Journal of Information Systems and Telecommunication
Vol.14, No.4, July-September 2026, 268-276
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Hybrid Embedding and GPT-5 Instruction-Augmented Framework for Arabic Hate Speech Detection

Anas Khudhur Abbas al juboori[1], Heshaam Faili[2*]

[1]. Computer Engineering, Department Pardis Alborz, University of Tehran, Tehran, Iran
[2]. School of Electrical and Computer Engineering, College of Engineering, University of Tehran , Tehran, Iran

Received: 10 Oct 2025/ Revised: 04 Sep 2025/ Accepted: 07 Oct 2026

Abstract
Hate speech is defined as language that creates hatred, hostility, or violence towards people or groups based on their religion, ethnicity, gender, and/or nationality. As social media continues to grow and expand, the influx of hate speech and abusive language on these platforms is on the rise and has become a significant threat to online safety and social cohesion. Despite improvements in the detection of hate speech with automation technologies, there is little research conducted on the detection of hate speech in Arabic, especially regarding varying dialects. Current approaches often fail to identify language patterns or contextual nuances that are specific to different dialects. Moreover, little research has been conducted focused solely on instruction-based learning in this field. We presented a methodology for detecting abusive language and hate speech in Arabic utilizing the L-HSAB database. Through the combination of AraBERT-v2 and Multilingual MPNet, hybrid embeddings were created that have both language-specific characteristics and cross-linguistic conceptual characteristics. UMAP and HDBSCAN were applied for both dimension reduction and cluster analysis, and iterative refinement of clustering was facilitated by the utilization of Silhouette and Coherence scores. Based on the identified resulting clusters generated, GPT-5 was used to produce linguistically grounded instructional prompts, which were then assessed by both LLM-as-a-Judge using prompt generation and by human validators. Finally, the identified instructions were incorporated into further fine-tuning of the models. Experimental results indicate that the proposed method yields 90.5% accuracy for the purpose of detecting hate speech in dialects of Arabic.

Keywords: Hate Speech; Large Language Model; Supervised Fine Tuning; Arabic.

1- Introduction

The rapid growth of the internet and social media has reshaped global communication, with over 5.56 billion internet users and 5.24 billion social media users worldwide in 2025. These platforms have become central to digital discourse, but alongside positive engagement, they have also facilitated the spread of harmful content such as hate speech and abusive language, often targeting sensitive attributes like religion, ethnicity, or gender [1]. In response, regulatory initiatives such as the EU hate speech code seek to address the issue, yet automated detection remains difficult due to the massive scale of online content and limitations of existing tools [2]. Although Arabic ranks among the most widely used online languages with more than 362 million native speakers [3], research on hate speech detection has largely focused on high-resource languages like English and Spanish, and Arabic receives far less attention [4].

Arabic hate speech detection remains underexplored, especially in dialects like Levantine. Existing resources are scarce and mostly centered on Modern Standard Arabic (MSA), which does not capture the informal, dialectal language of social media. Moreover, traditional methods struggle with noisy, imbalanced data and overlook linguistic cues, creating a need for dialect-specific and context-aware approaches [5,6]. The challenges related to the detection of Arabic hate speech have motivated the methodological decisions made in this research study. The first challenge pertains to the vast diversity of dialects in the Arabic language. Moreover, social media users often switch between Arabic and another language. Problems with

✉ **Corresponding Author**
hfaili@ut.ac.ir

having only one representation model and this linguistic diversity exist in that such a model may never encapsulate all of the pertinent semantic meaning and context. As a result, hybrid embedding can be used to combine Arabic-only embeddings, as well as multilingual embeddings (i.e., English, French, Spanish, etc.), in order to address this problem. The second challenge is the amount of noise and heterogeneity present in social media data, resulting in a reduced ability of traditional, supervised methods to detect (i.e., accurately identify) hate speech. To address this issue, clustering techniques are employed to identify the underlying semantic structure of social media data and cluster similar instances together prior to formal training of the model. The third challenge is that instruction-based learning has been shown to improve both understanding and generalization of tasks using LLMs recently. However, instruction-based learning has not yet been adequately explored for the problem of hate speech detection in Arabic. Thus, a unified model that utilizes hybrid embeddings, clustering, and instruction-based fine-tuning should be developed to meet the requirements of dialectical Arabic. In this study, we aim to develop a robust methodology for hate speech detection, leveraging hybrid embeddings, adaptive clustering, and instruction-based fine-tuning to address the limitations of existing approaches. Our research contributions can be summarized as follows:

- We focus on Levantine Arabic, one of the least studied yet widely spoken Arabic dialects, by employing the L-HSAB dataset, a benchmark corpus specifically designed for abusive and hate speech detection.
- We propose a hybrid embedding strategy that combines AraBERT-v2 with Multilingual MPNet, enabling both language-specific and cross-lingual contextual understanding.
- We introduce an adaptive clustering (UMAP + HDBSCAN) with iterative validation (Silhouette and Coherence scores), and leverage clustering outputs for the generation of linguistically grounded instructions via GPT-5.
- We develop a dual-layer evaluation framework (LLM-as-a-Judge + human review) to validate and filter high-quality instructions.
- We demonstrate that instruction-augmented Supervised Fine-Tuning (SFT+INST) substantially enhances performance in detecting hate speech in Arabic.

The remainder of this paper is structured as follows: Section 2 reviews the related works. Section 3 details the data and methodology employed in our study. Section 4 provides an evaluation of our results. Finally, Section 5 highlights our work's key contributions and suggests future research directions.

2- Related Works

Several studies have addressed hate speech detection in Arabic, each focusing on different aspects of the problem. Doghmash and Saad [7] approached Arabic hate speech detection and masking on social media. They used the SemEval-2020 Arabic offensive language dataset (10,000 tweets) to compare deep learning architectures to transformer-based models for the task of hate speech detection. While traditional models achieved reasonable accuracy of ~80%, and simpler transformer models were shown to outperform them, the QARiB model utilizing pre-trained AraBERT was able to achieve 92% macro F1 and 95% accuracy to surpass all prior results. In addition to hate speech detection, the authors also proposed hate speech masking as a machine translation task, where they generated a parallel corpus that replaced the offensive term with a masked token. Their transformer-based model was evaluated with BLEU scores of 0.30 (1-gram).

Salloum et al. [8] studied Arabic hate speech in the context of Twitter using the L-HSAB dataset, focusing on the Levantine dialects. The methodology incorporated text preprocessing, TF-IDF representation, dimensionality reduction using PCA, and clustering with K-Means. This study grouped tweets into meaningful clusters, demonstrating intrinsic properties of hate speech within the data that might not be visible to unsupervised analysis alone.

Ahmed et al. [9] examined the barriers to detecting hate speech within Arabic, which encompasses the dialect continuum across Syria, Jordan, Palestine, and Lebanon, using the L-HSAB dataset. Their experiments comparing the various embedding methods showed that models trained on Arabic, like fastText and TF-IDF, performed considerably better than English embeddings, such as GloVe and GoogleNews Word2Vec, which performed near zero.

Charfi et al. [10] proposed the ADHAR dataset in response to the identified gaps in previous Arabic hate-speech resources, which included either a single dialect or a significant imbalance in the proportions of hate-speech. ADHAR contains a total of 4,240 tweets from five different dialects that were annotated for hate speech as binary categories and target sources (nationality, religion, ethnicity, race) with a high degree of annotator agreement. Their experimental results showed that traditional models performed reasonably well, but fine-tuned AraBERT performed best in the detection of hate (≈0.94 F1).

Daouadi et al. [11] found Arabic hate-speech detection on Twitter useful to describe the task as a five-class classification problem (non-hate, general hate, racial hate, religious hate, sexist hate), annotated tweets. Their dataset included 11,634 tweets that they had annotated. Their approach was to fine-tune AraBERT and MARBERT and then ensemble those models. They attempted to address the imbalance by semi-supervised augmentation, combining externally labeled religious hate tweets and applying pseudo-labeling on an additional corpus using their best-performing model. The results produced in the paper showed ensembling

provided better performance than individual models, with the permutation producing a weighted F1 of ~85.6, with significant improvement in performance in minority classes of religious and sexist hate.

Aldjanabi et al. [12] introduced a Multi-Task Learning (MTL) framework utilizing pre-trained Arabic language models to improve performance on tasks related to offensive and hate speech in Arabic. Unlike single-task models, their MTL approach learns from multiple datasets simultaneously, allowing the model to capture both global and dataset-specific representations. They utilized several benchmark datasets, including OSACT-OFF, OSACT-HS, L-HSAB, and T-HSAB, and modified AraBERT and MarBERT for this purpose. The results showed that their MTL framework outperformed traditional single-task models, particularly with MarBERT, which better addressed dialectal variations in Arabic.

Research on Arabic hate speech detection can be divided into three major areas. The first area presents transformer models and fine-tuning methods as a means of improving performance [7,11,12]. Recent results demonstrate that Arabic language models (e.g., AraBERT, MarBERT) deliver strong results when using supervised methods. However, while these methods offer an approach to leverage the semantic structure of language to enhance model performance, they do not utilize the latent semantic structure of the data for their classification. The second area focuses on feature extraction and embedding approaches [9], demonstrating the importance of utilizing representations specific to the Arabic language to capture language characteristics. These investigations emphasize comparing different embedding methods but do not consider using complementary representations. The third area includes research focused on building datasets and conducting exploratory analysis of dialectical hate speech [8,10]. This area of research provides information and resources concerning Arabic dialects but utilizes traditional clustering techniques and representations.

While progress continues in each of these areas, there are still a number of gaps in the available literature. First, there have not yet been sufficient attempts made to develop both Arabic and multilingual equivalent representations to capture both dialect-specific and contextual-based semantic representations. Clustering methods have primarily been used for exploratory analysis and are not used as a guide for adapting downstream models. Finally, instruction-based learning, which has shown promise across various NLP tasks, remains a largely unexploited approach to detecting hate speech in Arabic. This study seeks to address these gaps by integrating hybrid embeddings, adaptive clustering, and instruction generation to form an integrated model for detecting hate speech in Levantine Arabic.

3- Methodology

We build our work on the integration of hybrid embeddings, adaptive outlier detection, and clustering with iterative validation to obtain multilingual instruction prompts that combine unsupervised clustering with SFT. The steps of our proposed approach are depicted in Figure 1. In this study, we performed a multiclass classification task utilizing the L-HSAB dataset, with the objective of categorizing tweets into three distinct classes: normal, hate, or abusive. As a first task, we created hybrid representations of text by combining AraBERT-v2 [13] with a multilingual MPNet model [14], taking advantage of both a language-specific and cross-lingual contextual features. We then utilized UMAP [15] to reduce the dimensionality of embeddings before clustering with HDBSCAN [16], a hierarchical density-based clustering algorithm. To validate the clustering process, we used silhouette and coherence scores to improve the reliability of the clusters as we tuned parameters until we achieved semantically similar clusters. From our identified clusters, we created sample instructions that provided linguistic to the Large Language Model (LLM) that distinguished between the target classes in a range of ways. Our created prompts were critically evaluated using automated LLM-based judgement and human judgement, and we retained the best prompts. We utilized the provided dataset and valid instructions to fine-tune the classification model, and we evaluated its performance using standard evaluation metrics of accuracy, precision, recall, and F1-score, improving upon baseline methods.

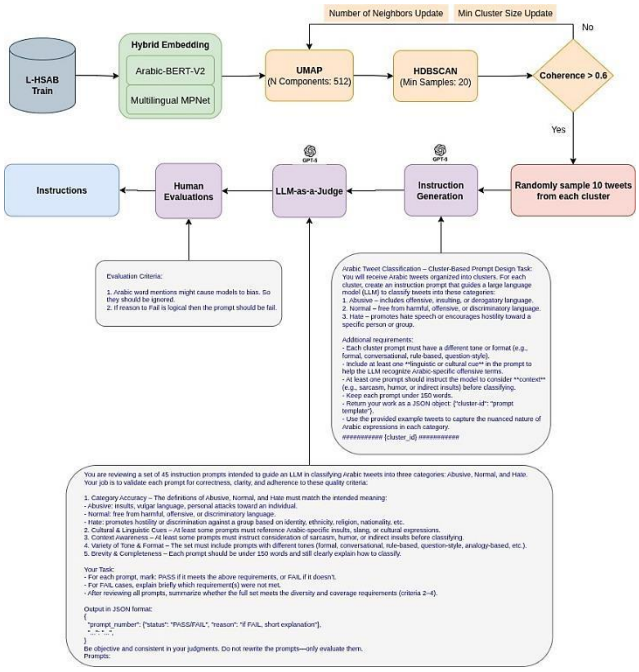


Fig. 1. The steps of our proposed approach.

3-1- Dataset

We utilized the L-HSAB dataset (Levantine Hate Speech and Abusive Language) [17], which is one of the first publicly available datasets aimed at hate speech detection for Levantine Arabic. L-HSAB is composed of tweets, gathered from Twitter and annotated into three categories: Normal, Abusive, and Hate. Table 1 shows the L-HSAB dataset statistics. L-HSAB is split into a training set and a testing set; the training set contains 1,369 abusive tweets, 2,968 normal tweets, and 339 hate tweets, which gives a total of 4,676 samples. In the testing set, there are 359 abusive, 682 normal, and 129 hate tweets, which is a total of 1,170 samples. As a whole, L-HSAB is an excellent starting point for research on hate speech and abusive language in Arabic. The strength of the dataset is that it specifically focuses on the Levantine dialect, which is less represented in existing multilingual corpora.

Table 1: Dividing the L-HSAB dataset.

Split	Abusive	Normal	Hate	Total
Train	1,369	2,968	339	4,676
Test	359	682	129	1,170

3-2- Hybrid Embedding Construction

To generate semantically enriched representations of the tweets, we employed a hybrid embedding strategy that integrates two complementary transformer-based models. The first model, AraBERT-v2, is specifically pre-trained on large-scale Arabic corpora and is optimized to capture morphological and syntactic properties unique to Arabic text. The second model, Multilingual MPNet-v2, is a multilingual sentence transformer designed to encode cross-lingual semantic similarity across diverse languages.

We chose both AraBERT-v2 and Multilingual MPNet due to their complementary abilities to represent dialectal Arabic content. AraBERT-v2 has been pretrained on a large scale using Arabic corpora, achieving exceptional results on multiple Arabic NLP tasks, whereas Multilingual MPNet has been designed specifically for language-agnostic semantic representations of multiple languages and can generalize contextually better than AraBERT-v2. Therefore, when using both models together, definitions of the Arabic language will include unique linguistic features as well as semantics that will allow for broader semantic relationships than what AraBERT would offer alone. We concatenated the embeddings from each model so that we could preserve all of the information generated from both models. If we were to average the embeddings of both models, we would force each model's embeddings into one common space and potentially diminish some of the features that pertain only to the dialect; thus, by concatenating the embeddings, we have preserved the unique features that were learned from each encoder. The end result of combining the two models created a representation with 1,536 features; however, the dimensionality reduction phase that followed reduced the computational cost while maintaining the most semantically informative structure for cluster analysis.

For each tweet, embeddings were independently computed using both models. These vectors were then horizontally concatenated (stacked) to form a unified feature vector that retains the strengths of both representations. AraBERT-v2 ensures coverage of Arabic-specific linguistic nuances, while Multilingual MPNet enriches contextual depth by embedding the tweet within a multilingual semantic space. This contextual enhancement strategy was designed to overcome the limitations of relying on a single embedding model, enabling the system to capture fine-grained semantic variations in Arabic tweets. The resulting hybrid embeddings served as the foundation for downstream clustering, validation, and classification tasks.

This process ultimately resulted in the creation of a large set of candidate instructions, from which only the most semantically coherent and linguistically rich prompts were retained for evaluation in the subsequent stage.

We used feature concatenation because it does not add extra fusion parameters and allows us to preserve all the information obtained from each encoder. Weighted fusion, on the other hand, needs pre-defined or learned weights, while concatenation makes use of the unique characteristics produced by both AraBERT-v2 and Multilingual MPNet at its original level. This feature allows the model to make use of the linguistic patterns in the Arabic language and enjoy some multilingual features as well. Although the use of concatenation creates a 1,536-dimensional feature vector, the usage of UMAP leads to a dimensionality reduction method that is efficient enough to compress the vector, taking into account its essential semantic structure for clustering.

3-3- Dimensionality Reduction and Clustering Validation

Given the high dimensionality of the hybrid embeddings, UMAP was first applied for dimensionality reduction. UMAP preserves local and global semantic structures while transforming the high-dimensional embeddings into a lower-dimensional space, thereby improving clustering efficiency. The complete parameter configurations used in UMAP and HDBSCAN, including their initial values, adjustment ranges, and final optimal values, are summarized in Table 2. The initial UMAP configuration included *n_neighbors=10*, *n_components=512*, and cosine similarity as the distance metric.

The rationale for using n_components = 512 was to maintain meaningful semantic information from 1,536-dimensional hybrid embeddings while reducing dimensionality for

improved clustering performance. Preliminary testing has demonstrated that larger amounts of dimensionality reduction have resulted in less coherent clusters. During refinement only n_neighbors and min_cluster_size were modified, as those two parameters have the most significant impact on building local neighborhoods and the degree of cluster granularity. All other parameters were held constant with the intention of decreasing the complexity of searches and maintaining consistency across iterations. A coherence threshold of 0.6 is to be utilized as an acceptable measure of semantic interpretability, meaning enough difference exists within the cluster content before being used to create instructional materials.

Following dimensionality reduction, we applied HDBSCAN for unsupervised clustering. HDBSCAN was chosen over K-Means due to its ability to automatically determine the number of clusters and to effectively handle heterogeneous datasets with varying density. Importantly, HDBSCAN assigns noisy or ambiguous data points as outliers (*label = -1*), ensuring that tweets without a clear semantic grouping are isolated rather than forcefully assigned to inappropriate clusters. The initial parameters for HDBSCAN included *min_cluster_size = 20*, *min_samples = 20*, *metric = euclidean*, and *cluster_selection_method = eom*.

To ensure semantic reliability of the clusters, we adopted a dynamic validation and refinement procedure. After each clustering run, two metrics were computed:

I) The Silhouette Score, which measures intra-cluster cohesion and inter-cluster separation, and
II) The Coherence Score, which evaluates the semantic interpretability of cluster contents.

If the coherence score of any cluster fell below 0.6, the clustering process was automatically re-executed with adjusted parameters. Specifically, *min_cluster_size* in HDBSCAN was randomly varied between *20* and *100*, while *n_neighbors* in UMAP was randomly varied between *10* and *200*. This process ensured that the clustering was iteratively refined until coherence thresholds were met.

The iterative optimization process stopped when all clusters reached the necessary coherence level of 0.6 or, in the case there were no changes after each algorithm optimization. During the optimization process, the parameters that were changed were only n_neighbors and min_cluster_size, as they were the only parameters that significantly changed the neighborhood maintenance process and contributed to the granularity level of the cluster, while other parameters were fixed to ensure the controlled and reproducible nature of the optimization process.

During experimentation, the initial run produced a Silhouette Score of *-0.156* and a Coherence Score of 0.452, indicating low semantic quality. After *52* iterations of parameter adjustment and re-clustering, the process converged to a solution with a Silhouette Score of *-0.018* and a Coherence Score of *0.636*, exceeding the required coherence threshold of *0.6*.

The final clustering results consisted of six coherent clusters alongside a substantial number of outliers (2,617 tweets), which were assigned to a separate group. Among the clusters, Cluster 0 was the largest, containing 1,049 tweets. These clusters, along with the outlier group, were subsequently used to guide instruction generation in later stages.

Table 2: Parameter settings for UMAP and HDBSCAN during clustering and refinement

Method	Parameter	Initial Value	Adjustment Range	Final Optimal Value
UMAP	n_neighbors	10	10 – 200	116
	n_components	512	–	512
	Metric	cosine	–	cosine
HDBSCAN	min_cluster_size	20	20 – 100	84
	min_samples	20	–	20
	Metric	euclidean	–	euclidean
	cluster_selection_method	eom	–	eom

Figure 2 illustrates the variation of Silhouette and Coherence scores across the 52 clustering iterations. Although it is true that the final Silhouette Score was nearly zero, this does not mean that the quality of clustering was poor. The characteristics of the sample datasets resulted in significant heterogeneity in the form of social media content, where normal, abusive, and hate speech have overlapping language constructs. In these types of environments, density-based clustering techniques also tend to produce lower silhouette scores due to there being no clear semantic distinction across different category types. Therefore, coherence scores were used as the primary metric for optimizing clustering, as it provides a measure of how semantically consistent the content found in each cluster is with each other. It should be noted that there wasn't a pre-established number of refinement iterations (52 iterations); this number was derived through the iterative optimization process through which each run satisfied the predetermined coherence level for the dataset in question. While the Silhouette score consistently remained around zero due to the heterogeneous and noisy nature of the dataset, the Coherence score showed significant improvement after repeated parameter adjustments, eventually exceeding the 0.6 threshold. This progression demonstrates the effectiveness of the iterative refinement strategy in achieving semantically meaningful clusters despite the challenging characteristics of the data.

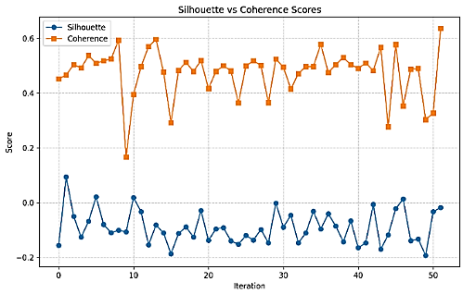


Fig. 2 Silhouette and Coherence scores across 52 clustering iterations.

3-4- Instruction Generation

Following the clustering and refinement stage, the next task was to create a series of instruction prompts for the classification model. To achieve this, a random sampling selection approach was used so that we randomly selected ten representative random samples of tweets belonging to each cluster. This ensured that the generated instructions captured the semantic diversity of each group while avoiding overfitting to a narrow set of examples. Outliers identified by HDBSCAN references were also grouped and considered in the instruction design as border ambiguous cases.

With the instructions generated from the sampled clusters, the next step was to use GPT-5 to assist with the instruction generation. GPT-5 was given the random sampling of clusters and was instructed to provide classification instructions designed to allow detection of Normal, Abusive, and Hate. Instructions were varied so that there was exploration of the various styles and directions the model was trained to guard against acceptance or overfitting to a single instruction type. The generated instructions included formal descriptions, conversational phrases, rule-based guidelines, and question-based directives, ensuring a broad coverage of styles.

Each instruction contained at least one linguistic element specific to Arabic, such as common abusive terms, culturally sensitive expressions, or region-specific slang. This was critical to expand the model's contextual understanding and limit the marginal ability to recognize hate speech or abusive speech spoken in Arabic. To maintain clarity and conciseness, all instructions were no more than 150 words, striking a balance between informativeness and brevity. This process ultimately resulted in the creation of a large set of candidate instructions, from which only the most linguistically rich and semantically coherent prompts were preserved for evaluation in the next phase.

3-5- Instruction Evaluation

The initial pool of instructions generated by GPT-5 was subjected to a rigorous multi-stage evaluation pipeline to ensure their reliability and effectiveness. The first phase was the LLM-as-Judge phase, where an LLM automatically assessed each instruction based on several predefined dimensions:

I) Category Accuracy or the degree to which the instruction correctly guides the model to differentiate between normal, abusive, and hate speech.

II) Linguistic cues, which measure the extent to which the instruction integrates relevant Arabic-specific abusive or hate-related expressions.

III) Contextual Awareness, or the degree to which the instruction addresses the nuance of meaning and borderline cases.

IV) Variation of Tone and Format, or variety of style across instructions; and,

V) brevity and completeness, confirming that the prompts remained under 150 words while being sufficiently informative.

Hence, automatically flagged instructions underperforming for one or more of these dimensions were discarded.

In the second stage, human evaluation was conducted to further refine the set of candidate instructions. Human annotators cross-verified the automated results and particularly focused on identifying instructions that contained explicit Arabic terms or culturally sensitive expressions that could lead to hallucinations or misinterpretations when used by the model. This step was crucial to prevent over-reliance on potentially misleading lexical markers.

To ensure that the evaluation process is as reliable as possible, a three-member team of independent annotators (all experienced in Arabic including linguistic analysis) conducted the human review stage. Each annotator provided their own review of the created instructions based on three major criteria (linguistic relevance, appropriateness to context, and conformity to the intended target category). Those that were either ambiguous, misleading, or incompatible with the intended category were discarded. Very little difference of opinion among the annotators occurred; the level of inter-annotator agreement was calculated using Fleiss' kappa coefficient ($\kappa = 0.78$), which demonstrates a substantial level of agreement between the evaluators. In cases where there were differences of opinion, the final decision on which instruction(s) to discard was made through discussion and majority agreement.

Through this two-tiered evaluation, the pool of candidate instructions was narrowed down to 21 high-quality prompts. To balance the dataset across categories, a subset of these instructions was selected: two were randomly assigned to the abusive category, while seven were retained for the hate category, ensuring that underrepresented classes received greater instructional coverage. The final selection was guided by performance metrics observed in preliminary zero-shot evaluations, where instructions were ranked according to their F1-score per class shown in Table 3. For example, instruction IDs [17, 19, 30, 35, 37, 42, 44] were chosen for the hate class based on their superior F1

performance, while instruction ID 36 was retained as the best performer for the abusive class. This procedure ensured that the final instruction set was not only diverse but also empirically validated for its effectiveness in guiding the classification model.

For the sake of hallucinations and biased output reduction, GPT-5 was making instructions only on the basis of tweets collected in each cluster and without using any other information. The instructions were evaluated using the framework of LLM-as-a-Judge according to their consistency regarding the category, relevance regarding the language, accuracy, and completeness. Finally, three separate human annotators checked the instructions and eliminated biased and misleading information.

Table 3: Ranking of instructions based on their F1-score in each class

	IDx	Abusive (F1-score)	Hate (F1-score)	Normal (F1-score)	Accuracy	Precision	Recall	F1-score
0	6	0.284069	0.29199	0.583732	0.420513	0.490735	0.50977	0.386597
1	8	0.485066	0.282828	0.62692	0.517949	0.483632	0.502789	0.464938
2	10	0.13964	0.278571	0.585227	0.390598	0.451822	0.482136	0.334479
3	14	0.49172	0.291262	0.682415	0.549573	0.503684	0.524856	0.488466
4	17	0.365482	0.296073	0.618215	0.463248	0.493004	0.517731	0.42659
5	19	0.437318	0.302285	0.613825	0.486325	0.493491	0.524255	0.451142
6	21	0.169565	0.289373	0.604186	0.413675	0.451872	0.490469	0.354374
7	23	0.458947	0.267606	0.518672	0.448718	0.482437	0.471891	0.415075
8	25	0.426554	0.290456	0.636522	0.501709	0.471005	0.499968	0.451177
9	30	0.514076	0.309859	0.671233	0.561538	0.502821	0.528698	0.498389
10	31	0.270073	0.275253	0.53	0.382906	0.463091	0.479884	0.358442
11	33	0.234615	0.275401	0.595149	0.412821	0.454409	0.478703	0.368389
12	34	0.51234	0.248996	0.445415	0.45812	0.499667	0.45929	0.40225
13	35	0.462703	0.314869	0.593284	0.500855	0.481938	0.49366	0.456952
14	36	0.528195	0.193548	0.614679	0.54188	0.511816	0.471156	0.445474
15	37	0.366492	0.322004	0.692053	0.523932	0.498211	0.534352	0.460183
16	39	0.366883	0.278003	0.520855	0.403419	0.486163	0.496193	0.38858
17	40	0.527596	0.19802	0.524793	0.483761	0.485903	0.463502	0.416803
18	42	0.405998	0.313646	0.509165	0.429915	0.464166	0.484573	0.409603
19	43	0.474895	0.253165	0.602996	0.503419	0.476112	0.47151	0.443685
20	44	0.46472	0.302521	0.652885	0.533333	0.4802	0.502119	0.473376

3-6- Model Training and Evaluation

LLAMA-3-8B was chosen for the final phase of SFT because it has an strong ability to follow instructions and competitive performance across in multilingual natural language understanding. The curated set of instructions amassed earlier was coupled with the labeled L-HSAB dataset during instruction-augmented supervised fine-tuning (SFT+INST) procedure. Through this coupling of the annotated examples with linguistically based instructions, the model learned to make task-specific distinctions while creating richer contextual representations.

4- Results and Discussion

The proposed approach was evaluated on the L-HSAB dataset using the SFT + INST. As shown in Table 4, the model achieved an accuracy of 90.5%, precision of 91.8%, recall of 85.3%, and an F1-score of 87.9%. These results confirm that

the combination of hybrid embeddings, adaptive clustering, and instruction-based guidance produces a competitive model.

For this paper, the results we report were derived from the official L-HSAB train-test data splits. Caution should be exercised when making direct comparisons between this work and previous studies since each may have used different data splits, preprocessing techniques, model configurations, and evaluation protocols. Therefore, the comparison shown in Table 4 was developed for general performance comparison purposes (not to be used as a strictly controlled benchmark).

We compared our proposed implementation with the study by Al-Jarrah et al. [18]. They explored an ensemble CNN-Bi-LSTM architecture and several transformer-based models. Their best results in the multi-class setting were obtained using Multilingual BERT, which achieved an F1-score of 80.0%, precision of 81%, recall of 79%, and accuracy of 81%. In contrast, our method substantially outperforms these results, reflecting a more balanced trade-off between precision and recall.

Additionally, Aldjanabi et al. [12] employed a cross-corpora multi-task learning framework incorporating datasets such as OSACT and L-HSAB, finding that the MarBERT-based multi-task model achieved an accuracy of 87.46% and an F1-score of 87.18% on these tasks. Although their multi-task setup improved over single-task methods, our single-task instruction-augmented model still edges ahead in both accuracy and F1-score, indicating the effectiveness of instruction-guided fine-tuning in dialectal contexts.

The performance enhancement noted is not only illustrated by the evaluation metrics but also by the ability of the presented model to deal with the major language hurdles that Arabic dialects face. The problems of key lexicon, spelling variation, code switching, and a low number of annotated texts are caused by the informal nature of this Arabic dialect. The hybrid embedding approach merges the benefits of AraBERT-v2 in terms of the representation of Arabic dialects, as it has an express understanding of the features related to the Arabic language, with multilingual semantic representations applied to the training of the Multilingual MPNet. The adaptive clustering organizes semantically similar tweets into the same group, which minimizes the negative influence of heterogeneous data. Hence, GPT-5 has more coherent information about the context. Therefore, this analysis shows how the fine-tuning method based on instructions allows the model to learn not only about the lexical features but also about the understanding of context and semantics, which are required for identifying hate speech in the dialectal varieties of Arabic.

Table 4: Evaluation and comparison of the results.

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Al-Jarrah et al. [18] (Multilingual BERT)	81	81	79	80.0
Aldjanab et al. [12] (MarBERT)	87.46	-	-	87.18
Our Implementation (SFT + INST)	**90.5**	**91.8**	**85.3**	**87.9**

5- Conclusion

The new method developed in this research for identifying hate speech in Arabic employs the L-HSAB dataset. The new system uses AraBERT-v2 and a multilingual MPNet hybrid embedding, followed by adaptive clustering using UMAP and HDBSCAN. Clusters produced as a result were employed in the creation of instructions as part of SFT+INST. These instructions were created using the GPT-5 model and later verified using LLM-as-Judge and human reviewers. The resulting SFT+INST model achieved an accuracy of 90.5%, proving that merged hybrid semantic representations and instruction-based mechanisms used proved to be efficient in identifying hate speech in the Levantine Arabic dialect.

Despite achieving impressive results, several shortcomings must be highlighted. Firstly, instruction generation relies heavily on GPT-5, which might lead to model-specific biases. As a result, the computation cost of this system considerably rises, reducing accessibility in resource-constrained environments. Additionally, the clustering step produced several outliers, indicating that a part of the dataset could not be organized into a coherent group semantically. Thus, the performance of the system evidently heavily depends on the clustering stage. Another point to mention is that the suggested method has been tested using only one dataset of Levantine Arabic L-HSAB. Therefore, it is still unclear how effective this method might be in the case of application to other dialects of Arabic, multilingual cases, or different moderation instances.

Future research will focus on the context of the suggested model in relation to other Arabic dialects, and larger datasets will be examined as well. It would also be compulsory to replace the GPT-5 with Smaller Language Models (SLMs), which would come at a lower computational cost and ensure similar performance. Extending the mechanism to more content moderation aspects, like offensive and abusive language, etc., is also a promising perspective for research.

References

[1] K. Gasmi et al., "Hybrid Feature and Optimized Deep Learning Model Fusion for Detecting Hateful Arabic Content," IEEE Access, 2025.

[2] K. E. Daouadi, Y. Boualleg, and O. Guehairia, "Systematic investigation of recent pre-trained language model for hate speech

detection in arabic tweets," ACM Transactions on Asian and Low-Resource Language Information Processing, 2024.
[3] B. Corporation, "Most Spoken Languages in the World," 2025. [Online]. Available: https://www.berlitz.com/blog/most-spoken-languages-world.
[4] F. Husain and O. Uzuner, "A survey of offensive language detection for the arabic language," ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), vol. 20, no. 1, pp. 1-44, 2021.
[5] M. M. Abdelsamie, S. S. Azab, and H. A. Hefny, "The dialects gap: A multi-task learning approach for enhancing hate speech detection in Arabic dialects," Expert Systems with Applications, vol. 295, p. 128584, 2026.
[6] A. Dahou et al., "A Survey on Dialect Arabic Processing and Analysis: Recent Advances and Future Trends," ACM Transactions on Asian and Low-Resource Language Information Processing, 2025.
[7] S. T. Doghmash and M. Saad, "Arabic Hate Speech Identification and Masking in Social Media using Deep Learning Models and Pre-trained Models Fine-tuning," arXiv preprint arXiv:2507.23661, 2025.
[8] S. Salloum, K. Tahat, A. Mansoori, R. Alfaisal, and D. Tahat, "Leveraging K-Means Clustering for Analysis of Arabic Hate Speech Tweets," in Global Congress on Emerging Technologies (GCET-2024), 2024: IEEE, pp. 282-285.
[9] A. H. Ahmed, R.-J. Yew, X. Minocher, and S. Venkatasubramanian, "Navigating Dialectal Bias and Ethical Complexities in Levantine Arabic Hate Speech Detection," arXiv preprint arXiv:2412.10991, 2024.
[10] A. Charfi, M. Besghaier, R. Akasheh, A. Atalla, and W. Zaghouani, "Hate speech detection with ADHAR: a multi-dialectal hate speech corpus in Arabic," (in English), Frontiers in Artificial Intelligence, Review vol. Volume 7 - 2024, 2024-May-30 2024, doi: 10.3389/frai.2024.1391472.
[11] K. E. Daouadi, Y. Boualleg, and K. E. Haouaouchi, "Ensemble of pre-trained language models and data augmentation for hate speech detection from Arabic tweets," arXiv preprint arXiv:2407.02448, 2024.
[12] W. Aldjanabi, A. Dahou, M. A. Al-Qaness, M. A. Elaziz, A. M. Helmi, and R. Damaševičius, "Arabic offensive and hate speech detection using a cross-corpora multi-task learning model," in Informatics, 2021, vol. 8, no. 4: MDPI, p. 69.
[13] W. Antoun, F. Baly, and H. Hajj, "Arabert: Transformer-based model for arabic language understanding," arXiv preprint arXiv:2003.00104, 2020.
[14] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," arXiv preprint arXiv:1908.10084, 2019.
[15] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," arXiv preprint arXiv:1802.03426, 2018.
[16] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," J. Open Source Softw., vol. 2, no. 11, p. 205, 2017.
[17] H. Mulki, H. Haddad, C. B. Ali, and H. Alshabani, "L-hsab: A levantine twitter dataset for hate speech and abusive language," in Proceedings of the third workshop on abusive language online, 2019, pp. 111-118.
[18] H. Al-Jarrah, M. Al-Smadi, M. Hammad, and F. Shannaq, "Using Deep Learning Techniques to Detect Hate and Abusive Language in Arabic Tweets," International Journal of Intelligent Engineering & Systems, vol. 17, no. 5, 2024.