A Web-Based Hate Speech Detection System for Dialectal Arabic

Anis Charfi[1], Andria Atalla[1], Raghda Akasheh[1], Mabrouka Bessghaier[1] and Wajdi Zaghouani[2]
[1]*Carnegie Mellon University in Qatar, Education City, Doha, Qatar*
[2]*Hamad Bin Khalifa University, Education City, Doha, Qatar*

Keywords: Natural Language Processing, Hate Speech, Arabic Language, Dialectal Arabic, Annotation, Arabic Corpus.

Abstract: A significant issue in today's global society is hate speech, which is defined as any kind of expression that attempts to degrade an individual or a society based on attributes such as race, color, nationality, gender, or religion (Schmidt and Wiegand, 2017). In this paper, we present a Web-based hate speech detection system that focuses on the Arabic language and supports its various dialects. The system is designed to detect hate speech within a given sentence or within a file containing multiple sentences. Behind the scenes, our system makes use of the AraBERT model trained on our ADHAR hate speech corpus, which we developed in previous work. The output of our system discerns the presence of hate speech within the provided sentence by categorizing it into one of two categories: "Hate" or "Not hate". Our system also detects different categories of hate speech such as race-based hate speech and religion-based hate speech. We experimented with various machine learning models, and our system achieved the highest accuracy, along with an F1-score of 0.94, when using AraBERT. Furthermore, we have extended the functionality of our tool to support inputting a file in CSV format and to visualize the output as polarization pie charts, enabling the analysis of large datasets.

1 INTRODUCTION

The increasing adoption of social media platforms over the past years gave rise to some issues such as the spread of hate speech. While these platforms provide a free environment for individuals to converse and express their viewpoints, the vast quantity of posts, comments, and messages exchanged poses significant challenges in effectively controlling their content (Watanabe et al., 2018) and detecting if they contain hate speech or offensive content. As a result, the necessity of creating efficient automated systems for detecting and dealing with hate speech in both online and offline settings is becoming more apparent.

The process of identifying and classifying words or other materials that promote hatred, prejudice, or acts of violence against specific people or groups is known as hate speech detection. In order to find patterns and indicators suggestive of hate speech, this procedure frequently makes use of linguistic analysis, machine learning (ML) algorithms, and natural language processing (NLP) techniques.

While few works created hate speech detection tools such as (Chaudhari et al., 2020) and (Vrysis et al., 2021), there is still a lack of systems that focus on Arabic and can support the large variety of Arabic dialects. To bridge this gap, we present a new Web-based hate speech detection system designed explicitly for Arabic, which also accommodates diverse dialects including Egyptian, Levantine, Maghreb, and Gulf in addition to Modern Standard Arabic (MSA), as a variant of the Arabic language used in official communication, education and media. Our system automatically identifies instances of hate speech within Arabic sentences. Moreover, it offers the capability to analyze CSV files containing multiple sentences, producing an output file with the prediction of each sentence as well as two pie chart visualizations for polarization analysis. The first pie chart illustrates the distribution of hate vs not-hate sentences in the analyzed input file. The second pie chart illustrates the distribution among hate speech categories such as race-based hate, religion-based hate, etc.

The remainder of this paper is organized as follows: Section 2 discusses prior research on hate speech detection for Arabic. Section 3 provides an overview of our Web-based hate speech detection system. Section 4 details the methodology used in creating our system, including insights into the dataset and ML models employed. In Section 5, we present a real-world use case, in which we evaluate our system and its accuracy in detecting hate speech. Section 6 summarizes our contributions and outlines directions for future work.

Charfi, A., Atalla, A., Akasheh, R., Bessghaier, M. and Zaghouani, W.
A Web-Based Hate Speech Detection System for Dialectal Arabic.
DOI: 10.5220/0012812000003756

In *Proceedings of the 13th International Conference on Data Science, Technology and Applications (DATA 2024)*, pages 478-485
ISBN: 978-989-758-707-8; ISSN: 2184-285X

2 RELATED WORK

In this section, we report on existing works on hate speech detection in Arabic. While there are some works on this topic there is no freely accessible system to detect hate speech in Arabic. We consulted (Ahmed et al., 2022) and (Zaghouani, 2017) who conducted reviews on freely and accessible Arabic language corpora from peer-reviewed papers.

In (Magnossão de Paula et al., 2022), transformer-based models like AraBERT and XLM-Roberta were used for detecting hate speech and offensive language in Arabic tweets. The experiments conducted in that work confirmed that ensemble methods achieved better results for that purpose compared to individual models. In (Almaliki et al., 2023), the authors addressed the task of hate speech detection in Arabic Twitter data using the Arabic BERT-Mini Model (ABMM) and achieved an accuracy of 98.6%.

Another work on hate speech detection in Arabic was presented by (Al-Ibrahim et al., 2023). The authors developed deep learning models like bidirectional LSTM and CNN to detect hate speech in Arabic Twitter data. The authors worked on an Arabic hate speech dataset of 15,000 tweets and their experiments confirmed that deep learning models outperformed traditional ML models. The top bidirectional LSTM model yielded an accuracy of 92.2% and an F1-score of 92%.

Furthermore, some shared tasks were organized on hate speech detection for Arabic such as those proposed by the 4th and 5th workshops on Open-Source Arabic Corpora and Processing Tools (OSACT). The shared task proposed by OSACT 4 [1] included a subtask on hate speech identification and another subtask on offensive language detection. In both tasks, one common corpus was used, which included 10,000 tweets that were annotated to indicate if they contain hate speech and if they contain offensive language. One issue with this corpus is that it is very imbalanced, with only 5% of the tweets annotated as hate speech (Mubarak et al., 2020).

The shared task organized by OSACT 5 [2] included three sub-tasks: hate speech detection, detection of hate speech class, and detection of offensive language. In this shared task, the dataset used was the one proposed by (Mubarak et al., 2022), which was balanced between the classes ”offensive” and ”clean” but is was quite imbalanced with respect to the hate speech sub-classes.

Teams in OSACT 4 used traditional ML techniques like SVM and Logistic Regression, as well as Deep Neutral Networking (DNN) approaches, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) including LSTM, biLSTM, and GRUs with and without attention, and fine-tuning of contextual embeddings such as BERT and AraBERT. The top-ranking submission for the hate speech detection task achieved an F1-score of 95%. This work used a Linear SVM-based classifier with a character-based count vectorizer, with the use of different pre-processing techniques (Husain, 2020).

On the other hand, for OSACT 5, the teams used various fine-tuned transformer versions, including mT5, AraBERT, ARBERT, MARBERT, AraElectra, QARiB, Albert-Arabic, AraGPT2, mBert, and XLMRoberta. The GOF team developed the winning system for the hate speech detection task (Mostafa et al., 2022), which employed an ensemble of three different deep learning models, using a majority voting mechanism among the following models: QARiB trained with dice loss, MARBERT with VS loss, and MARBERTV2 with focal loss and label smoothing. This ensemble achieved an F1-score of 85.2% and an accuracy of 86.7%.

3 SYSTEM OVERVIEW

In this section, we present our Web-based hate speech detection system for Arabic, whose architecture is illustrated in Figure 2. Our system showcases the effectiveness of the ADHAR dataset (Charfi et al., 2024) and the AraBERT model (Antoun et al., 2020) for the task of hate speech detection in dialectal Arabic. Next, we will discuss the tool interface, the tokenizer, the model, and the deployment strategy.

Figure 1: The Web Interface of our System.

[1] https://edinburghnlp.inf.ed.ac.uk/workshops/OSACT4/

[2] https://osact-lrec.github.io/

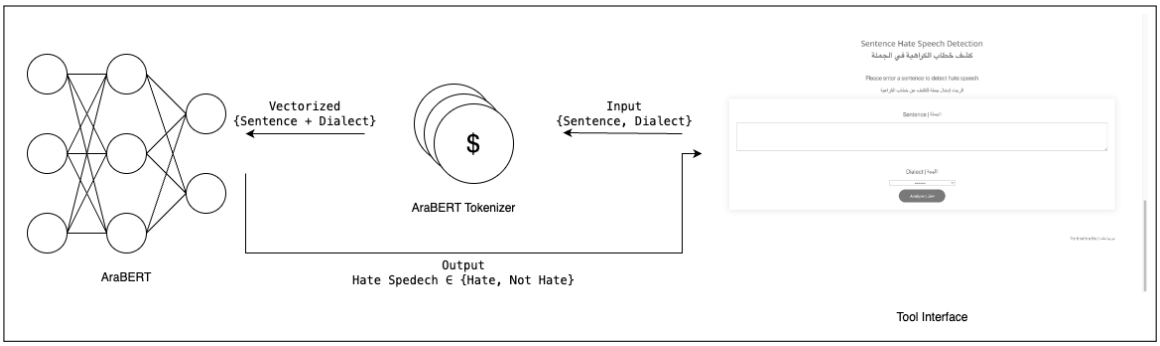

Figure 2: Overall System Architecture.

3.1 Interface

Our system has a Web interface that allows users to access the hate speech detection functionalities as shown in Figure 1. Our system is built on top of Python's Django model-controller-template architecture while making use only of the latter two. The controllers control the flow of data between what the user sees and the tokenizer and the model as shown in Figure 2. User inputs are made through forms and get sent as HTTP POST requests to the controllers.

The system processes two types of inputs: a sentence and its dialect or a file with multiple sentences and their corresponding dialects in CSV format. The inputs are tokenized and then sent to the model, which outputs two predictions: whether the sentence contains hate speech, and the hate speech category it falls under such as religion-based hate speech, race-based hate speech, etc. These predictions are returned to the controllers, which are then sent to the templates to be displayed on the Web interface.

When a file is passed as input to our system, the file is parsed into a string and passed to the template along with insights into the results, allowing the generation of a new CSV file with two additional columns for the predictions (i.e. `Hate Speech Label` and `Predicted Category`.) as well as two pie chart graphs, which illustrate the distribution of the hate speech predictions graphically.

3.2 Tokenizer

The AraBERT tokenizer breaks down the Arabic text into tokens, which are later fed into the model. Hosted in the Django framework files, the tokenizer is loaded as an instance when the server is initially loaded. The tokenizer was trained on our ADHAR hate speech corpus (Charfi et al., 2024) and it transforms the user's input into a format that can be processed by our AraBERT-based model for hate speech detection.

3.3 Model

The model is trained based on AraBERT. Similar to the tokenizer, a model instance is created when the server is initially loaded. The model is trained on tokens of the ADHAR dataset and generates either *Hate* or *Not hate* as output for hate speech detection, as shown in Figure 2. The controller uses the model instance to produce the output, which is then presented to the user by the template.

3.4 Deployment

A demo of our hate speech detection system is hosted on a CMU-Q Unix virtual machine and is accessible at the following URL[3]. The Django-based demo uses Gunicorn to serve HTTP requests to the Unix Apache Server. The code is maintained and stored in a Git repository, and large files (e.g., the model and tokenizer) are tracked using Git Large File Storage.

4 METHODOLOGY

In this section, we describe the backend side of our hate speech detection system as well as the method we used to collect and annotate the ADHAR hate speech corpus. We also report on the experiments and evaluation work conducted to determine the most suitable ML model for our system.

Table 1: Statistics about ADHAR corpus.

Category	Hate	Not Hate	Total
Nationality	583	508	1,091
Religious beliefs	514	536	1,050
Ethnicity	541	520	1,061
Race	513	522	1,035

[3]https://adhar.qatar.cmu.edu/hatespeech/

4.1 ADHAR Corpus Overview

In (Charfi et al., 2024), we presented a hate speech corpus for Arabic called ADHAR, which includes different Arabic dialects alongside MSA, the standardized form of the Arabic language. To accommodate the linguistic diversity within the Arabic language, we arranged the data collection process according to regions with similar dialectal characteristics. We focused on four main regions: Egypt, the Levant (including Palestine and Jordan), the Gulf, and the Maghreb (including Morocco, Algeria, Libya, and Tunisia). The corpus addresses various categories of hate speech, including nationality, religion, race, and ethnicity, with instances from all covered dialects.

To ensure comprehensive representation, we carefully collected a minimum of 1,000 sentences for each category. Furthermore, to ensure balance among dialects, we carefully curated the dataset to include a minimum of 200 sentences for each category-dialect combination. Within these 200 sentences, we ensured equal distribution of hate speech and not-hate speech instances, with 100 sentences dedicated to each category-dialect combination. Overall, our corpus consists of 70,369 words and 4,237 sentences.

The corpus was collected manually from Twitter using seed keywords for each hate speech category. Seed keywords included dialectal hate words, general insults, and slurs specific to each dialect. This methodology allowed us to gather a high number of hateful examples. Furthermore, seed keywords included words exclusive to each dialect commonly used in daily conversations, to enhance the ability to identify posts specifically written in the corresponding dialect. These words stated basic meanings, such as "What", "Like this", "I want", "Seriously", etc.

Our research team included native Arabic speakers and experts in multiple dialects who annotated the collected sentences in our ADHAR corpus following three stages. The initial stage involved annotating sentences as either "Hate" or "Not Hate" within the respective categories of hate speech. The second stage involved annotating sentences as either "Hate," "Not Hate," or "Discuss" if the annotator found that the sentence was unclear or unrelated to the specified category. If the annotations did not match, a meeting involving all three members was held to discuss such cases. Then, the sentence was either removed, relabeled, or moved to a different category.

Table 1 shows the number of sentences in our ADHAR corpus and their distribution per class and category. Each entry in the corpus contains the following columns for the sentence: ID, Source, Dialect, Hate/Not-Hate label, and Hate Speech Category.

4.2 Experiments and Model Evaluation

We carried out a series of machine learning tests to evaluate our corpus for the task of hate speech detection in dialectal Arabic. Our models take the Arabic text along with the respective dialect as input and return a binary classification: either 'hate' or 'not-hate', as well as the corresponding hate speech category, namely nationality, religious beliefs, ethnicity, or race. To pre-process the text data, we removed URLs, emails, stop words, punctuation, and non-Arabic characters.

We used various feature extraction techniques, including word n-grams, character n-grams, TF-IDF, and word embeddings. Word n-grams were defined with an n-gram range of (1,3), including unigrams, bigrams, and trigrams, while character n-grams had a range of (2,5).

We proceeded by training a diverse array of machine learning models, including classical and neural network classifiers, to determine the best fit for our classification task. We employed classic classifiers such as Support Vector Machines (SVM), Logistic Regression (LR), Random Forest (RF), and Decision Trees (DT), followed by the Multi-Layer Perceptron (MLP) and Convolutional Neural Network-Bidirectional Long Short-Term Memory (CNN-BiLSTM) models. Furthermore, we used the pre-trained language model AraBERT, which is based on Google's Bert architecture and specifically tailored for Arabic text processing.

Table 2 displays the evaluation results for the different classifiers using our ADHAR corpus for hate speech label detection (hate or not hate), as well as hate speech category detection (i.e., religion, ethnicity, race, nationality). These classifiers were evaluated using four distinct feature sets: TF-IDF, n-grams, and character-level features were used for SVM, LR, RF, DT, and MLP classifiers, whereas word embeddings were used for the CNN-BiLSTM model.

As shown in Table reftab:Evaluation1, the MLP classifier performed best in detecting hate speech label using n-gram character feature, with an accuracy and F1-score of 89%. Logistic Regression with character n-grams performed best for detecting the hate speech category, with an accuracy and F1-score of 93%.

On the other hand, the AraBERT model demonstrated superior results for these tasks on the ADHAR corpus, achieving an F1-score and accuracy of 94% for hate speech label detection and an F1-score and accuracy of 95% for hate speech category detection, as shown in Table 3.

Table 2: Performance for hate speech label and category detection using different classifiers.

Classifier	Features		Hate speech label detection		Hate category detection	
			Accuracy	F1-Score	Accuracy	F1-score
Support Vector Machine	n-grams	Char	0.83	0.83	0.89	0.89
		Word	0.74	0.73	0.64	0.65
	TF-IDF		0.87	0.87	0.90	0.90
Logistic Regression	n-grams	Char	0.88	0.88	**0.93**	**0.93**
		Word	0.85	0.85	0.85	0.86
	TF-IDF		0.87	0.87	0.90	0.90
Random Forest	n-grams	Char	0.83	0.83	0.90	0.90
		Word	0.80	0.79	0.84	0.85
	TF-IDF		0.83	0.83	0.85	0.85
Decision Tree	n-grams	Char	0.75	0.75	0.85	0.85
		Word	0.79	0.79	0.82	0.83
	TF-IDF		0.77	0.77	0.81	0.81
Multi-Layer Perceptron	n-grams	Char	**0.89**	**0.89**	0.91	0.91
		Word	0.88	0.88	0.89	0.89
	TF-IDF		0.87	0.87	0.88	0.88
CNN-BiLSTM	Word embeddings		0.87	0.86	0.87	0.87

Figure 3: Input of Hate Speech Detection System.

Table 3: Performance for hate speech label and category detection using AraBERT.

Class	Precision	Recall	F1-score
Hate	0.96	0.93	0.94
Not Hate	0.92	0.95	0.93
		macro F1	0.94
		Accuracy	0.94
Nationality	0.97	0.92	0.94
Religion	0.96	0.98	0.97
Race	0.95	0.96	0.96
Ethnicity	0.94	0.96	0.95
		macro F1	0.95
		Accuracy	0.95

5 USE CASE SCENARIO

The aim of our hate speech detection system is to automatically detect and classify instances of hate speech within Arabic text, with the goal of enhancing online safety and fostering inclusive digital environments. While our system demonstrated very good performance when tested with data from our ADHAR dataset, we sought to validate its effectiveness more by testing it on external real-world data.

To accomplish this, we selected the ”Druze” faith, which is a religious group in the Levant as a topic of interest and we collected a dataset containing 100 random sentences about that topic, primarily sourced from X (formerly Twitter). Some examples of the collected tweets along with their translation to English are shown in Table 4. The collected tweets about this topic were written in either of the dialect groups (Gulf Region, Levant Region, Egyptian, Maghreb) or in MSA. Moreover, we ensured diversity in the collected sentences, including both ”hate” and ”not-hate” categories. Then, we evaluated the effectiveness and accuracy of our system by testing it on these sentences.

To use our system, the user needs to type or paste the sentence that they need to analyze, specify the corresponding Arabic dialect and then press the button *analyze* as shown in Figure 3. This will trigger our system to invoke the ML models and generate a prediction based on the input. Once a prediction is generated, the result will be displayed on the Web interface with a red font for ”hate”, and a green font for ”not hate” as shown in Figure 4.

In addition, the user has the option to upload a

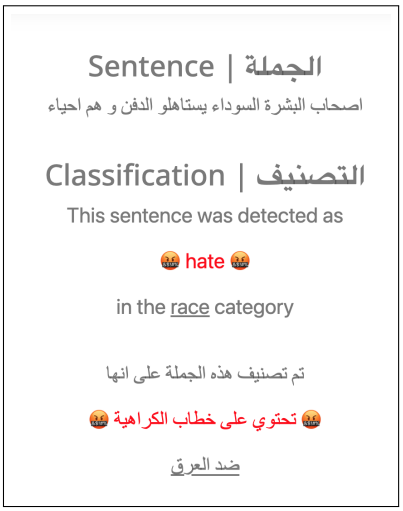

Figure 4: Output of Hate Speech Detection System.

Table 4: Examples of the sentences collected about the Druze faith.

Region's dialect	Tweet	Translation	Hate label
Levant Region		The Druze are infidel dogs, pigs, agents of the Jews. This video shows the Druze welcoming their soldiers returning from the war fronts	Hate
Egyptian		Our Druze brothers believe that the end of the world will begin in Egypt, from the dam	Not Hate
Gulf		Druzes do not fast	Not Hate
MSA		The Druze will remain Druze, People of treachery and hypocrisy, May God curse them and purify the Levant of their treachery, meanness and hypocrisy	Hate

CSV file with multiple sentences. This file should include two columns that must be labeled "Sentence" and "Dialect". Our system will then generate a CSV output file with four columns: the two input columns and two additional columns: the first is labeled as "Hate speech prediction", which includes the hate prediction for each sentence, and the second column is labeled as "Predicted Category" where the predicted hate category is shown.

Moreover, in case a file is provided as input, our system enables users to visualize the predicted hate speech detection results using pie charts. Specifically, our system produces two interactive pie charts: the first illustrates the distribution of hate predictions and shows the number of sentences per each label when hovering over each section, while the second pie chart shows the distribution of hate categories across the input dataset as well as the number of sentences per hate speech category when hovering over each section.

To test our system, we put the collected 100 sentences about the Druze faith in a CSV file and uploaded it to our system, which generated a new CSV file with the hate speech predictions for the input sentences as well as 2 pie charts. The first pie chart illustrates the distribution between the hate labels, i.e., hate vs non-hate sentences as shown in Figure 5. The second pie chart shows the distribution of hate speech categories as shown in Figure 6.

For this real-world scenario, Figure 5 shows that almost half the sentences are classified as hate sentences with 51 sentences detected as containing hate speech vs 49 sentences detected as not hateful. As shown in Figure 6, the most detected hate category is the religion-related hate speech with 48 out of the 51 hateful sentences related to religion.

It is noteworthy that the data used in this scenario is external to our system. It is neither part of the training set nor the testing set that we used to develop our model. In order to assess the results returned by our system on this external data, we manually annotated the collected sentences and then compared the manual annotations with the system's predictions. The results showed that 95 sentences out of the 100 sentences were correctly classified, while 5 sentences were misclassified in terms of their hate label. This is in line with the 0.94 accuracy reported in our experiments for hate speech label detection presented in Table 3. This demonstrates the high accuracy of our hate speech detection system even when used on external data from other domains.

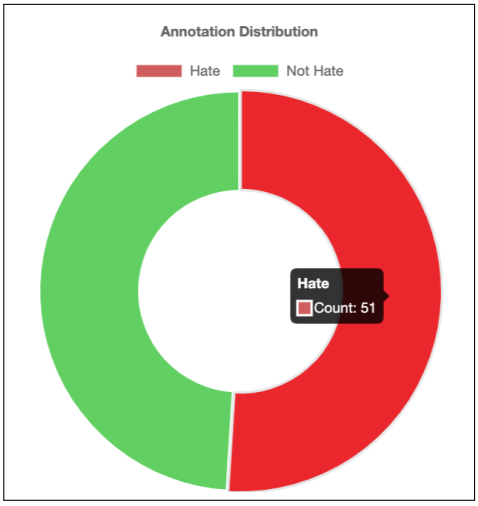

Figure 5: Pie chart of the hate labels distribution for the Druze topic.

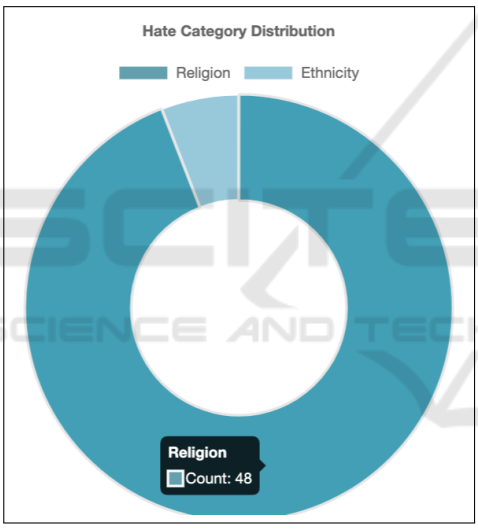

Figure 6: Pie chart of the hate category distribution for the Druze topic.

6 CONCLUSION

In this paper, we presented a Web-based system for hate speech detection in Arabic. Our system takes a sentence in Arabic along with the respective Arabic dialect. The proposed system employs the pre-trained language model AraBERT, fine-tuned on our ADHAR hate speech corpus for Dialectal Arabic, which we developed in a previous work. This AraBERT-based model exhibits superior performance compared to various other machine learning models tested in our experiments. The used corpus ADHAR includes over 4000 tweets and covers several Arabic dialects and four classes of hate speech namely nationality, religion, ethnicity, and race. As output, our system detects if the input sentence contains hate speech or not. If hate speech is detected, our system also outputs the specific hate speech category. Our system can also handle a CSV file as input and generate a CSV as output with the hate speech detection results for each sentence included as well as pie chart visualizations to better illustrate the distribution of hate vs not-hate sentences and the distribution of hate speech across the different hate speech categories.

As part of our future work, we will explore providing the capabilities of our system as an API that can be integrated with third-party systems. For instance, our hate speech detection API could be used in the furture to detect hate speech in comments on Websites where users can post messages. We also plan to collect more data belonging to other real-world scenarios to evaluate our system further.

ACKNOWLEDGEMENTS

This publication was made possible by NPRP13S-0206-200281 from the Qatar National Research Fund and by the generous support of the Qatar Foundation through Carnegie Mellon University in Qatar's Seed Research program. The contents herein reflect the work and are solely the authors' responsibility.

DISCLAIMER

Due to the nature of this work, some examples contain hate speech or offensive language. This does not reflect the authors' opinions by any mean. We hope this work can help in detecting and preventing spread of hate speech.

REFERENCES

Ahmed, A., Ali, N., Alzubaidi, M., Zaghouani, W., Abdalrazaq, A. A., and Househ, M. (2022). Freely available arabic corpora: A scoping review. *Computer Methods and Programs in Biomedicine Update*, 2:100049.

Al-Ibrahim, R. M., Ali, M. Z., and Najadat, H. M. (2023). Detection of hateful social media content for arabic language. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(9).

Almaliki, M., Almars, A. M., Gad, I., and Atlam, E.-S. (2023). Abmm: Arabic bert-mini model for hate-speech detection on social media. *Electronics*, 12(4).

Antoun, W., Baly, F., and Hajj, H. (2020). Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Charfi, A., Besghaier, M., Akasheh, R., Atalla, A., and Zaghouani, W. (2024). Hate speech detection with adhar: A multi-dialectal hate speech corpus in arabic. *Frontiers in Artificial Intelligence*, 7:1391472.

Chaudhari, A., Parseja, A., and Patyal, A. (2020). Cnn based hate-o-meter: A hate speech detecting tool. In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 940–944. IEEE.

Husain, F. (2020). Osact4 shared task on offensive language detection: Intensive preprocessing-based approach. *arXiv preprint arXiv:2005.07297*.

Magnossão de Paula, A. F., Rosso, P., Bensalem, I., and Zaghouani, W. (2022). UPV at the Arabic hate speech 2022 shared task: Offensive language and hate speech detection using transformers and ensemble models. In Al-Khalifa, H., Elsayed, T., Mubarak, H., Al-Thubaity, A., Magdy, W., and Darwish, K., editors, *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 181–185, Marseille, France. European Language Resources Association.

Mostafa, A., Mohamed, O., and Ashraf, A. (2022). Gof at arabic hate speech 2022: breaking the loss function convention for data-imbalanced arabic offensive text detection. In *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 167–175.

Mubarak, H., Darwish, K., Magdy, W., Elsayed, T., and Al-Khalifa, H. (2020). Overview of osact4 arabic offensive language detection shared task. In *Proceedings of the 4th Workshop on open-source arabic corpora and processing tools, with a shared task on offensive language detection*, pages 48–52.

Mubarak, H., Hassan, S., and Chowdhury, S. A. (2022). Emojis as anchors to detect arabic offensive language and hate speech.

Schmidt, A. and Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In Ku, L.-W. and Li, C.-T., editors, *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Vrysis, L., Vryzas, N., Kotsakis, R., Saridou, T., Matsiola, M., Veglis, A., Arcila-Calderón, C., and Dimoulas, C. (2021). A web interface for analyzing hate speech. *Future Internet*, 13(3):80.

Watanabe, H., Bouazizi, M., and Ohtsuki, T. (2018). Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. *IEEE access*, 6:13825–13835.

Zaghouani, W. (2017). Critical survey of the freely available arabic corpora. *CoRR*, abs/1702.07835.