May 2025

Hate Speech and Emojis: the journalist case

David LARROSA-CAMPS [a], Jaume SUAU [b] and Xavier VILASÍS-CARDONA [b]
[a] *Smart Society Research Group, La Salle-Universitat Ramon Llull*
[b] *Digilab, Blanquerna-Universitat Ramon Llull*

Abstract. Work in progress towards a simple and efficient use of emojis for hate speech detection is reported. The dataset used has been generated in the framework of studying hate speech against journalists. The results of the method are promising, yet not fully conclusive.

Keywords. Natural Language Processing, Hate Speech, Emojis

1. Introduction

Hate speech, which is strongly related with the spread of disinformation [1], has become one of the biggest challenges that democratic societies are facing today. The relation between disinformation and hate speech in journalism is twofold. First, attacks on and threats to journalists and the media constrain freedom of expression and dissemination of certain information, and generate fear and self-censorship, undermining the media's role in democratic societies [2]. Second, disinformation campaigns often exploit hate narratives to amplify polarization, delegitimize the press, and erode trust in democratic institutions.

For reasons such as the ones described above, hate speech detection has become a pivotal challenge in natural language processing (NLP). Recent advancements leverage transformer-based models, multimodal analysis, and fairness-aware techniques to improve detection accuracy while addressing biases [3,4]. Despite progress, challenges such as contextual ambiguity, cross-lingual generalization, and ethical trade-offs persist.

State-of-the-art hate speech detection systems predominantly rely on transformer architectures such as BERT [5] and RoBERTa [6], which excel at capturing implicit hate speech and sarcasm through contextual embeddings. Fine-tuning these models on domain-specific datasets (e.g., Gab, Twitter) has proven effective, though performance drops in cross-platform scenarios [7]. Spanish hate speech detection presents unique challenges due to linguistic diversity (e.g., regional dialects, Spanglish) and cultural context. Models like BETO [8] (a Spanish BERT variant) and multilingual approaches (e.g., XLM-T [9]) have shown promise, but performance varies across regions (e.g., Mexico vs. Spain).

In our analysis of data from the X platform in search for hate speech against journalists in Spain, we have remarked the heavy use of emojis. Emojis, faithful to

the name, represent the emotions of the writer and are a valuable tool to identify hate speech. However, they are not part of the grammar of natural language, with which recurrent or transformer models have been trained. Because of this, these models may be confused by their presence. In this note, we investigate a strategy to use them in the detection task, combined with text analysis.

2. Data Collection

The data were collected from the social media platform X (formerly Twitter), chosen due to its high levels of toxicity and the presence of all selected journalists. The data collection process took place during the 2024 European elections and the days immediately following. A curated list of journalists to monitor was created, ensuring balance in terms of sector, gender, and other relevant criteria. Tweets and replies were automatically downloaded using Python scripts via the official X API. The endpoints used included user lookup, user tweets, and user mentions.

In total, 41,791 tweets were manually labeled, with only 4.6% (1,947 tweets) identified as containing hate speech, resulting in a highly imbalanced dataset. The labeling was carried out by four annotators, each of whom labeled approximately 10,447 tweets. To address the class imbalance problem, the majority class was undersampled to match the minority class. This resulted in a final dataset composed of 3,894 tweets, with an equal distribution of 1,947 hateful tweets (50%) and 1,947 non-hateful tweets (50%). Among the non-hateful tweets, only 256 (6.57%) contained emojis, while 1,691 (43.43%) did not. Similarly, among the hateful tweets, 259 (6.65%) included emojis, while 1,688 (43.35%) did not. Descriptive analysis revealed a higher incidence of hate speech directed toward women and sports journalists.

3. Dealing with Emojis

When processing the data, a major challenge was the dual nature of tweets, which often contain both textual content and emojis. While most Natural Language Processing (NLP) techniques are well-suited for textual data, they are not inherently designed to interpret emojis, which convey affective and contextual nuances that can significantly alter the meaning of a message. Ignoring emojis can therefore lead to a substantial loss of sentiment-related information.

To address this issue, the preprocessing pipeline was designed to treat text and emojis as two separate input modalities. The text was cleaned by removing user mentions, retweet markers, punctuation, numbers, and any emojis. This step ensured that only clean textual content was passed to the language processing models. For the emoji data, the process involved extracting all emojis from the original tweet and assigning each one a sentiment score. To obtain these scores, we used the `emosent` Python library[1], which is based on the sentiment values published in the study "Sentiment of Emojis" [10]. This study provides manually annotated sentiment scores—expressed as numerical continuous values represent-

[1] `https://github.com/omkar-foss/emosent-py/blob/master/README.md`

ing positive, neutral, or negative sentiment—for a wide range of emojis, based on human perception. However, since the original dataset is from 2015 and many new emojis have been introduced since then, we manually annotated the missing emojis to ensure completeness and consistency in the scoring process.

After assigning sentiment values to each emoji, we computed an aggregated sentiment score for the emoji set in each tweet using a frequency-weighted formula,

$$Emoji\ score = \sum_{n=0}^{N} f(n) \cdot s(n) \tag{1}$$

where $f(n)$ is the frequency of the n-th emoji and $s(n)$ is its sentiment score. This formula takes into account all the emojis present in the tweet, giving more weight to the most frequently used ones, thus emphasizing the predominant sentiment. Finally, the resulting score was normalized to fall within a predefined range, ensuring consistency across tweets regardless of the total number of emojis used.

4. Experiments

Experiments were conducted by splitting the dataset into three blocks: one for training, one for testing, and one for final evaluation, with proportions of 2726, 584, and 584 samples respectively.

In order to investigate the key components required for effective hate speech detection, a sequence of progressively refined models was designed. The first baseline model consists of a Recurrent Neural Network (RNN) built on top of frozen BETO embeddings. This serves as a foundational system to evaluate the capacity of contextual embeddings combined with a lightweight recurrent architecture.

Next, an improved version was developed by fine-tuning the BETO model itself, allowing the embeddings to adapt specifically to the task. To further enhance performance, especially in the presence of emojis—which often carry emotional or contextual nuance—two additional models were proposed. Both of these incorporate an MLP layer after the RNN output to combine textual and emoji information simultaneously. In the first approach, emojis are represented using sentiment-based scores, while in the second, they are encoded using one-hot vectors. These configurations aim to assess the contribution of emoji-aware features in the context of hate speech classification.

4.1. RNN with BETO Embeddings

The first model explored is a Recurrent Neural Network (RNN) built on top of frozen BETO embeddings. BETO is a Spanish-language BERT model used here solely to generate contextualized word representations, which are then fed into a GRU-based RNN. The architecture includes a single GRU layer with a hidden size of 64, followed by a dropout layer (0.5) and a fully connected output layer. This setup serves as a baseline to evaluate whether a simple RNN, leveraging pre-

trained contextual embeddings without fine-tuning, can effectively detect hate speech.

To understand the effect of emoji presence, the model was trained and evaluated under two conditions using Twitter posts: one where emojis were removed from the text, and another where they were preserved and tokenized as part of the input sequence. Interestingly, the configuration without emojis outperformed the emoji-preserving version across all metrics. These results suggest that, without specific emoji-aware modeling, the presence of emojis may introduce noise into the input representation, slightly degrading classification performance.

The performance metrics obtained by the model are shown in the following table:

Table 1. Performance of the RNN model with BETO embeddings on Twitter data, comparing text with and without emojis.

Metric	**No Emojis**	**Tokenized Emojis**
F1 Score	$0.7262 \pm 0.17 \times 10^{-3}$	$0.6977 \pm 0.13 \times 10^{-3}$
Accuracy	$0.7266 \pm 0.16 \times 10^{-3}$	$0.6978 \pm 0.13 \times 10^{-3}$
Precision	$0.7366 \pm 0.19 \times 10^{-3}$	$0.7083 \pm 0.19 \times 10^{-3}$
Recall	$0.7357 \pm 0.25 \times 10^{-3}$	$0.6986 \pm 0.28 \times 10^{-3}$
ROC-AUC	$0.7262 \pm 0.17 \times 10^{-3}$	$0.6978 \pm 0.13 \times 10^{-3}$

After analyzing the results, it becomes evident that retaining emojis within the input text introduces noise, negatively affecting model performance. This suggests that emojis may require dedicated preprocessing or separate modeling strategies in order to contribute meaningfully to hate speech detection.

Moreover, the overall performance metrics indicate that the model has learned effectively. The F1 score, precision, and recall are well balanced, and the ROC-AUC score confirms that the model is able to distinguish between classes with a high degree of reliability.

4.2. RNN with Fine-Tuned BETO

The second model builds upon the previous architecture by introducing partial fine-tuning of the BETO encoder. Specifically, six of BETO's twelve transformer layers are unfrozen and updated during training, allowing the model to adapt contextual embeddings to the specific hate speech detection task. The RNN architecture remains unchanged, with a single GRU layer of size 64, followed by dropout and a fully connected classification layer.

As before, the model was evaluated under two input conditions: one with emojis removed from the text, and another with emojis preserved and tokenized. The results demonstrate that fine-tuning BETO significantly improves overall performance when compared to the frozen-embedding setup. However, consistent with the previous model, the inclusion of tokenized emojis slightly degrades performance across most metrics, again suggesting that simple tokenization of emojis may introduce noise rather than semantic value.

The performance metrics obtained by the model are summarized in Table 2:

Table 2. Performance of the RNN model with partially fine-tuned BETO embeddings on Twitter data, comparing text with and without emojis.

Metric	No Emojis in Text	Tokenized Emojis in Text
F1 Score	$0.7556 \pm 0.12\times10^{-3}$	$0.7157 \pm 0.05\times10^{-3}$
Accuracy	$0.7558 \pm 0.12\times10^{-3}$	$0.7166 \pm 0.05\times10^{-3}$
Precision	$0.7714 \pm 0.21\times10^{-3}$	$0.7150 \pm 0.20\times10^{-3}$
Recall	$0.7509 \pm 0.44\times10^{-3}$	$0.7451 \pm 0.87\times10^{-3}$
ROC-AUC	$0.7559 \pm 0.12\times10^{-3}$	$0.7159 \pm 0.05\times10^{-3}$

The results indicate that increasing the complexity of the embeddings via partial fine-tuning leads to a clear improvement in classification performance. Additionally, the presence of tokenized emojis continues to degrade results, reinforcing the idea that emojis should be treated separately rather than as part of the regular token stream.

Overall, both versions of the model exhibit balanced performance across precision, recall, and F1 score. The high ROC-AUC values confirm that the models are capable of correctly distinguishing between hateful and non-hateful content.

To further explore the potential of this architecture, several variations of the model were tested, including different numbers of GRU layers and configurations of fine-tuned BETO layers. The combination presented above yielded the best overall performance across all evaluation metrics.

In the next phase of the study, a new family of models was introduced to explicitly incorporate emoji information. These models combine the textual analysis performed by the RNN with parallel emoji processing, aiming to improve classification by capturing emotional and contextual cues expressed through emojis.

4.3. RNN with Fine-Tuned BETO and Emoji Score Integration

The third model extends the previous architecture by incorporating explicit emoji information into the classification process. As in the earlier setup, the model uses a partially fine-tuned BETO encoder, where six of the twelve layers are trainable, followed by a GRU-based RNN to capture sequential dependencies in the textual embeddings. To enrich the representation with emotional cues carried by emojis, a second input stream is introduced: a numerical score summarizing the sentiment conveyed by all emojis in each tweet. This score is concatenated with the RNN output and passed through a Multi-Layer Perceptron (MLP) for final classification.

The architecture of this model is illustrated in Figure 1, where the integration between textual and emoji-based features can be seen.

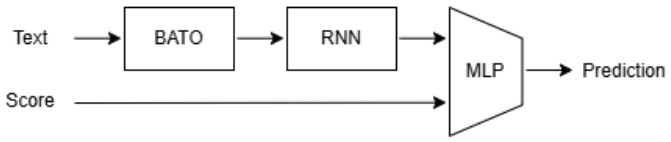

Figure 1. Architecture of the RNN+MLP model integrating emoji score with fine-tuned BETO embeddings.

As with previous experiments, the model was tested under two conditions: with and without emojis included in the tokenized text. The performance results are shown in Table 3:

Table 3. Performance of the RNN+MLP model with emoji score integration.

Metric	**No Emojis in Text**	**Tokenized Emojis in Text**
F1 Score	$0.7912 \pm 0.26\times10^{-3}$	$0.7630 \pm 0.40\times10^{-3}$
Accuracy	$0.7933 \pm 0.24\times10^{-3}$	$0.7645 \pm 0.39\times10^{-3}$
Precision	$0.7850 \pm 2.18\times10^{-3}$	$0.7705 \pm 1.06\times10^{-3}$
Recall	$0.8334 \pm 5.10\times10^{-3}$	$0.7830 \pm 5.06\times10^{-3}$
ROC-AUC	$0.7936 \pm 0.37\times10^{-3}$	$0.7638 \pm 0.37\times10^{-3}$

The results indicate that the combination of textual embeddings with an aggregated emoji score leads to improved classification performance, surpassing all previous models in both accuracy and F1 score when emojis are excluded from the input text. This reinforces the hypothesis that emoji information is indeed valuable for hate speech detection—particularly when handled separately from the main text.

The model also exhibits a bias towards recall, favoring the identification of potentially hateful content even at the expense of precision. This trade-off may be acceptable or even desirable in hate speech detection scenarios, where failing to flag harmful messages can be more critical than occasionally misclassifying benign ones.

However, as observed in prior models, directly tokenizing emojis within the input still introduces noise, leading to a consistent performance drop across all metrics. These findings support the strategy of processing textual and emoji information through separate paths in future model designs.

4.4. RNN with Fine-Tuned BETO and One-Hot Emoji Encoding

This model shares the same architectural foundation as the previous one, combining a fine-tuned BETO encoder and a GRU-based RNN for text representation. However, instead of using a scalar emoji sentiment score, it introduces a richer representation by encoding the presence of each emoji in a tweet using a one-hot vector. This allows the model to preserve the identity of individual emojis and potentially capture more nuanced associations between specific emojis and hate speech.

The one-hot vector is concatenated with the output of the RNN and passed through a Multi-Layer Perceptron (MLP) for classification. This architecture is

designed to evaluate whether fine-grained emoji features provide additional discriminatory power when compared to a single aggregated score.

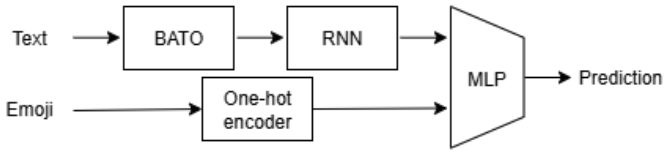

Figure 2. Architecture of the RNN+MLP model using one-hot emoji encoding.

The model was tested under the same conditions as before: with emojis included in the tokenized text and with emojis entirely removed. The results are summarized in Table 4.

Table 4. Performance of the RNN+MLP model with one-hot emoji encoding.

Metric	**No Emojis in Text**	**Tokenized Emojis in Text**
F1 Score	$0.7959 \pm 0.18\times10^{-3}$	$0.7564 \pm 0.22\times10^{-3}$
Accuracy	$0.7973 \pm 0.15\times10^{-3}$	$0.7571 \pm 0.20\times10^{-3}$
Precision	$0.7861 \pm 1.15\times10^{-3}$	$0.7654 \pm 0.77\times10^{-3}$
Recall	$0.8382 \pm 1.42\times10^{-3}$	$0.7671 \pm 1.42\times10^{-3}$
ROC-AUC	$0.7960 \pm 0.18\times10^{-3}$	$0.7568 \pm 0.22\times10^{-3}$

The results show that this one-hot emoji representation leads to the best overall performance among all tested models. In contrast to the previous experiments, including tokenized emojis in the input no longer degrades performance. In fact, when used together with the one-hot encoding vector, the model achieves its highest F1 score and recall. This suggests that combining rich textual embeddings with structured emoji features allows the model to leverage emoji signals effectively without introducing noise.

As in the previous setup, the model demonstrates a bias toward recall, which may be desirable in safety-sensitive hate speech detection tasks. The balance across metrics further confirms the robustness of this architecture.

5. Discussion and Conclusion

Treating separately the emojis seems to reinforce the performance of the detection system. However, we cannot consider these results conclusive. The amount of data is scarce, due to the imbalance and may limit the capability of the RNNs. Further data collection and labeling is under way to confirm the analysis. ANOVA and Tukey HSD tests confirmed statistically significant differences between all models ($p < 0.001$), with RNN + MLP (emoji score) outperforming the rest. This model also showed the highest recall, though without introducing bias, as its F1 score remained balanced and differences were not statistically significant.

After evaluating the performance metrics of each model, an ANOVA test was conducted to assess whether the observed differences are statistically signifi-

cant. The test aims to determine whether the models exhibit genuinely different performance or whether the differences could be attributed to random variation.

A Tukey HSD post-hoc test was conducted following the ANOVA to identify specific pairwise differences between models. The results indicate that all pairwise comparisons are statistically significant at a 95% confidence level ($p < 0.001$), even those with relatively small differences (e.g., between RNN (BETO finetuned) and RNN+MLP (one-hot), $\Delta = 0.0008$). This confirms that all four models achieve significantly different F1-scores, with RNN+MLP (score) outperforming the rest. The largest performance gap is observed between RNN (embeddings) and RNN+MLP (score) ($\Delta = 0.0368$), providing strong evidence that including emoji scores as features improves classification.

Considering both performance metrics and computational efficiency, the most effective model is the RNN + MLP with emoji score encoding. This model achieves the highest values across all evaluated metrics. Notably, it presents the highest recall, which could suggest a tendency to favor one class over the other. However, this potential bias can be reasonably dismissed, as the difference in recall compared to the other models—as well as the difference with the F1 score—is minimal (both approximately 0.02) and not statistically significant.

5.1. Conclusions

In order to facilitate visual comparison between models, the following chart was created:

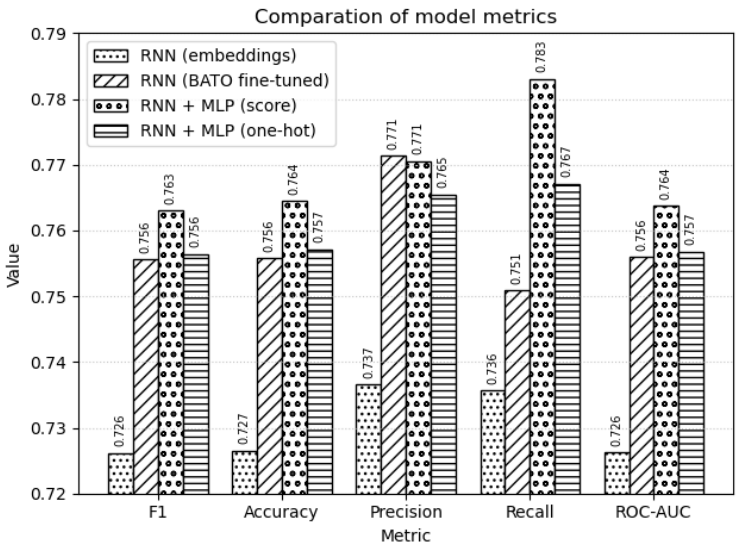

Figure 3. Performance comparison of the RNN+MLP models integrating emoji score and one-hot emoji encoding, both using fine-tuned BETO.

As illustrated in the figure, the model combining a fine-tuned BETO encoder, a recurrent layer, and an aggregated emoji score achieves the best overall per-

formance across evaluation metrics. Surprisingly, this model slightly outperforms the more complex one that uses one-hot emoji encoding, despite its simpler emoji representation. This suggests that aggregated emoji sentiment information is not only effective, but also computationally efficient.

Although the score-based model exhibits a minor bias toward one class—as indicated by its higher recall compared to precision—the difference is approximately 2% and is not considered substantial. This level of imbalance is acceptable in hate speech detection contexts, where maximizing recall is often prioritized to reduce the risk of false negatives.

In summary, the experiments demonstrate that integrating emoji features—especially when modeled separately from text—can significantly enhance hate speech detection. The findings also highlight the importance of choosing the right representation for non-textual features, balancing model complexity with performance gains.

6. Label Quality and LLM-Assisted Relabeling

After analyzing the performance of the models, it became evident that label quality was a significant limiting factor. Upon closer inspection of the original dataset, numerous inconsistencies were identified in the annotations, which negatively impacted both model learning and generalization. This observation is consistent with previous studies using the same dataset, where reported F1 scores typically plateau around 0.79. Notably, this performance ceiling has also been observed in additional models not covered in this paper, as well as in ongoing parallel research projects at the same institution using the same dataset. These recurring results suggest that the observed limitations are not specific to any single architecture but are instead likely caused by noisy or subjective labeling practices.

To address this limitation, a label refinement phase was introduced using Large Language Models (LLMs), selected from publicly available repositories on Hugging Face [11,12]. The goal was twofold: first, to evaluate whether LLMs could replicate the annotation logic used by human labelers; and second, to explore whether LLMs might apply more consistent or effective labeling criteria, potentially allowing downstream models to surpass the existing performance ceiling. Two types of LLMs were tested: general-purpose instruction-following models, and models specifically fine-tuned for hate speech detection.

6.1. LLM Evaluation and Label Quality Analysis

Among the models tested, two were selected for detailed evaluation: the specialized hate speech model `mrm8488/bert-tiny-finetuned-sms-spam-detection`, and the general-purpose model `meta-llama/Meta-Llama-3-8B-Instruct`. To assess the similarity between their label outputs and the original human annotations, Kendall's Tau rank correlation coefficient was used. This non-parametric statistic measures ordinal association between two sequences: values close to 1 indicate strong agreement, while values near 0 reflect weak or no correlation.

As shown in Table 5, the specialized model exhibits almost no correlation with the human labels, while the general model shows moderate agreement. This

Table 5. Kendall's Tau correlation between LLM-generated labels and human annotations.

Model	Tau	p-value
mrm8488/bert-tiny-finetuned-sms-spam-detection	0.05364	0.00117
meta-llama/Meta-Llama-3-8B-Instruct	0.36233	1.27×10^{-106}

suggests that the specialized model may be using an entirely different internal labeling criterion.

To assess the impact of these new label sets, the RNN+MLP (emoji score) model was retrained using the datasets relabeled by each LLM. The performance results are presented in Table 6.

Table 6. F1 scores of the RNN+MLP (score) model trained on LLM-generated labels.

Model	F1 Score
mrm8488/bert-tiny-finetuned-sms-spam-detection	0.8574 ± 0.572
meta-llama/Meta-Llama-3-8B-Instruct	0.7940 ± 0.294

The results lead to several key insights. The general-purpose model, despite following human-defined labeling criteria, did not exceed the performance ceiling observed with the original annotations. In contrast, the specialized hate speech model—although poorly aligned with human labels—enabled the downstream classifier to achieve an F1 score of 0.8574, the highest in the entire study. This suggests that the fine-tuned model did not necessarily discover a novel labeling criterion, but instead applied a distinct and internally consistent one that proved more effective for training purposes. The low Kendall's Tau score reinforces the idea that this alternative criterion diverges substantially from human annotations, yet performs better in terms of functional outcomes.

Interestingly, the high performance of the specialized LLM may also be attributed to its consistent application of a single decision-making framework. Unlike human annotations, which often result from the combination of written guidelines and individual judgment—leading to intra-annotator variability—the LLM follows its own fixed internal logic throughout the entire dataset. This consistency, even if misaligned with the original intent of the annotation scheme, may explain its superior downstream results.

These findings invite a broader reflection on how labeling guidelines are developed. Rather than aiming solely for inter-annotator agreement or replicating subjective human criteria, it may be worthwhile to consider whether alternative, machine-consistent labeling schemes can lead to improved downstream model performance.

6.2. Conclusions

This study highlights the critical role of annotation quality in machine learning pipelines for hate speech detection. The presence of labeling inconsistencies imposes a clear performance ceiling, regardless of model complexity. Experimentation with LLM-generated labels demonstrated that alternative annotation schemes—particularly those derived from specialized models—can surpass traditional human-labeled datasets in terms of classifier effectiveness.

These results underscore the importance of standardized and consistent labeling practices when creating reliable datasets. While human annotation remains the gold standard in many domains, this work shows that, depending on available resources and annotator expertise, LLMs can serve as a viable alternative or complement. Their use may be particularly justified in contexts where human labeling is cost-prohibitive, lacks consistency, or cannot be easily scaled. Future work should further explore strategies for aligning LLM-generated annotations with domain-specific requirements while ensuring reproducibility and interpretability.

7. Final Conclusions and Future Work

7.1. Conclusions

This study has explored multiple aspects of hate speech detection using neural models and enriched text representations. A key finding is that emojis—often overlooked in natural language processing—carry important semantic and emotional cues. However, when they are treated as regular text tokens, they introduce noise and reduce model performance. Results consistently showed that emojis should be processed separately from the main text in order to contribute meaningfully to the classification task.

Moreover, it was demonstrated that the complexity of emoji processing plays a significant role. More sophisticated representations—such as one-hot encodings and aggregated sentiment scores—led to improved results over simpler or tokenized approaches. Still, despite architectural improvements, all models encountered a performance ceiling around an F1 score of 0.79. This plateau was attributed not to model limitations, but rather to the inconsistency and noise present in the original dataset labels.

To investigate this further, Large Language Models (LLMs) were introduced as relabeling agents. Their application revealed that it is possible to obtain better-performing models when trained on LLM-generated annotations. In particular, a specialized LLM for hate detection, although not aligned with human criteria, provided consistent and functionally effective labels that surpassed the performance ceiling. These findings highlight the potential of LLMs to support or even replace human annotators when professional annotation resources are scarce or inconsistent.

7.2. Future Work

Several promising directions emerge from this research that could enhance the understanding and performance of hate speech detection systems.

First, a new data collection effort is recommended, with a focus on increasing dataset size and improving annotation quality. Building a dataset with clearly defined guidelines and consistent labeling—potentially combining human experts with LLM-based support—would help reduce annotation noise and allow models to reach their full potential.

Second, future work should explore and compare different strategies for emoji representation. While this study evaluated two effective approaches—emoji

sentiment scores and one-hot encodings—other methods such as *emoji2vec* or transformer-based embeddings for emojis may offer richer semantic representations. A comparative study would help identify the most robust and generalizable encoding scheme.

Lastly, incorporating additional linguistic and contextual features into the models could further improve classification accuracy. Features such as the use of capital letters, punctuation intensity, or the political context surrounding the tweet may provide complementary signals that improve the model's ability to detect hate speech, especially in more subtle or implicit cases.

These future directions aim not only to improve model performance, but also to contribute to the development of more transparent, fair, and generalizable hate speech detection tools.

8. Acknowledgments

Authors want to thank Alvaro García-Piquer and Luis Servera for the data collection under the EU project Media Councils in the Digital Age. This work is supported by the project *Disargue*, reference TED2021-130810B-C22, from the Spanish MICINN.

References

[1] Anat Ben-David and Ariadna Matamoros Fernández. Hate speech and covert discrimination on social media: Monitoring the facebook pages of extreme-right political parties in spain. *International Journal of Communication*, 10(0), 2016.

[2] M. Clark and A. Grech. *Journalists Under Pressure: Unwarranted Interference, Fear and Self-censorship in Europe*. Council of Europe, 2017.

[3] M. Sap, D. Card, S. Gabriel, Y. Choi, and N. A. Smith. The risk of racial bias in hate speech detection. *Proc. ACL*, pages 1668–1678, 2019.

[4] B. Mathew, P. Saha, S. M. Yimam, C. Biemann, P. Goyal, and A. Mukherjee. Hatexplain: A benchmark dataset for explainable hate speech detection. *Proc. AAAI*, 35:14867–14875, 2021.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proc. NAACL*, 1:4171–4186, 2019.

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[7] C. J. Kennedy, G. Bacon, A. Sahn, and C. von Vacano. Constructing interval variables via faceted rasch measurement and multitask deep learning: A hate speech application. *PLoS ONE*, 15(12):e0243700, 2020.

[8] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, and J. Pérez. Spanish pre-trained bert model and evaluation data. *Proc. PML4DC at ICLR*, 2020.

[9] F. Barbieri, L. Espinosa-Anke, and J. Camacho-Collados. Xlm-t: Multilingual language models in twitter for sentiment analysis and beyond. *Proc. LREC*, pages 258–266, 2022.

[10] Borut Sluban Igor Mozetič Petra Kralj Novak, Jasmina Smailović. Sentiment of emojis. *PLoS ONE 10(12): e0144296*, 2015.

[11] mrm8488. bert-tiny-finetuned-sms-spam-detection. `https://huggingface.co/mrm8488/bert-tiny-finetuned-sms-spam-detection`, 2020. Accessed: 2025-07-21.

[12] Meta AI. Meta-llama-3-8b-instruct. `https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct`, 2024. Accessed: 2025-07-21.