Data Augmentation using Generative AI for Better Detection of Hate Speech in Mexican Memes*

Ramón Zatarain Cabada*1*, María Lucía Barrón Estrada*1*, Víctor Manuel Bátiz Beltrán*1*, Aldair González Robles*1* and Néstor Leyva López*1*

1 Tecnológico Nacional de México: Instituto Tecnológico de Culiacán, Juan de Dios Bátiz 310 Pte, Guadalupe, 80220 Culiacán Rosales, Sinaloa, México

Abstract

This article presents the work done for the detection of inappropriate content, hate speech, or neither of them in Mexican memes within the DIMEMEX competition as part of IberLEF 2025. In contemporary society, the employment of memes as a medium for conveying ideas or messages has become a prevalent practice across a wide range of social networks utilized by users worldwide. The automatic detection of inappropriate content or hate speech has become a subject of significant interest for the scientific community. In this study, we propose an approach that utilizes paraphrasing to augment data, employing Transformers-based models for the classification of messages within memes. The proposal that is the subject of this study is a BETO-based model. This model obtained an f1-score of 0.52, which placed it in fourth place in the final phase for task 1. It is concluded that, despite the encouraging results, the task is quite complex. This conclusion is based on the analysis of the evaluated metrics, which revealed that all of the results fell below 0.60.

Keywords

hate speech, inappropriate content, data augmentation, transformers

1. Introduction

This research demonstrates the approach utilized and the results obtained by the ITC team in the DIMEMEX 2025 competition [1], which was organized as part of IberLEF 2025 [2]. The competition comprised three tasks with the objective of identifying the presence of hate speech, inappropriate content, or neither of them in Mexican memes.

The contemporary significance of social networks in modern life is indisputable; they have evolved into an immediate and universal medium of communication. The present study analyzes the information transmitted through text messages, voice, and images or combinations of these. The case study of this research focuses on memes, which are combinations of images and text that seek to express an idea or message. Despite their seemingly innocuous nature, memes have been known to contain hate speech or inappropriate messages.

In this study, the approach employed by the ITC team to address Task 1 is presented. Task 1 entailed the classification of the message contained in Mexican memes into one of the following categories: inappropriate content, hate speech, or neither of them. The methodology employed in the present study is delineated, and the ensuing results are hereby presented.

2. Related work

The task of recognizing hate speech presented in memes requires understanding of the context in which they are used, as well as the visual content of the memes. Therefore, in works such as the one presented in [3], they address the identification of language in writings and hate detection using machine learning (ML) and deep learning (DL). For which they explore various approaches such as Support Vector Machines (SVM) and Random Forest. Different labeled datasets from the CHiPSAL 2025 challenge [4] were used for language classification and hate recognition in Asian languages. In the work, it was obtained that the hybrid Convolutional Neural Networks (CNN) model with BiLSTM achieved the best accuracy (F1-score of 0.9941) in writing identification, while MuRIL-BERT obtained an F1-score of 0.6832 in hate speech detection. The authors conclude that the approaches employed outperform traditional approaches in these areas, identifying limitations such as data imbalance and the need for more refined models for this specific task. Similarly, the work described in [5] addresses the analysis of sentiment in memes in social networks, using textual and visual information. In this work, a Bi-modal model was proposed to take full advantage of meme data to improve classification. The proposed approach uses feature extraction with the VGG-19 model and textual analysis with a GRU (Gate Recurrent Unit) network. This was integrated into a CNN model to classify sentiments by testing various text and image models to evaluate the classification efficiency. For the tests they used the Memotion dataset, which is composed of 6,992 memes labeled as positive, negative and neutral. Finally, they obtained that the Bi-modal model (GRU + VGG19 with CNN) achieved an accuracy of 60% and an F1-score of 0.3904. Evidencing that the SVM analysis achieved better accuracy with up to 88% and only image analysis with a performance of 58.56%. This paper concludes that DL models improve the interpretation of feelings in memes, reporting the same limitation as the previous work, where it is explained that unbalanced data make it difficult to classify emotions in memes. Similarly, the research in [6] addressed the classification of memes with hate content using ML techniques. For this, they proposed two approaches: image-to-text conversion and text-to-image-to-vector conversion. For this, the dataset of the Facebook Hateful Memes Challenge [7] competition was used in combination with a new dataset of memes randomly extracted from the internet. Models such as Long-Short Term Memory (LSTM), SBERT; Xception and CNN were used, focusing on multimodal strategies to analyze the relationship between image and text, implementing image segmentation methods and advanced models for feature extraction and classification. After the tests they found that what they call "generic models" without specific knowledge about hate speech can learn the definition of it. Concluding that multimodal models improve accuracy in detecting offensive content in memes, but unbalanced data and subjectivity in the definition of "hate speech" remain important challenges to solve.

Regarding specific classification tasks of the type of hate speech (racism, sexism, classism, etc.), there are also works that have addressed these issues, as is the case of the work in [8]. In this paper, the authors proposed a system for detecting racism in social networks using ML and DL techniques. For this purpose, a model based on LSTM was developed to analyze and classify racist content, with the objective of reducing detection time and improving accuracy in the identification of discriminatory speech. BERT and Lemmatizer were used to extract the features, and a web application was built in Flask for implementation, allowing users to enter content and the system to analyze and classify them as racist or non-racist. They found that their proposed LSTM model had better accuracy than other tested networks (such as BERT, GNN and RNN), identifying data imbalance in the ensemble as one of the major challenges for the future. On a similar topic, Nguyen et al. [9] propose the detection of sentiment and hate speech in social networks using multimodal ML models, integrating text and image to improve classification, focusing on posts related to racism and sexual preference discrimination. To do this, approximately 56 million tweets and a little more than 3 million posts on Meta platforms were collected. To process this data, models such as BERT for text and VGG-16 for images were used, in addition to some multimodal models such as

CLIP and VisualBERT. They report that CLIP obtained the best accuracy, reaching up to 96% in hate speech, while VisualBERT excelled in sentiment analysis. The authors emphasize that the combination of multimodal models outperforms unimodal models, commenting that adding text and image provides greater context and accuracy. Like the other authors, they report that data imbalance is a limiting factor in improving detection. While, in problems related only to misogyny, in [10] the authors propose a multimodal approach to detect misogynistic memes, for which they use XLM-RoBERTa to extract textual features and VisionTransformer for visual features. Once they acquired the features, they concatenated the textual and visual attributes to build a multimodal representation. This was tested with ML algorithms such as KNN and SVM, and DL algorithms such as LSTM and GRU. At the end of their study, they report that the GRU model achieves an F1-score of 0.88, indicating that it generalizes best in language comprehension. They conclude that multimodal models outperform unimodal models, especially in the classification of memes with discriminative content, highlighting the problem of data imbalance.

In a similar vein, though detecting hate speech using Large Language Models (LLM), we find works such as the one in [11], which proposes to test the performance of the LLM GPT-2 for detecting hate speech. For this, the authors explored two approaches: Fine-Tuning of the model with a specific hate speech dataset and the application of Few-Shot-Learning (FSL) techniques, seeking to leverage the language capabilities of GPT-2. The results showed that the GPT-2 Fine-Tuning obtained the best performance with an F1-score of 0.6164 with test data, although an over-fit is observed compared to the validation F1-score which has an F1-score 0.7364. On the other hand, in the FSL approach with 0 shots was the most effective, achieving an F1-score of 0.58. With this, they conclude that providing more labeled data to a model is more beneficial than just instructions and examples for GPT-2 to learn hate speech features. A work with a different approach to LLMs is presented in [12], which describes a framework for evaluating hate speech detectors against LLM-generated content. This study seeks to understand the effectiveness of existing models in the face of such content and to reveal the potential of LLM-driven hate campaigns. For this purpose, the authors built a dataset of 7,838 hate speech samples generated by 6 different LLMs for 34 identity groups and subsequently evaluated the effectiveness of 8 hate speech detectors on this dataset. After testing, they found that hate speech detectors showed a decrease in performance when confronted with content generated by newer versions of LLMs. They also reveal that LLMs possess significant potential to drive hate campaigns, representing a new threat, underlining the need to develop more robust and adaptive detectors. Finally, the work in [13] was reviewed, which proposes the detection of offensive memes for online content moderation, especially in a culturally diverse context. For this, they propose a pipeline that integrates Multimodal Large Language Models (VLM) and Fine-Tuning techniques with a dataset annotated by GPT-4V, seeking to handle languages with few resources and local social biases. The dataset created consisted of 112,000 memes, including Optical Character Recognition (OCR) to extract text, translation for low-resource languages, and a VLM of 7 billion parameters for final classification. The proposed solutions achieved an accuracy of 80.62%, demonstrating the effectiveness of the VLMs fitted with the created dataset. The authors conclude that their pipeline can significantly help human moderators, especially in specific cultural contexts.

3. Task description and dataset

The competition explained in more detail in [1] was divided into 3 subtasks:

- Subtask 1: detection of hate speech, inappropriate content, and neither of them.
- Subtask 2: finer-grained detection of hate speech.
- Subtask 3: same categories from subtask 1 but participants are restricted to focus exclusively on the use of Large Language Models (LLMs) for detecting the specified categories.

For this work we focus on subtask 1, which involves 3-way classification, where each meme must exclusively and uniquely adhere to one of the following classes: hate speech, inappropriate content, and neither of them. In order to evaluate the proposed solutions for subtask 1, the competition organizers established that the macro-averages of precision, recall and F1-score would be reported. However, the macro-average of F1-score would be the primary evaluation metric for each subtask. The challenge site available at Codalab platform [14] was used for the submission of proposals and their evaluation.

4. Methodology

4.1. Dataset description

The competition organizers provided a training dataset, which is composed of 2,263 records. Each element is structured in three fields, the first is "MEME-ID", which corresponds to the unique identifier or file name of the image. The second field stores the text present in the meme, while the third provides a textual description of the meme's visual content. In addition, a link to download the associated images was provided. Table 1 presents an example of this dataset.

Table 1
Dataset Example

MEME-ID	Text	Description
DS_IMG_2448.jpg	NO ESTÁ MAL	La imagen es un meme que presenta un primer plano de un hombre con piel oscura y cabello corto. Tiene una expresión facial que combina una sonrisa contenta con un guiño, lo que sugiere un tono ligero o cómico. El fondo es de un color marrón suave, posiblemente una pared o cortina. En la parte superior de la imagen, hay un texto en letras mayúsculas que dice \"NO ESTÁ MAL\". El estilo del meme es característico de aquellos que utilizan expresiones faciales para transmitir humor o ironía.

In addition, a file containing the corresponding labels to each record was provided. These labels are represented using One-Hot Encoding, indicating the presence or absence of the elements to be classified. The assignment of tags follows a specific order from left to right, defined by the following values: hate speech, inappropriate content and neither. For example, if a record has the tag "1, 0, 0", it means that it has been classified as hate speech.

4.2. Data analysis

The analysis of the dataset was crucial in the early stages, verifying the consistency between the data provided and the original images. The content was reviewed to avoid external elements (such as links or watermarks) that could affect the integrity of the set.

For this purpose, a random sample was taken, and the extracted text was compared with the text present in each image. The results confirmed that the text in the dataset corresponded to that of the image, excluding irrelevant elements. In addition, the recorded description accurately reflected the meaning of the meme (see Figure 1).

Figure 1: MEME example.

4.3. Implemented workflow

Figure 2 shows the workflow used from the preprocessing of the original dataset to the generation of the result files to be uploaded to the challenge platform.
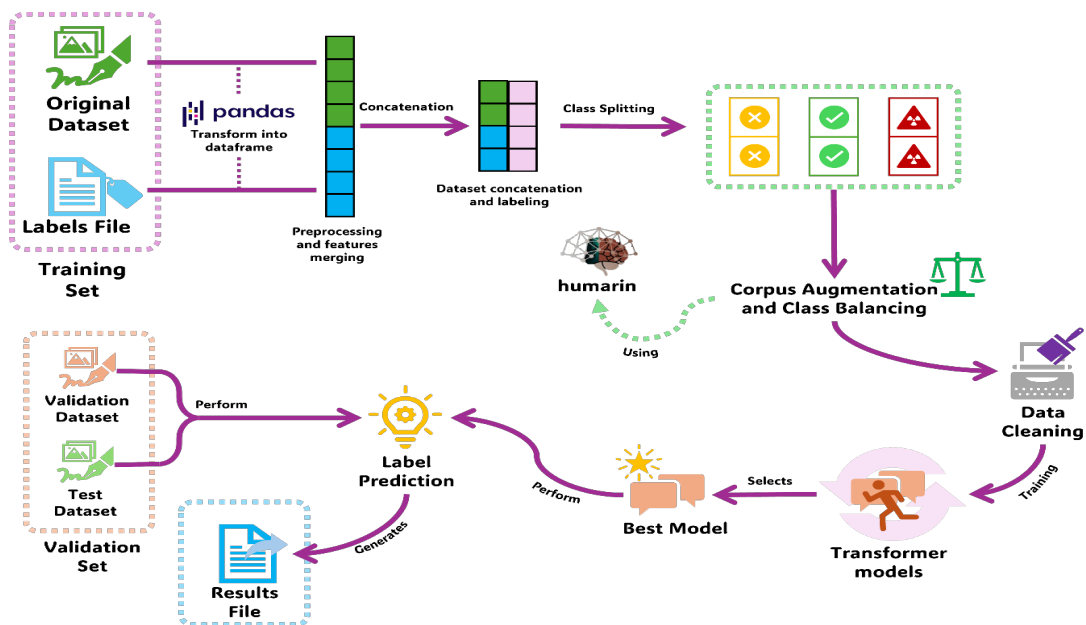

Figure 2: Full methodology to process the dataset, train the models, and generate the results file.

The following sections detail this workflow.

4.3.1. Preprocessing and features merging

In order to simplify the data structure and avoid label separation, the training set with its labeling was integrated into a single column. The process started with the conversion of the corpus from

the train_data file, from JSON format to a dataframe using Pandas. Before unifying the features, "MEME-ID" was removed, since it does not provide relevant information for training. Then "description" was added, followed by the phrase "there is a text in the image that contains the following", and finally the content "text" was added. This resulted in a Corpus with a single column of unlabeled data.

4.3.2. Dataset concatenation and labeling

For labeling, the training label file for the 3 subtasks was processed, transforming its content into a single column called "labels". Using the One-Hot Encoding format, the representation was simplified to a single numeric value: 0 for hate speech, 1 for inappropriate content and 2 for neither.

This column was incorporated into the unlabeled corpus, maintaining the correspondence between records and labels, resulting in the labeled corpus. Finally, it was divided into two sets: training (80% of the total, with 1,810 records) and test (20%, with 453 records).

4.3.3. Class splitting

The original corpus was separated into three subsets according to the tags: hate speech, inappropriate content and neither, thus obtaining three independent corpora, each containing exclusively records from a single category.

4.3.4. Corpus augmentation and class balancing

We used generative AI to expand and balance the corpora, ensuring that each category had the same number of records. To this effect, key parameters such as the amount of data per corpus, number of beams, penalties, temperature, among others, were established. In addition, input and output routes, the Humarin paraphrase model [15] and the language (Spanish) were defined. Although several hyperparameter configurations were reviewed, the best results were obtained with those proposed by the creators of the Humarin model [15]. Finally, the generated corpora were stored in an array, and with the data obtained a total corpus with balanced classes was constructed.

The process was applied to each corpus individually. First, the number of elements was counted and the difference with the desired total number was calculated. Then, the minimum number of repetitions per item was determined using generative AI and recorded in the "times_repeat" feature.

Subsequently, the residual amount needed to complete the total increase was adjusted by randomly selecting some elements and increasing their frequency in "times_repeat". This ensured a proper balance between classes. Finally, an array was created that integrated the original corpus with the AI-generated data.

Each element of the corpus is introduced into the text generation transformers-based model Humarin [15]. This model allows us to specify the number of similar texts to be generated, paraphrasing the input text and returning an array with the desired number of reformulated texts. Each new item was tagged with its corresponding original text and stored as new rows in the AI-generated corpus.

All items in the base corpus were added to the new corpus, which was stored in an array together with the sets generated for each label. The previous steps were then repeated for each corpus, until a total of 1,500 records for each of the three categories was reached, resulting in a final corpus of 4,500 records.

4.3.5. Data Cleaning

This process aims to prepare the textual data of the augmented corpus for training. Cleaning and normalization were applied to the "text" column, eliminating stopwords and lemmatizing the

words to reduce them to their base form. Finally, punctuation marks were removed, and the text was tokenized. The result was stored in a new corpus with the processed data.

4.3.6. Classification models training

Transformers-based classification models were used, including BERT, BETO and RoBERTa. The same hyperparameters were used in all of them (epochs=5, train batch size = 8, optimizer = "AdamW" and learning rate = 0.0001). In addition, the corpus was divided into 80% for training and 20% for validation. Each of the training iterations stored the model in a folder for later use and generated accuracy, f1-score, recall and precision metrics, which were stored in a "txt" file. The training process was repeated 10 times to ensure variability and improve results through randomized runs. The models were trained using two versions of the augmented dataset: one without cleaning and the other with cleaning, applying the same processes in both for training the models.

4.3.7. Best model selection

For each transfomers-based model (BERT, BETO and RoBERTa), accuracy, f1-score, recall and precision metrics were recorded. They were then evaluated with the test corpus, comparing the results with those obtained in training. The average of the four metrics was calculated for each model and the one with the best value was selected. The remaining models were discarded.

After completing the process with both data sets (corpus with and without cleaning), it was determined that BETO-based model without cleaning achieved the best results.

4.3.8. Prediction of validation and test dataset labels

Finally, the data from the validation dataset (development phase) and test dataset (final phase) were preprocessed for labeling. Each item of the provided datasets was processed using the selected BETO-based model, generating the predicted labels and storing them in a ".csv" file for subsequent submission to the challenge platform.

5. Results

The results obtained in the two phases of the competition for subtask 1 are shown below. As illustrated in Table 2, the results of the development phase are presented. During this stage, the unlabeled dataset provided was utilized. The submissions of the team are collectively referred to as ITC. In this phase, the proposed solution was ranked sixth, with a value of 0.48 in the f1 metric. It is noteworthy that the first six positions obtained values in a range of 11 hundredths, varying from 0.48 to 0.59 in the f1 metric, 10 hundredths in the precision metric with values from 0.49 to 0.59, and 12 hundredths in the recall metric, taking values from 0.47 to 0.59.

Table 2
Subtask 1 Development Phase Results

#	User/Team	f1	Precision	Recall
1	Ryuan	0.59 (1)	0.59 (3)	0.58 (1)
2	HoracioJarquin	0.55 (2)	0.61 (2)	0.54 (4)
3	hugojair	0.55 (3)	0.56 (5)	0.55 (3)
4	ymlopez	0.54 (4)	0.56 (4)	0.53 (5)
5	girish_koushik	0.53 (5)	0.53 (7)	0.58 (2)

6	**ITC (Vickbat)**	**0.48 (6)**	**0.49 (8)**	**0.47 (6)**
7	dmoctezuma	0.41 (7)	0.66 (1)	0.42 (7)
8	csuazob	0.27 (8)	0.54 (6)	0.34 (8)

The results of the final phase are presented in Table 3. In this stage, the proposed solution was ranked fourth among the three metrics evaluated. It is noteworthy that the initial four positions were the only ones that attained values above 0.50, with a mere 0.06 separating the first and fourth positions. The observation that all teams obtained values below 0.6 in all metrics indicates the complexity of the task.

Table 3
Subtask 1 Final Phase Results

#	User/Team	f1	Precision	Recall
1	Ryuan	0.58 (1)	0.58 (2)	0.58 (1)
2	Onarion	0.57 (2)	0.58 (1)	0.56 (2)
3	ymlopez	0.55 (3)	0.57 (3)	0.55 (3)
4	**ITC (Vickbat)**	0.52 (4)	0.54 (4)	0.51 (4)
5	michaelibrahim	0.44 (5)	0.46 (5)	0.43 (5)
6	John94	0.43 (6)	0.45 (6)	0.42 (6)
7	VeronicaNeriMendoza	0.42 (7)	0.42 (8)	0.42 (7)
8	csuazob	0.34 (8)	0.42 (7)	0.36 (8)
9	AngelBaron	0.33 (9)	0.35 (9)	0.34 (8)

6. Conclusions

This article presents the methodology and results of the ITC team's participation in the DIMEMEX competition as part of IberLEF 2025. Our team's efforts were concentrated on subtask 1, which entailed the classification of memes in Mexican Spanish into three categories: hate speech, inappropriate content, and neither of these categories. The most effective approach was identified as an incremental augmentation of elements within each class, employing generative AI to optimize the balance to 1,500 elements per class. Subsequently, a model based on Transformers was utilized. Specifically, the BETO-based model provided the most optimal result.

Our proposal was recognized for its noteworthy performance, securing fourth place in the final phase of subtask 1. However, it is important to note that the results generally demonstrate the complexity of the task and that, although the results are encouraging, it is essential to continue refining the classification techniques to enhance the performance of the models.

Acknowledgements

Declaration on Generative AI

The authors have not employed any Generative AI tools during the preparation of this work.

References

[1] H. Jarquín-Vásquez, I. Tlelo-Coyotecatl, D.I. Hernández-Farías, H.J. Escalante, L. Villaseñor-Pineda, M. Montes-y-Gomez, DIMEMEX@IberLEF2025: Detection of Inappropriate Memes from Mexico. Procesamiento del Lenguaje Natural 75 (2025).

[2] J. A. González-Barba, L. Chiruzzo, S. M. Jiménez-Zafra, In Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2025), co-located with the 41st Conference of the Spanish Society for Natural Language Processing (SEPLN 2025), CEUR-WS.org, 2025.

[3] M. Refaj Hossan, N. Sakib, M. Alam Miah Jawad Hossain, M. Moshiul Hoque, CUET_Big_O@NLU of Devanagari Script Languages 2025: Identifying Script Language and Detecting Hate Speech Using Deep Learning and Transformer Model, 2025. URL: https://github.com/vikaskumarjha9/hindi_.

[4] K. Sarveswaran, S. Thapa, S. Shams, A. Vaidya, B. K. Bal, Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025), 2025. URL: https://sites.google.com/view/chipsal/.

[5] M. Velmala, S. Rajiakodi, K. Pannerselvam, B. Sivagnanam, Multimodal Sentiment Analysis of Online Memes: Integrating Text and Image Features for Enhanced Classification. Procedia Computer Science, 258, (2025) 355–364. doi:10.1016/j.procs.2025.04.272.

[6] J. Badour, J. A. Brown, Hateful Memes Classification using Machine Learning. 2021 IEEE Symposium Series on Computational Intelligence, SSCI 2021 – Proceedings, 2021. doi:10.1109/SSCI50451.2021.9659896.

[7] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, D. Testuggine, The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes, 2020. doi:10.48550/arXiv.2005.04790.

[8] L. Sukanya, J. Aniketh, E. Abhiman Sathwik, M. Sridhar Reddy, N. Hemanth Kumar, Racism detection using deep learning techniques. E3S Web of Conferences, 391, 2023. doi:10.1051/e3sconf/202339101052.

[9] T. T. Nguyen, X. Yue, H. Mane, K. Seelman, P. S. P. Mullaputi, E. Dennard, A. S. Alibilli, J. S. Merchant, S. Criss, Y. Hswen, Q. C. Nguyen, Decoding Digital Discourse Through Multimodal Text and Image Machine Learning Models to Classify Sentiment and Detect Hate Speech in Race- and Lesbian, Gay, Bisexual, Transgender, Queer, Intersex, and Asexual Community–Related Posts on Social Media: Quantitative Study. Journal of Medical Internet Research, 27(1) (2025). doi:10.2196/72822.

[10] Chauhan, S., & Kumar, A. (2025). MNLP@DravidianLangTech 2025: Transformer-based Multimodal Framework for Misogyny Meme Detection. https://github.com/stopwords-iso/.

[11] M. Choudhary, B. Agarwal, V. Goyal, Hate Speech Detection: Leveraging LLM-GPT2 with Fine-Tuning and Multi-Shot Techniques. Procedia Computer Science, 258, (2025) 2817–2825. doi:10.1016/j.procs.2025.04.542.

[12] X. Shen, Y. Wu, Y. Qu, M. Backes, S. Zannettou, Y. Zhang, HateBench: Benchmarking Hate Speech Detectors on LLM-Generated Content and Hate Campaigns, 2025. doi:10.48550/arXiv.2501.16750.

[13] C. Yuxuan, W. Jiayang, A. C. L. Chuen, B. S. Guanrong, B. S. Jen, S. C. Z. Shen, Detecting Offensive Memes with Social Biases in Singapore Context Using Multimodal Large Language Models, 2025. doi:10.48550/arXiv.2502.18101.

[14] DIMEMEX, Challenge Website, 2025. URL: https://codalab.lisn.upsaclay.fr/competitions/22012.

[15] V. Vorobev, M. Kuznetsov, A paraphrasing model based on ChatGPT para-phrases, 2023. URL: https://huggingface.co/humarin/chatgpt_paraphraser_on_T5_base.