Detecting Hate Speech Targeting Protected Groups in Arabic Using Hypothesis Engineering and Zero-Shot Learning with Ground Validation via ChatGPT

Ahmed Fat'hAlalim, Yongjian Liu, Qing Xie, Alhag Alsayed, Musa Eldow
School of Computer Science and Artificial Intelligence,
Wuhan University of Technology, Hubei, Wuhan, China

***Abstract*—Automatic detection of hate speech in low-resource languages presents a persistent challenge in natural language processing, particularly with the rise of toxic discourse on social media platforms. Arabic, characterized by its rich morphology, dialectal variation, and limited annotated datasets, is underrepresented in hate speech research, especially regarding content targeting marginalized and protected groups. This study proposes a zero-shot learning approach that leverages Natural Language Inference (NLI) models guided by carefully engineered hypotheses in native Arabic to detect hate speech against protected groups, such as women, immigrants, Jews, Black people, transgender individuals, gay people, and people with disabilities. We formulated nine different Arabic hypothesis groups and employed a zero-shot XNLI model with a baseline embedding-based model, incorporating preprocessing techniques on the HateEval Arabic dataset. The results indicate that the XNLI model achieves up to 80% accuracy in detecting targeted hate speech, significantly outperforming baseline models. Furthermore, a real-world validation using GPT-3 via the ChatGPT interface achieved 54% accuracy in zero-shot conversational settings. These findings highlight the importance of hypothesis design and linguistic preprocessing in zero-shot hate speech detection, particularly in low-resource and culturally nuanced languages offering a scalable and culturally aware solution for moderating harmful content in Arabic online spaces.**

***Keywords*—Hate speech detection; low resource Arabic language, zero-shot learning; natural language processing; ChatGPT; transfer learning; online safety**

I. Introduction

The proliferation of hate speech on digital platforms, particularly social networks, has raised serious social concerns due to the harm it causes marginalized communities. Hate speech refers to any type of expression that involves discrimination against individuals or groups based on their identity, such as race, ethnicity, religion, gender, sexual orientation, and other factors [1]. It can range from offensive discourse targeting inherent traits to speech, gestures, or physical expressions that threaten individuals or groups. However, distinguishing between hate speech and merely offensive or controversial opinions remains complex, especially in contexts where freedom of expression is a sensitive issue [2]. In response, researchers have increasingly turned to Natural Language Processing (NLP) techniques to automate the detection and classification of hate speech.

Despite advancements in NLP and Machine Learning, hate speech detection continues to present significant challenges [3], particularly in low-resource languages Arabic, which presents unique complexities due to its linguistic characteristics, including the complex morphology of the language, and the wide array of dialects spoken across different regions [4], [5]. Recently, Many scientific studies have used machine learning and deep learning to automatically detect hate speech [6], However Traditional supervised learning approaches depend heavily on annotated datasets, which are often scarce or nonexistent for languages such as Arabic [7], [8]. Additionally, these approaches need help to adapt to the dynamic and context-dependent nature of hate speech, resulting in suboptimal performance in real-world situations. Therefore, there is an urgent need to develop efficient methods to address this problem. One promising approach involves reusing natural language inference (NLI) models for text classification, which has shown promising results in zero- and few-shot classification tasks [9]. Recent research by Goldzycher and Schneider [10] has also highlighted the potential for zero-shot NLI-based settings to outperform traditional few-shot fine-tuning approaches in English. While prior studies have explored the identification of hate speech in several languages, such as English, there remains a need for specialized approaches that target protected groups within these languages. This highlights the importance of investigating novel methodologies, such as hypothesis engineering for zero-shot learning, to address these challenges and advance hate speech detection capabilities across diverse linguistic contexts.

The objective of this study is multifaceted, aiming to address several key challenges in hate speech detection within the context of low-resource languages. Firstly, we propose to develop a set of hypotheses tailored to the characteristics of hate speech targeting protected groups in low-resource languages, such as Arabic. These hypotheses will serve as the basis for our zero-shot learning approach, facilitating the model's ability to generalize to unseen instances of hate speech. Secondly, we seek to create a scenario-based framework for hate speech detection, wherein the model is trained to recognize nuanced forms of hate speech directed towards specific protected groups, including women, minorities, and marginalized communities. By incorporating scenario-based training data, we aim to enhance the model's sensitivity to context and improve its performance in real-world applications. Additionally, we plan to conduct ground validation experiments using a chat-based interface, such as GPT, to assess the practical effectiveness of our proposed approach. This iterative validation process will provide insights into the model's performance in natural language interactions, further

validating its utility in real-world settings.

In addition to the contribution of this study to detecting hate speech in low-resource languages, it also conducts in-depth experiments on carefully selected hypotheses that fit the rich nature of the Arabic language, which is characterized by an abundance of synonyms and extensive linguistic dictionaries. Furthermore, we explore these hypotheses in scenarios in which hate speech is directed toward protected groups in Arab societies. These contributions can be summarized as follows:

- We developed a comprehensive set of hypotheses in Arabic specifically tailored to detect hate speech.
- We proposed a novel approach using zero-shot learning to detect hate speech in low-resource Arabic settings, guided by meticulously formulated hypotheses considering contextual and linguistic challenges.
- Our study presents an enhanced methodology employing zero-shot learning for detecting hate speech in Arabic, targeting protected groups such as women, immigrants, Jews, Black people, transgender individuals, gay people, and disabled people.
- Through ground validation using ChatGPT, we demonstrated the practical usability of our hypotheses in real-time conversations, verifying their potential for proactive moderation.

The organization of this paper is as follows: Section II presents an overview of previous research conducted in identifying hate speech. Section III provides a detailed explanation of the technique used in this study, while Section IV presents the experimental setup, which includes the building of the model, the division of the data, and the metrics used for evaluation. Section V outlines the methods utilized in our investigation, Our results are presented in Section VI, while Section VII provides the discussion of the findings. In Section VIII we analyze the top errors of our study, conclude the work, and suggest possible directions for future research in Section IX.

II. Related Work

Current developments in automated hate speech detection move from machine learning models to the use of deep learning and transformer-based models, a comprehensive review by Abdelsamie et al. [11] discuss the latest techniques in natural language processing for hate speech detection in Arabic, including Lexicon-based, ML, DL, and transformer-based models. Each of these approaches addresses the complexity of the Arabic language in a different way. ML models like Support Vector Machines (SVM) combined with word embeddings have shown high accuracy in classifying offensive content in Arabic tweets [12], [13]. Convolutional neural networks (CNNs) and their hybrid models, such as CNN-LSTM and BiLSTM-CNN, have been effective in binary, ternary, and multi-class classification tasks for hate speech detection [8], [14]. Lexicon-based approaches, which involve creating specialized lexicons of offensive terms, have been employed to identify and classify hate speech, especially in the context of religious hatred [15]. The impact of tokenization strategies and vocabulary sizes on the performance of Arabic language models in downstream natural language processing tasks has also been examined [16]. Sentiment analysis techniques are also utilized to capture the meaning of Arabic words and classify tweets as hateful or non-hateful. The integration of genetic algorithms with classifiers like XGBoost and SVM has been used to optimize hyperparameters and improve detection accuracy [12].

Pre-trained word embedding models like AraVec and fastText, fine-tuned on specific datasets, have proven beneficial in capturing the semantic nuances of Arabic hate speech [17]. Additionally, Elmadany et al. explore the use of affective bidirectional transformers for offensive language detection in Arabic, demonstrating the utility of training models on sentiment and emotion data to enhance performance [18]. Daouadi et al. introduce an ensemble approach that combines pre-trained language models and data augmentation to improve hate speech detection from Arabic tweets, achieving encouraging results. Their methodology addresses the issues of limited performance and imbalanced data, common challenges in Arabic hate speech detection [19].

A. Zero-Shot Learning Approaches

Zero-shot learning has emerged as a promising approach for hate speech detection, particularly in scenarios with limited labeled data and high variability across languages and contexts. Research indicates that ZSL can effectively leverage large language models, such as T5 and BLOOM, to achieve performance comparable to traditional fine-tuned models, even in under-resourced languages [20], [21]. Techniques like hypothesis engineering enhance ZSL by combining multiple predictions to improve accuracy, demonstrating significant gains over standard models [10]. Furthermore, Goldzycher et al. [22] employed fine-tuned models based on the XNLI dataset to evaluate the effectiveness of NLI models in detecting hate speech across languages. Experiments were conducted in Arabic, Hindi, Italian, Portuguese, and Spanish, with multilingual models initially adapted for detecting hate speech in English and further refined using language-specific data. Further research by Zia et al. explores zero-shot cross-lingual hate speech detection, highlighting the effectiveness of pseudo-label fine-tuning of transformer language models in improving detection performance across different languages [23]. These advancements highlight the potential of ZSL to address the challenges of hate speech detection across diverse linguistic landscapes.

B. Research Gap

Despite significant advancements in hate speech detection, several critical gaps persist, particularly concerning low-resource languages like Arabic. The complexity of Arabic, characterized by diverse dialects and rich vocabulary, poses substantial challenges for traditional supervised learning approaches that rely heavily on large annotated datasets. Moreover, existing research predominantly focuses on resource-rich languages, leaving a significant void in developing effective detection models tailored for Arabic.

Traditional deep learning and transformer-based models, while powerful, often require extensive labeled data for training, which is scarce for Arabic [24]. This scarcity hampers the development of robust hate speech detection systems for

Arabic-speaking communities. Furthermore, there is a noticeable lack of research on hypothesis engineering specifically tailored to zero-shot learning frameworks for Arabic hate speech detection. Previous studies have also insufficiently addressed hate speech targeting protected groups within the Arabic-speaking community, such as women, immigrants, and religious minorities.

ZSL emerges as a promising alternative, enabling models to generalize to new tasks without task-specific training data. By leveraging ZSL, it's possible to overcome the limitations posed by data scarcity, allowing for the development of hate speech detection models that are both accurate and adaptable to the nuances of the Arabic language and its dialects. This approach not only reduces the dependency on large annotated datasets but also facilitates the rapid deployment of detection systems across different contexts and communities.

III. Methodology

The methodology adopted in this study follows a structured, systematic approach to detect hate speech targeting protected groups in the Arabic language utilizing zero-shot learning techniques, presented in Fig. 1. The process is initiated by formulating hypotheses specifically tailored to the Arabic language. These hypotheses are designed to encapsulate various facets of hate speech, making them suitable for a zero-shot learning approach. Subsequently, these hypotheses are subjected to initial experiments using the XNLI model, which employs zero-shot learning, and several preprocessing techniques are applied to the Arabic text data, ensuring the text is clean, consistent, and ready for analysis. Following the initial experiments, the best-performing hypotheses are selected and refined to improve their effectiveness in detecting hate speech. This step is crucial for tailoring the detection system to the specific linguistic and cultural nuances of Arabic. A comparative analysis is then conducted to evaluate the performance of the XNLI model with the zero-shot learning approach against the embedding baseline model. Finally, the refined hypotheses are tested in real-life conversations using a chat-based interface with GPT-3.

A. Zero-Shot Learning in Hate Speech Detection

Traditional zero-shot learning methods rely on providing a descriptor or information about an unseen class [25]. This descriptor can be in the form of visual attributes, the name of the class, or any other relevant information. By providing this descriptor, the model can make predictions for the unseen class even without having any training data specifically for that class. In other words, the model uses the provided information to generalize and recognize the characteristics of the unseen class. This approach enables the model to extend its knowledge beyond the classes it has been trained on and make accurate predictions for new and previously unseen classes.

The objective of Zero-Shot Learning is to learn a model (f) that maps instances (x) and auxiliary information (a) to class labels (y). Mathematically, this can be expressed as:

$$f : (x, a)ßy \tag{1}$$

where x represents an input instance from the dataset, a represents the auxiliary information associated with each class, and y represents the class label.

To train the Zero-Shot Learning model, a loss function (L) is defined to measure the discrepancy between the predicted class labels and the ground truth labels. The loss function guides the model to minimize the classification error during training. Mathematically, the loss function can be represent as:

$$L(f(x, a), y) \tag{2}$$

where L represents the loss function, $f(x, a)$ represents the predicted class label for instance x based on the auxiliary information a, and y represents the ground truth class label for instance x.

The key aspect of Zero-Shot Learning is its ability to generalize to unseen classes. During inference, the model can predict the class labels for instances belonging to new classes that were not present in the training data. This is achieved by using the learned relationships between the auxiliary information and the class labels. Mathematically, the generalization can be expressed in Eq. 3.

$$f(x\_new, a\_new)ßy\_new \tag{3}$$

where x_new represents a new instance from an unseen class, a_new represents the auxiliary information associated with the new class, and y_new represents the predicted class label for the new instance.

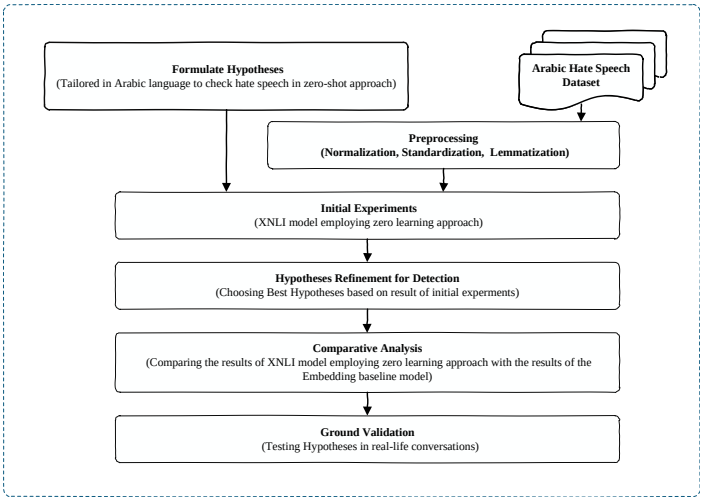

Fig. 1. An Illustration of the methodology used in this study to detect hate speech targeting protected groups in Arabic highlights the essential steps.

B. HateCheck Dataset

The HateCheck [2] is a meticulously curated resource designed to evaluate the performance of hate speech detection models. It encompasses a wide variety of hate speech examples, targeting diverse protected groups, and is structured to test models across multiple dimensions of hate speech. This

includes explicit and implicit hate speech, different types of hate (e.g. racism, sexism, homophobia), and varying intensities and forms of hateful expressions. The dataset is notable for its comprehensive coverage, which aims to mimic the complexity and variability of hate speech encountered in real-world settings. In this study, we utilized a subset of HateCheck specifically adapted for Arabic, known as HateCheck Arabic, which was instrumental in validating the effectiveness of our zero-shot learning methodology for detecting hate speech in Arabic. This dataset has been painstakingly annotated to identify hate speech and provides valuable insights into the prevalence and nature of offensive content in the Arabic-speaking context. Table I contains details of the Hatecheck-Arabic dataset statistics in terms of size, sub-groups targeted by hate speech, and hate statements percentages.

TABLE I. STATISTIC OF HEATCHECK-ARABIC DATASET

Class/Target	Size	Hate statements (%)	Not hate statements (%)
Women	534	406 (76%)	128 (24%)
immigrants	437	333 (76%)	104 (24%)
Jews	437	333 (76%)	104 (24%)
black_people	485	369 (76%)	116 (24%)
trans_people	437	333 (76%)	104 (24%)
gay_people	509	387 (76%)	122 (24%)
disabled_people	437	333 (76%)	104 (24%)
No Class	294	0 (0%)	294 (100%)
HateCheck-arabic	3570	2494 (70%)	1076 (30%)

IV. EXPERIMENTAL SETUP

In this section, we provide an overview of the experimental setup, detailing the model employed, the split of the Arabic HateCheck dataset, and the performance metrics used to assess the model's efficacy.

A. Model Selection

Within the model selection, we systematically explored our zero-shot learning approach's effectiveness in the Arabic language context. To achieve this, we employed two distinct models to evaluate a set of hypotheses meticulously formulated for our experiments. The first model entailed leveraging an embeddings-based approach, while the second model involved harnessing the XNLI model, tailored for hate speech classification. After comprehensive experimentation, we identified the most promising hypothesis that yielded the highest performance in hate speech detection using our methodology. Subsequently, we conducted an additional validation step using chatGPT, which we employed to test the accuracy of the best-performing hypothesis. This validation procedure allowed us to gauge our zero-shot learning approach's real-world applicability and robustness when integrated with advanced language models. The following is an explanation of the architecture of these models.

1) NLI Model: NLI (Natural Language Inference) models have gained prominence in various natural language processing tasks, including zero-shot topic classification [26]. NLI models are designed to determine the relationship between two given sentences: whether the second sentence contradicts, entails, or is neutral concerning the first sentence. Leveraging the capabilities of NLI models, zero-shot topic classification enables the classification of text into predefined topics or categories without explicitly training on labeled examples from those topics. By encoding the topic description as a premise and the input text as a hypothesis, NLI models can infer the topic relevance or compatibility. This approach proves particularly useful in scenarios where labeled data for all target topics is limited or unavailable. The NLI model's ability to generalize across topics makes it a promising choice for zero-shot topic classification tasks, including hate speech detection.

NLI is a task where the model is given a premise (P) and a hypothesis (H) and is required to predict the relationship between them, typically as entailment, contradiction, or neutral [27]. This can be represented mathematically in Eq. 4.

$$NLI(P, H) -> \{entailment, contradiction, neutral\} \quad (4)$$

XNLI is a specific variant of the NLI trained on the XNLI dataset, which is a multilingual natural language inference dataset. The methodology of XNLI involves fine-tuning the pre-trained XLM-RoBERTa-Large model on the XNLI task. It takes the premise (P) and hypothesis (H) as inputs and predicts the relationship between them.

Both NLI and XNLI methodologies involve training a model to understand the relationships between premises and hypotheses. The models are trained on large amounts of data to learn the semantic representations and context required for accurate inference. These methodologies enable the models to generalize well to new instances and perform effectively in various natural language understanding tasks, including textual entailment and inference-based classifications. For our experiment, we utilized the XNLI[1] model as the base model.

2) Embeddings: Embeddings play a critical role in NLP in numerically representing textual data while retaining semantic relationships and contextual information [28]. They convert words or phrases into high-dimensional vectors, making machine learning models more capable of grasping linguistic meanings and patterns. Mathematically, an embedding for a word w_i can be denoted as:

$$E(w_i) \quad (5)$$

where E represents the embedding function. For a given text S, it can be represented as a sequence of word embeddings:

$$S = [E(w_1), E(w_2), ..., E(w_n)] \quad (6)$$

where n signifies the length of the text. Zero-shot learning is used in conjunction with embeddings to improve the detection of hate speech. Because zero-shot learning allows models to generalize to previously unseen classes, it is useful for classifying hate speech directed at protected groups. The model learns to associate embeddings with specific hate speech categories by leveraging auxiliary information or hypotheses. For example, hypothesizing "this text contains hate speech targeting immigrants" directs the model to recognize instances of hate speech directed at immigrants. A similar approach for

[1] https://huggingface.co/joeddav/xlm-roberta-large-xnli

zero-shot topic classification was demonstrated by Yin et al. [9].

In our experimental approach, we used embeddings as a critical component of our analysis. We fine-tuned the Embedding model of OpenAI to classifying hate speech, specifically the text-embedding-ada-002I[2] version. We used embeddings to increase the depth of our investigation after developing and testing hypotheses using the XNLI model. By passing these hypotheses to the embedding model, we aimed to conduct a comprehensive comparative analysis between the two models. This approach enabled us to delve into the intricate nuances of hate speech detection, leveraging both the semantic relationships captured by embeddings and the cross-lingual understanding facilitated by the XNLI model. We hoped to select the most effective model for detecting hate speech directed at protected groups using this two-pronged approach.

3) GPT-3: Developed by OpenAI, represents a groundbreaking achievement in natural language processing (NLP). This state-of-the-art language model has garnered significant attention for its exceptional ability to generate coherent and contextually relevant human-like text across a wide array of tasks. Its advanced capabilities stem from extensive pretraining on vast datasets, allowing it to capture intricate language patterns and subtleties [29]. GPT-3 utilizes a transformer architecture featuring multiple attention mechanisms, enhancing the model's understanding of long-term dependencies in textual data. The self-attention mechanism, fundamental to its architecture, can be mathematically expressed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

Where, Q, K, and V are the query, key, and value matrices respectively, and d_k is the dimension of the key vectors. The softmax function scales the dot product of the query and key vectors by the square root of d_k. The resulting attention scores are then used to weight the value vectors, producing the attended representation.

The Attention function takes in three inputs: the query matrix (Q), the key matrix (K), and the value matrix (V). It also considers the dimension of the key vectors, represented as (d_k). The function first calculates the dot product of the query and key matrices, and then scales this result by the square root of the dimension of the key vectors. The softmax function is then applied to these scaled values, resulting in what are known as attention scores. These attention scores are subsequently used to weight the value vectors, yielding the final output, which is the attended representation. This process essentially allows the model to focus on different parts of the input sequence when producing the output.

B. Data Split

Our data was sourced from the Arabic HateCheck as we mentioned in Section III-B , which is composed of a wide variety of text samples that include hate speech directed at different protected groups. In order to enhance the resilience of our model, we executed a stratified split of the data, taking into account the groups targeted. Table I provides a detailed breakdown of these groups.

C. Model Performance Metrics

To comprehensively evaluate our hate speech detection system, we employed four widely recognized metrics that collectively assess different facets of model performance.

1) Accuracy (ACC): quantifies the model's overall correctness by calculating the percentage of all predictions that align with the true labels.

$$ACC = \frac{T_{Postive} + T_{Negative}}{T_{Postive} + T_{Negative} + F_{Postive} + F_{Negative}} \tag{8}$$

2) Precision (Pre): evaluates the reliability of positive predictions, emphasizing the model's ability to minimize false alarms.

$$Pre = \frac{T_{Postive}}{(T_{Postive} + F_{Postive})} \tag{9}$$

3) Recall (Rec): measures how effectively the model identifies all instances of hate speech within the dataset, prioritizing the detection of true positives.

$$Rec = \frac{T_{Postive}}{(T_{Postive} + F_{Negative})} \tag{10}$$

4) F1-score (F1-s): harmonizes precision and recall into a single metric, ensuring a balanced assessment of the model's performance even when class distributions are uneven.

$$F1 - s = \frac{2 * (Pre * Rec)}{(Pre + Rec)} \tag{11}$$

V. Methods

We describe the methods employed in our experiments, organized into the following three subsections that outline the key steps of our experimentation.

A. Hypothesis Generation, Initial Experiments, and Preprocessing

Guided by the hypothesis engineering proposed by Goldzycher et al. [10], We formulated our hypotheses in Arabic according to the proposed method in Fig. 2, where we formulated the hypotheses in the form of "It is / That text is / This example is / This example contains / This text is / This text contains / This is / Containing / Contains + hate speech / hate-inciting speech / provocative hate speech / hateful". Table II presents the hypotheses formulated in Arabic and their corresponding literal translations in English. The translations were generated using the chatGPT model.[3]

Following the development of these hypotheses, we conducted initial experiments using embeddings as a baseline in conjunction with the XNLI model. We utilized the XNLI

[2]https://platform.openai.com/docs/models/embeddings

[3]https://chat.openai.com/

Algorithm 1 Hate Speech Detection in Arabic

Ensure: Labels for each text sample in D'
Input: Dataset D, Hypotheses H, NLI Model XNLI, Threshold θ
Output: Labels for each text sample in D'
Preprocessing Steps:
$D \leftarrow \text{Normalize}(D)$ ▷ Convert text to a canonical form (e.g.,Unified number format, removing diacritics)
$D \leftarrow \text{RemoveNoise}(D)$ ▷ Remove unnecessary characters (e.g., punctuation, stop words)
$D \leftarrow \text{Lemmatize}(D)$ ▷ Reduce words to their base or root form
$D' \leftarrow D$ ▷ Final preprocessed dataset
for all $t \in D'$ **do** ▷ For each text sample t in the preprocessed dataset
  for all $h \in H$ **do** ▷ For each hypothesis h
    $S(t, h) \leftarrow \text{XNLI}(t, h)$ ▷ Calculate the semantic similarity score
    if $S(t, h) \geq \theta$ **then**
      $\text{Label}(t) \leftarrow$ Hate Speech
    else
      $\text{Label}(t) \leftarrow$ Non-Hate Speech
    end if
  end for
end for
Return Labels for each text sample in D'

model on the HatCheck dataset, inputting the formulated premises and hypotheses. This stage allowed us to evaluate the performance of the generated hypotheses and to compare the embeddings-based approach to the XNLI model.

Continuing from the initial experiments, the hypotheses showing promising results underwent further refinement through preprocessing using the Kurdish Language Processing Toolkit (KLT)[4] to preprocess the data. The toolkit served as a valuable resource for performing various language processing tasks specific to the Kurdish and Arabic languages. It offered a range of functionalities, including tokenization, stemming, and morphological analysis [30], which helped us effectively preprocess and prepare the data for further analysis, As shown in Fig. 2 Where we took the following steps on the dataset before sending it to model:

1) Normalization: involves handling diverse encodings and merging characters based on dialects and scripts

2) Standardization: takes a normalized text and produces a standardized version according to Arabic orthographies

3) Lemmatization: It involves determining the root form of a word and applying various morphological transformations, such as removing prefixes and suffixes, to obtain the lemma.

To improve data quality and consistency, techniques such as normalization and lemmatization were used. This preprocessing step aimed to alleviate the impact of dialectal variations and improve the overall performance of the model. As detailed in Algorithm 1, we systematically applied various preprocessing techniques to the dataset, which included tokenization, normalization, noise removal, and lemmatization. Following that we re-evaluated the refined hypotheses after preprocessing, enabling us to quantitatively measure the improvement achieved through these preprocessing techniques.

[4]https://github.com/sinaahmadi/klpt

TABLE II. THE HYPOTHESES FORMULATED IN ARABIC AND THEIR CORRESPONDING LITERAL TRANSLATIONS IN ENGLISH

	hypothesis	literal translation
P1-1	إنه حديث كراهية	It is hate speech
P1-2	إنه حديث يحض على كراهية	It is hate-inciting speech
P1-3	انه خطاب مثير للكراهية	It is provocative hate speech
P1-4	انه كراهية	It is hateful
P2-1	ذلك النص حديث كراهية	That text is hate speech
P2-2	ذلك النص حديث يحض على الكراهية	That text is hate-inciting speech
P2-3	ذلك النص خطاب مثير للكراهية	That text is provocative hate speech
P2-4	ذلك النص كراهية	That text is hateful
P3-1	هذا المثال به حديث كراهية	This example is hate speech
P3-2	هذا المثال به حديث يحض على كراهية	This example is hate-inciting speech
P3-3	هذا المثال به خطاب مثير كراهية	This example is provocative hate speech
P3-4	هذا المثال به كراهية	This example is hateful
P4-1	هذا المثال يحتوي على حديث كراهية	This example contains hate speech
P4-2	هذا المثال يحتوي على حديث يحض على كراهية	This example contains
P4-3	هذا المثال يحتوي على خطاب مثير كراهية	This example contains provocative hate speech
P4-4	هذا المثال يحتوي على كراهية	This example contains hateful
P5-1	هذا النص حديث كراهية	This text is hate speech
P5-2	هذا النص حديث يحض على كراهية	This text is hate-inciting speech
P5-3	هذا النص خطاب مثير كراهية	This text is provocative hate speech
P5-4	هذا النص كراهية	This text is hateful
P6-1	هذا النص يحتوي على حديث كراهية	This text contains hate speech
P6-2	هذا النص يحتوي على حديث يحض على كراهية	This text contains hate-inciting speech
P6-3	هذا النص يحتوي على خطاب مثير كراهية	This text contains provocative hate speech
P6-4	هذا النص يحتوي على كراهية	This text contains hateful
P7-1	هذا حديث كراهية	This is hate speech
P7-2	هذا حديث يحض على كراهية	This is hate-inciting speech
P7-3	هذا خطاب مثير كراهية	This is provocative hate speech
P7-4	هذا كراهية	This is hateful
P8-1	يحتوي على حديث كراهية	Containing hate speech
P8-2	يحتوي على حديث يحض على كراهية	Containing hate-inciting speech
P8-3	يحتوي على خطاب مثير كراهية	Containing provocative hate speech
P8-4	يحتوي على كراهية	Containing hateful
P9-1	يحوي حديث كراهية	Contains hate speech
P9-2	يحوي حديث يحض على كراهية	Contains hate-inciting speech
P9-3	يحوي خطاب مثير كراهية	Contains provocative hate speech
P9-4	يحوي كراهية	Contains hateful

B. Hypothesis Refinement and Subsetting by Protected Groups

To narrow down our focus and enhance the model's ability to detect hate speech targeting specific protected groups, we selected the two best-performing hypotheses from the refined pool. These selected hypotheses were then subjected to further experimentation. Experiments were conducted for each subset of the dataset representing protected groups, such as women, disabled people, trans people, etc. The same models, namely embeddings and XNLI, were utilized in these subsequent experiments as they were in the initial experiments. This enabled us to examine the effectiveness of the selected hypotheses in detecting hate speech that was directed toward specific

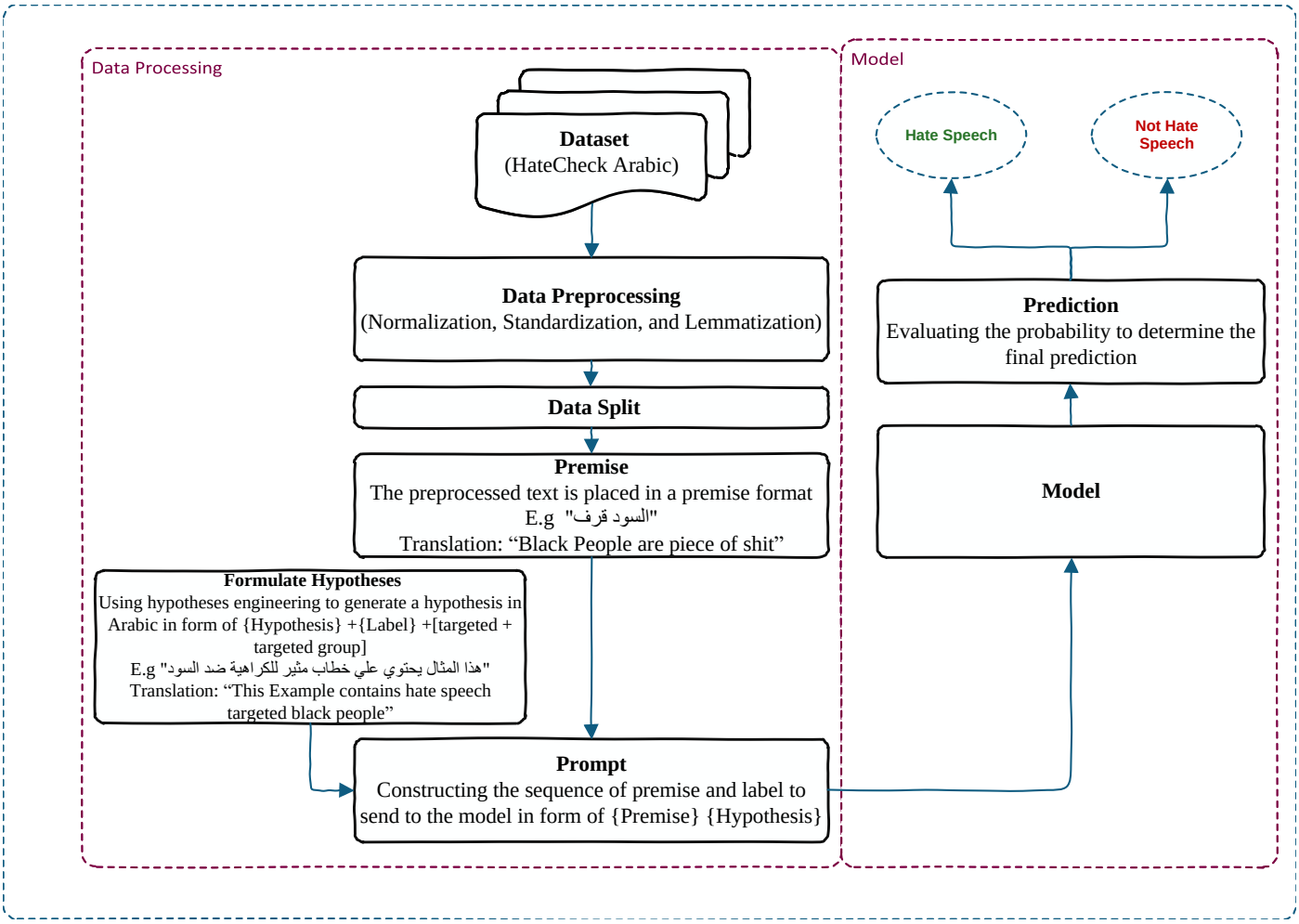

Fig. 2. Methodological framework for hate speech detection in Arabic targeting protected groups.

protected groups within the Arabic language.

C. Ground Validation Using GPT Chat

In the third phase of our methodology, detailed in Algorithm 2, we validated the effectiveness of the refined hypotheses in real-world conversational scenarios using the GPT-3.5 turbo model and the GPT chat interface from OpenAI. We input the hypotheses into the GPT chat interface to evaluate their practical relevance in detecting hate speech targeting protected groups in real-life conversations. The validation results were then compared to the outcomes from the initial experimental phase and the revised hypotheses following preprocessing. This comprehensive assessment facilitated the evaluation of performance enhancement achieved through hypothesis refinement and preprocessing techniques in the context of Zero-Shot Learning for hate speech detection. The comparison underscored the practical applicability and robustness of our approach in real-world settings.

VI. Results

A. Initial Experiments and Hypothesis Performance

Our initial experiment aimed to comprehensively assess the effectiveness of various hypotheses in the detection of hate speech. The hypotheses that are included in our study are presented in Table II. This table contains a total of nine main hypotheses, each of which is further divided into four sub-hypotheses. The hypotheses were carefully constructed to encompass the intricate features of hate speech. To assess the efficacy of these hypotheses, we utilized the XNLI model as our analytical instrument. The utilization of this model enabled the assessment of the efficacy of each hypothesis in accurately identifying hate speech within the particular context of our research. The results obtained from these experiments provide significant insight into the effectiveness of each hypothesis in comprehensively capturing the various manifestations of hate speech.

B. Impact of Preprocessing Techniques

Our systematic application of preprocessing techniques resulted in significant improvements in both data quality and model performance, albeit with notable differences between architectures. As shown in Table III, the embedding model showed a big increase in accuracy for detecting hate speech after preprocessing, indicating that normalizing features was crucial for improving its ability to recognize patterns. The performance data for the XNLI model in Table IV showed

Algorithm 2 Real-World Validation with GPT-3

Require: Preprocessed Dataset D', Hypotheses H, GPT-3 Model GPT-3
Ensure: Real-World Validation Accuracy A_{GPT3}
Input: Preprocessed Dataset D', Hypotheses H, GPT-3 Model GPT-3
Output: Real-World Validation Accuracy A_{GPT3}
Initialize: Correct Detections $C \leftarrow 0$, Total Validations $T \leftarrow 0$
for all $t \in D'$ **do**
  for all $h \in H$ **do**
    prompt $\leftarrow$ ConstructPrompt(t, h)
    response $\leftarrow$ GPT-3(prompt)
    if response = Hate Speech **then**
      Label$(t) \leftarrow$ Hate Speech
    else
      Label$(t) \leftarrow$ Non-Hate Speech
    end if
    $T \leftarrow T + 1$
    if Label(t) = Ground Truth Label(t) **then**
      $C \leftarrow C + 1$
    end if
  end for
end for
$A_{GPT3} \leftarrow \frac{C}{T}$ ▷ Calculate the real-world validation accuracy
Return Real-World Validation Accuracy A_{GPT3}

TABLE III. PERFORMANCE METRICS OF EMBEDDING MODEL BEFORE AND AFTER PREPROCESSING

	Experiment before preprocessing				Experiment after applying KLT			
Hypothesis	Pre	Rec	F1-S	Acc	Pre	Rec	F1-S	Acc
P1-1	0.56	0.55	0.55	0.71	0.65	0.52	0.49	0.76
P1-2	0.54	0.55	0.5	0.53	0.53	0.52	0.52	0.69
P1-3	0.54	0.52	0.51	0.72	0.72	0.52	0.48	0.77
P1-4	0.57	0.55	0.55	0.71	0.65	0.52	0.49	0.76
P2-1	0.54	0.54	0.54	0.67	0.62	0.53	0.52	0.76
P2-2	0.56	0.56	0.56	0.67	0.61	0.54	0.53	0.75
P2-3	0.53	0.53	0.52	0.59	0.55	0.53	0.53	0.71
P2-4	0.54	0.55	0.51	0.55	0.55	0.54	0.54	0.7
P3-1	0.53	0.54	0.49	0.53	0.53	0.51	0.5	0.71
P3-2	0.55	0.57	0.51	0.53	0.53	0.52	0.52	0.7
P3-3	0.54	0.54	0.54	0.65	0.61	0.54	0.53	0.75
P3-4	0.53	0.55	0.49	0.51	0.54	0.53	0.52	0.71
P4-1	0.5	0.5	0.36	0.36	0.49	0.49	0.45	0.49
P4-2	0.5	0.5	0.4	0.41	0.5	0.5	0.47	0.51
P4-3	0.53	0.54	0.52	0.58	0.52	0.52	0.52	0.64
P4-4	0.5	0.5	0.38	0.38	0.51	0.51	0.49	0.54
P5-1	0.53	0.54	0.53	0.61	0.56	0.53	0.51	0.74
P5-2	0.55	0.56	0.54	0.61	0.57	0.53	0.53	0.74
P5-3	0.53	0.54	0.53	0.63	0.57	0.54	0.53	0.73
P5-4	0.54	0.55	0.53	0.6	0.53	0.52	0.52	0.71
P6-1	0.5	0.5	0.34	0.34	0.49	0.49	0.43	0.45
P6-2	0.49	0.49	0.32	0.32	0.49	0.48	0.38	0.38
P6-3	0.52	0.53	0.5	0.56	0.52	0.52	0.51	0.61
P6-4	0.48	0.49	0.25	0.27	0.51	0.51	0.44	0.46
P7-1	0.54	0.53	0.53	0.7	0.62	0.53	0.5	0.76
P7-2	0.55	0.57	0.54	0.59	0.56	0.53	0.53	0.72
P7-3	0.57	0.54	0.53	0.73	0.71	0.53	0.5	0.77
P7-4	0.57	0.58	0.57	0.67	0.58	0.53	0.53	0.74
P8-1	0.5	0.5	0.32	0.33	0.51	0.51	0.41	0.41
P8-2	0.53	0.51	0.23	0.26	0.49	0.49	0.32	0.32
P8-3	0.49	0.48	0.41	0.43	0.5	0.49	0.46	0.49
P8-4	0.49	0.49	0.39	0.4	0.5	0.5	0.47	0.51
P9-1	0.52	0.51	0.32	0.32	0.52	0.53	0.47	0.49
P9-2	0.55	0.51	0.23	0.27	0.49	0.49	0.32	0.32
P9-3	0.51	0.52	0.41	0.42	0.52	0.52	0.52	0.64
P9-4	0.53	0.54	0.46	0.48	0.52	0.53	0.52	0.64

TABLE IV. PERFORMANCE METRICS OF THE XNLI MODEL BEFORE AND AFTER PREPROCESSING

	Experiment before preprocessing				Experiment after applying KLT			
Hypothesis	Pre	Rec	F1-S	Acc	Pre	Rec	F1-S	Acc
P1-1	0.72	0.66	0.59	0.58	0.72	0.57	0.56	0.54
P1-2	0.73	0.69	0.6	0.6	0.72	0.6	0.57	0.55
P1-3	0.7	0.95	0.6	0.68	0.7	0.97	0.59	0.69
P1-4	0.72	0.47	0.52	0.5	0.72	0.37	0.47	0.46
P2-1	0.71	0.57	0.55	0.54	0.71	0.49	0.53	0.51
P2-2	0.71	0.5	0.53	0.51	0.71	0.43	0.5	0.48
P2-3	0.71	0.71	0.59	0.59	0.7	0.79	0.6	0.62
P2-4	0.71	0.66	0.6	0.59	0.72	0.57	0.57	0.55
P3-1	0.72	0.68	0.6	0.6	0.72	0.58	0.56	0.54
P3-2	0.72	0.62	0.58	0.57	0.72	0.54	0.55	0.53
P3-3	0.72	0.62	0.58	0.57	0.7	0.98	0.59	0.69
P3-4	0.72	0.63	0.59	0.57	0.74	0.53	0.56	0.54
P4-1	0.72	0.83	0.64	0.66	0.72	0.73	0.6	0.6
P4-2	0.73	0.73	0.62	0.62	0.72	0.63	0.58	0.57
P4-3	0.7	0.96	0.6	0.69	0.7	0.99	0.59	0.7
P4-4	0.73	0.79	0.63	0.65	0.72	0.71	0.61	0.61
P5-1	0.7	0.48	0.51	0.49	0.71	0.69	0.59	0.58
P5-2	0.69	0.39	0.47	0.45	0.71	0.58	0.55	0.54
P5-3	0.7	0.67	0.58	0.57	0.7	0.96	0.6	0.69
P5-4	0.71	0.55	0.55	0.53	0.71	0.6	0.57	0.55
P6-1	0.72	0.79	0.63	0.64	0.7	0.38	0.56	0.45
P6-2	0.72	0.67	0.6	0.59	0.7	0.33	0.44	0.43
P6-3	0.7	0.93	0.6	0.68	0.7	0.75	0.59	0.6
P6-4	0.72	0.71	0.61	0.6	0.72	0.45	0.51	0.49
P7-1	0.72	0.55	0.55	0.53	0.72	0.59	0.57	0.55
P7-2	0.72	0.63	0.59	0.57	0.71	0.5	0.53	0.51
P7-3	0.7	0.94	0.59	0.67	0.7	0.97	0.6	0.69
P7-4	0.71	0.51	0.54	0.52	0.72	0.4	0.49	0.47
P8-1	0.72	0.92	0.64	0.69	0.71	0.93	0.62	0.69
P8-2	0.72	0.76	0.62	0.63	0.71	0.8	0.62	0.64
P8-3	0.7	0.97	0.58	0.68	0.71	0.99	0.58	0.7
P8-4	0.72	0.87	0.64	0.68	0.71	0.92	0.62	0.68
P9-1	0.72	0.9	0.64	0.69	0.71	0.68	0.59	0.59
P9-2	0.72	0.79	0.62	0.64	0.71	0.74	0.54	0.52
P9-3	0.7	0.97	0.58	0.68	0.7	0.97	0.6	0.69
P9-4	0.72	0.85	0.64	0.67	0.72	0.58	0.57	0.55

more detailed improvements, with some language features being less affected by standardization. Fig. 3 shows important details about these different results, illustrating how changes from preprocessing affected the evaluation metrics in different ways. The distribution patterns especially show that while most hypotheses improved with preprocessing, some only had slight improvements or even got worse, highlighting the complicated link between Arabic language features and how well preprocessing works.

Furthermore, we conducted a statistical comparison using the Wilcoxon signed-rank test for both the XNLI and Embeddings models across four key metrics. Our analysis indicates that the effect of preprocessing was different for each model and hypothesis, as shown in Table V and Fig. 4. The accuracy of the embedding model went up a lot by 10.9%, $p < 0.001$, showing that preprocessing can improve structured metrics, while the XNLI model only had a small drop in recall of 4.5%, p = 0.028, with precision staying the same. Fig. 4 show this variation: Many hypotheses are close to the "no change" line, like P1-1, but outliers such as P5-2 with +48.7% recall and P6-2 with -50.7% recall reveal that preprocessing can both enhance some patterns and hide others.

Cohen's d values in Table V show these trade-offs: the Embedding Model had a big increase in accuracy with d = 2.17 but a notable drop in recall with d = -0.68, while the XNLI Model's F1 score went down with d = -0.51 because it had trouble balancing precision and recall. These results emphasize that preprocessing is not always beneficial;

its efficacy hinges on both model architecture and linguistic nuances. For instance, dialect-specific hate speech (e.g. P6-2) resisted standardization, while context-dependent patterns (e.g. P7-3's +24.6% Precision) thrived.

These findings suggest that we should customize our methods: we need to check each preprocessing idea one by one, paying close attention to the balance between different measurements and the complexity of the language. As Fig. 4 vividly illustrates, preprocessing acts as a selective lens—enhancing clarity in some contexts while unintentionally blurring others.

TABLE V. STATISTICAL COMPARISON OF MODEL PERFORMANCE METRICS BEFORE VERSUS AFTER PREPROCESSING. RESULTS SHOW WILCOXON SIGNED-RANK TEST STATISTICS, SIGNIFICANCE LEVELS (ASTERISKS INDICATING SIGNIFICANCE PF P-VALUE), MEAN DIFFERENCES (AFTER - BEFORE), AND EFFECT SIZES (COHEN'S D)

Model	Metric	Wilcoxon	p-value	Sig	Diff	Cohen's d
XNLI	Precision	64.0	0.189		-0.002	-0.232
	Recall	181.0	0.028	*	-0.045	-0.287
	F1	111.0	0.001	**	-0.023	-0.513
	Accuracy	159.0	0.018	*	-0.026	-0.369
Embedding	Precision	111.5	0.007	**	0.023	0.478
	Recall	62.0	0.001	***	-0.012	-0.675
	F1	159.0	0.018	*	0.030	0.502
	Accuracy	0.0	0.000	***	0.109	2.171

C. Hate Speech Detection Targeting Protected Groups

In the final phase of our experimental framework, we extended our zero-shot hate speech detection approach to specifically target protected groups within Arabic discourse. To tailor our model for this scenario, we identified and employed the two most promising hypotheses based on our earlier experiments, including P7-3 for the Embeddings model and P8-3 for the XNLI model. We carefully crafted these hypotheses to accurately represent targeted hate speech expressions in Arabic, ensuring semantic generality for zero-shot classification. Using these refined hypotheses, we evaluated the performance of both models across seven protected groups: women, immigrants, Jews, Black people, transgender people, gay people, and disabled people.

The results from this detailed evaluation are displayed in Table VI, which shows the precision, recall, F1-score, and accuracy of both models for each group. In parallel, Fig. 5 provides a visual overview of the model performance per metric, facilitating a clearer comparison of strengths and weaknesses.

Our results indicate that the XNLI model is better than the Embeddings model, particularly with hypothesis P8-3, reaching an average accuracy of up to 80% for protected groups. For example, the XNLI model showed remarkable robustness in detecting hate speech targeting Jews, black people, and disabled individuals, groups that often experience nuanced and implicit forms of discrimination. On the other hand, the Embeddings model performed okay with hypothesis P7-3, but its results were less consistent and affected by the way language was used. These outcomes underscore two important insights: First, hypothesis engineering plays a crucial role in adapting zero-shot models to detect group-specific hate speech. Second, semantically informed models like XNLI, when paired with well-formulated hypotheses, can serve as powerful tools for hate speech detection in low-resource and linguistically diverse settings such as Arabic.

TABLE VI. COMPARISON OF HATE SPEECH DETECTION RESULTS TARGETING PROTECTED GROUPS USING DIFFERENT MODELS AND HYPOTHESES

		Embedding				XNLI			
Target	hypoth	Pre	Rec	F1-S	Acc	Pre	Rec	F1-S	Acc
Women	P7-3	**0.64**	0.54	0.53	**0.76**	0.81	0.87	0.63	0.75
	P8-3	0.58	0.54	0.54	0.74	0.81	0.9	0.74	0.76
immigrants	P7-3	0.62	0.54	0.53	**0.76**	**0.84**	0.74	0.71	0.7
	P8-3	0.59	**0.57**	**0.58**	0.73	**0.84**	0.8	0.74	0.73
Jews	P7-3	0.47	0.47	0.36	0.36	0.83	0.92	**0.78**	**0.8**
	P8-3	0.47	0.46	0.41	0.42	0.82	**0.95**	0.77	**0.8**
black_people	P7-3	0.59	0.5	0.55	0.73	0.8	0.9	0.73	0.74
	P8-3	0.53	0.53	0.52	0.61	0.8	**0.95**	0.74	0.78
trans_people	P7-3	0.62	0.54	0.53	**0.76**	**0.84**	0.71	0.69	0.68
	P8-3	0.55	0.54	0.54	0.7	**0.84**	0.83	0.75	0.75
gay_people	P7-3	0.59	0.54	0.54	0.74	0.81	0.79	0.69	0.69
	P8-3	0.54	0.54	0.54	0.66	0.81	0.9	0.75	0.77
disabled_people	P7-3	0.55	0.54	0.54	0.7	**0.84**	0.89	0.77	0.78
	P8-3	0.52	0.52	0.52	0.65	**0.84**	0.9	0.78	0.79

D. Ground Validation Using GPT-3

To meticulously validate the real-world feasibility of our carefully enhanced hypotheses, we adeptly harnessed the capabilities of the GPT-3 model via the GPT chat interface. To streamline the validation process, we devised an innovative strategy by formulating a Hypothesis (P8-3) as a singular prompt in form of النص التالي يحتوي على خطاب مثير للكراهية ضد [المجموعة]، النص: [النص]., literal translations in English; The following text contains hate speech against [group], text: [text]. The responses provided by the model were subsequently examined to determine to what extent they aligned with our hypotheses. Remarkably encouraging results emerged from this validation endeavor, particularly in the context of complex and low-resource languages like Arabic, The results of this ground validation are meticulously detailed in Table VII. The GPT chat interactions served as a robust testament to the effectiveness of our methodology in quickly and accurately determining hate speech directed at protected groups, thus solidifying the pragmatic utility of our methodology in real-world conversational scenarios.

TABLE VII. THE RESULTS GROUND VALIDATION USING GPT-3

Target	Precision	Recall	F1-Score	Accuracy
Women	0.77	0.43	0.45	0.47
immigrants	0.78	0.30	0.42	0.40
Jews	0.81	0.51	**0.51**	**0.54**
black_people	0.77	**0.53**	0.48	0.52
trans_people	0.79	0.31	0.43	0.41
gay_people	0.77	0.34	0.42	0.42
disabled_people	**0.84**	0.29	0.41	0.42

VII. DISCUSSION

Our results underscore the importance of hypothesis design and preprocessing. While preprocessing boosted embedding model accuracy, its impact on XNLI was nuanced, revealing trade-offs between precision and recall. As shown in Fig. 3 and Fig. 4, some hypotheses, like P5-2, demonstrated exceptional improvement, while others, such as P6-2, declined. This suggests that preprocessing may enhance or suppress specific linguistic cues depending on the model and hypothesis

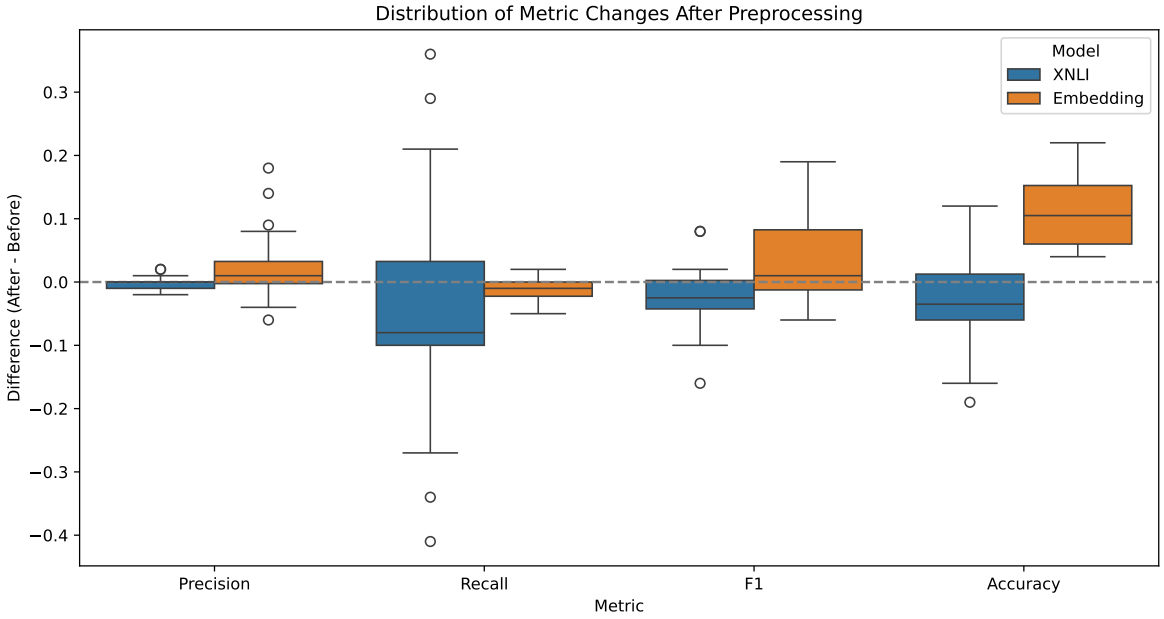

Fig. 3. Distribution of metric changes (After – Before preprocessing) for Precision, Recall, F1-score, and Accuracy across XNLI and embedding models.

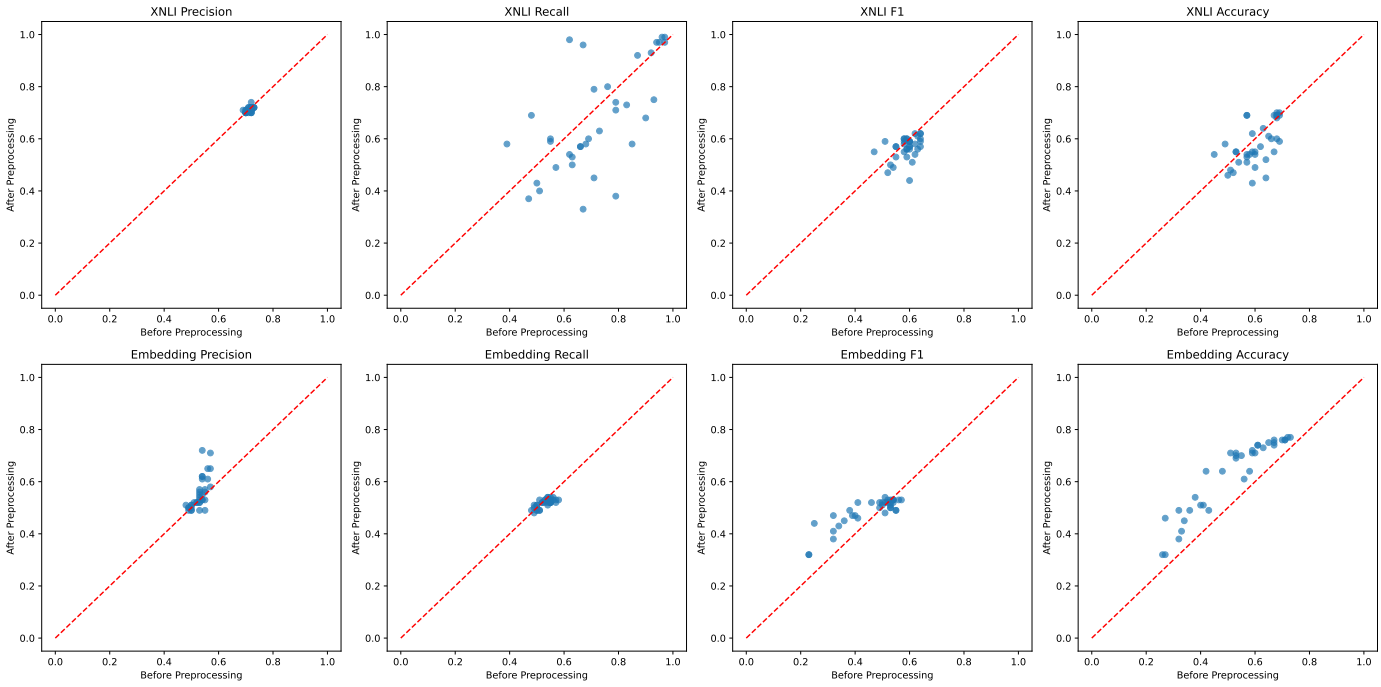

Fig. 4. Comparison of model performance metrics before versus after preprocessing. Each point represents a single hypothesis, plotted by its preprocessed (y-axis) versus original (x-axis) scores.

structure. Furthermore, Cohen's d metrics further confirmed that these improvements are statistically significant, especially in the case of structured models like embeddings. However, the decrease in XNLI's recall emphasizes that overly aggressive normalization can reduce the sensitivity needed to detect more subtle forms of hate speech.

A. Protected Group Detection and Model Robustness

The experiment targeting protected groups reinforces the critical role of hypothesis engineering. Our choice of P8-3 with the XNLI model resulted in robust performance across all groups, particularly for subtle, implicit hate speech such

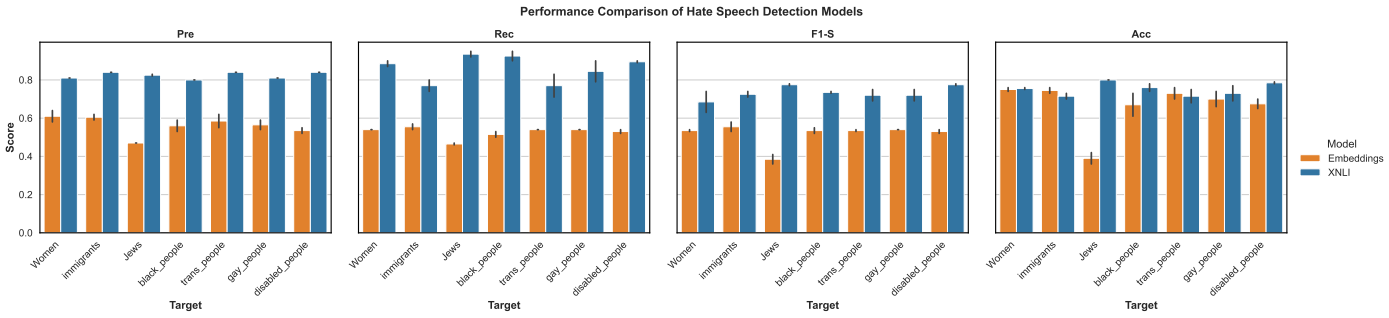

Fig. 5. Comparative performance of embeddings and XNLI models across protected groups (P7-3 vs. P8-3 hypotheses).

as against Jews or disabled individuals. In contrast, the Embedding model showed greater sensitivity to language usage patterns but lacked the consistency needed for generalization.

B. Validation and Practical Implications

GPT-3 validation confirmed the real-world applicability of our hypotheses in a zero-shot conversational setting, particularly for Arabic's informal communication. This bridges a critical gap in hate speech detection, where most studies lack conversational validation. Our framework's success in low-resource settings suggests its potential for ethical AI deployment in multilingual platforms. As shown in Table VII, our research results showed clear convergence across all categories, confirming the consistency of our hypotheses and opening the way for future research to improve these results based on this research.

C. Comparison with Previous Studies

Our results show meaningful progress in detecting hate speech for Arabic, a language often overlooked in AI research. Table VIII offers a comparative overview of performance across related studies that utilize zero-shot and few-shot learning techniques for hate speech detection. Notably, many prior works have focused on general hate speech detection in English using supervised transformer-based models or multilingual adaptations without tailoring hypothesis design or model evaluation to Arabic linguistic contexts.

As seen in Table VIII, our approach using the XNLI model achieved up to 80% accuracy on our expirements, outperforming earlier Arabic-focused studies (61%) and nearing the performance of top English models (75%). Furthermore, our study incorporates a real-world validation step using the GPT-3 model, providing evidence of its practical applicability in conversational contexts. While previous studies rarely go beyond benchmark datasets or synthetic stimuli, our approach bridges this gap, combining quantitative performance with qualitative validation in natural conversational settings.

These comparative results demonstrate the robustness of our proposed framework and its ability to overcome the limitations of experimental learning in resource-limited language environments. The improvements in precision, recall, and F1 score across multiple protected corpora demonstrate that well-designed hypotheses and language-specific preprocessing are vital for achieving accurate and ethical hate speech detection.

TABLE VIII. COMPARISON OF OUR RESULTS WITH PREVIOUS STUDIES USING HATE SPEECH DETECTION IN THE ARABIC LANGUAGE TARGETING PROTECTED GROUPS

Study	Model Used	Dataset	Acc(%)
Röttger et al. (2020) [2]	BERT fine-tuned on [31]	HateCheck-en	0.63
Goldzycher et al. (2022) [10]	NLI	HateCheck-en	0.75
Ggoldzycher et. al (2023) [22]	NLI	HateCheck-Ar	0.61
Our Study (Baseline)	Embeddings	HateCheck-Ar	**0.76**
Our Study (XNLI)	XNLI	HateCheck-Ar	**0.80**
Our Ground Validation	GPT-3.5	HateCheck-Ar	**0.54**

VIII. ERROR ANALYSIS

Our error analysis reveals critical linguistic challenges in detecting Arabic hate speech targeting protected groups. Three main factors emerged, as shown in Table IX, contributing to the misclassification. Each is rooted in the dynamic nature of informal Arabic communication on social media platforms.

First, the prevalence of Arabizi introduces unique complexities. The term "Arabizi" itself is a generic combination of the words "Arabi" (Arabic) and "engliszi" (English) [31]. It's a hybrid writing system combining English letters and numbers to approximate Arabic sounds. Commonly used in online spaces, as in example 1, "Allah ya5od kol el mthlyoon" [May God take (kill) all the gays], this practice forces models to navigate non-standard spellings where numbers like "5" replace Arabic letters like "خ" Such substitutions create lexical ambiguities, such as when automated character replacement systems struggle to restore the original Arabic script without context.

Secondly, separated letters—intentional spacing between normally connected Arabic characters—serve as an evasion tactic to evade automatic hate speech detection systems. This tactic is used to circumvent detection because the Arabic language is primarily composed of connected letters. As illustrated in Examples 3 and 4, words like "ا ق ت ل" instead of "اقتل" [kill] fracture word structures, bypassing pattern-matching algorithms trained on conventional text. These deliberate distortions require specialized preprocessing to reconstruct meaningful words, a step our initial pipeline lacked.

Finally, dialectal diversity profoundly impacts model performance. The Arabic language includes multiple dialects, resulting in differences in word usage and sentence structure across regions. Some words may be specific to certain dialects, making the intent of a sentence unclear to machine learning

TABLE IX. ILLUSTRATIVE EXAMPLES OF MISCLASSIFICATION SENTENCES

Example	Sentence	Group	Label_gold	Predicted	Error category
Example 1	Allah ya5od kol el mthlyoon. May God take (kill) all the gays.	Gays	hateful	**non-hateful**	Arabizi
Example 2	ما فيش7اجة اسواء من اليهودي There is no one worse than the Jews.	Jews	hateful	**non-hateful**	Arabizi
Example 3	لازم حد ي و ل ع في المثلين دول. Someone needs to set (fire) to these gay people.	Gays	hateful	**non-hateful**	Separated Letters
Example 4	انا عاوز ا ق ت ل كل اللاجئين. I want to kill all the refugees.	immigrants	hateful	**non-hateful**	Separated Letters
Example 5	انا مبعرفش كتير من المعاقيين. I don't know many disabled people.	disabled people	non-hateful	**hateful**	Dialectal confusion
Example 6	ازي بتقول النساء مقرفين ومتفتكرش انه ده كره اعمى. How do you say women are disgusting and don't think that this is blind hatred?	women	non-hateful	**hateful**	Dialectal confusion

models. Regional variations, such as the Egyptian phrase in example 5, "مبعرفش" (I don't know)—a fusion of "ما", "بعرف", and "ش" ("ma" and "ba'arf" with the Egyptian suffix "sh")—often defy standard grammatical rules. Models trained primarily on Modern Standard Arabic misinterpret such constructions, mistaking dialect-specific syntax for benign or ambiguous content. These intertwined challenges underscore a fundamental tension: the fluidity of informal Arabic communication clashes with the rigid patterns detectors typically recognize. While Arabizi and separated letters represent active circumvention strategies, dialectal variations expose gaps in linguistic coverage. Addressing these issues requires not just improved algorithms but a paradigm shift—integrating dialectal lexicons, adversarial training with manipulated text, and context-aware transliteration systems.

IX. CONCLUSION

In this study, we address the complex and increasingly important problem of detecting hate speech in Arabic, a linguistically rich but resource-poor language. Focusing specifically on hate speech targeting protected groups, we propose a comprehensive methodology that leverages hypothesis engineering and zero-shot learning through a Natural Language Inference (NLI) framework.

We started by preparing a set of Arabic-based hypotheses, written in pure Arabic, capable of capturing various expressions of hate speech. We then evaluated these hypotheses using two model architectures: a baseline embedding-based model and an XNLI model. Our experiments demonstrated that hypothesis engineering, especially when supported by preprocessing techniques such as normalization and lemmatization, significantly improves model performance in detecting hate speech. The XNLI model, in particular, demonstrated high accuracy results, achieving up to 80% accuracy in detecting targeted hate speech.

Furthermore, we validated our hypotheses using the GPT-3 model in real-time conversational scenarios via the ChatGPT interface. This step showed that we could successfully use our methodology on real-world systems that users interact with, achieving an accuracy of 54%, offering promising results for real-world moderation tools.

Future directions could focus on developing semi-automated hypothesis generation frameworks that could reduce reliance on manual curation, and adversarial training with synthetic Arabizi/text manipulation samples may enhance robustness. Cross-lingual adaptations of the methodology could benefit other low-resource languages, complemented by collaborative annotation efforts with affected communities to ensure ethical and culturally informed detection systems.

REFERENCES

[1] M. Bilewicz and W. Soral, "Hate speech epidemic. the dynamic effects of derogatory language on intergroup relations and political radicalization," *Political Psychology*, vol. 41, pp. 3–33, 2020.

[2] P. Röttger, B. Vidgen, D. Nguyen, Z. Waseem, H. Margetts, and J. B. Pierrehumbert, "Hatecheck: Functional tests for hate speech detection models," *arXiv preprint arXiv:2012.15606*, 2020.

[3] K. Müller and C. Schwarz, "Fanning the flames of hate: Social media and hate crime," *Journal of the European Economic Association*, vol. 19, no. 4, pp. 2131–2167, 2021.

[4] R. AlYami and R. Al-Zaidy, "Weakly and semi-supervised learning for arabic text classification using monodialectal language models," in *Proceedings of The Seventh Arabic Natural Language Processing Workshop (WANLP)*, 2022, pp. 260–272.

[5] Z. Obied, A. Solyman, A. Ullah, A. Fat'hAlalim, and A. Alsayed, "Bert multilingual and capsule network for arabic sentiment analysis," in *2020 international conference on computer, control, electrical, and electronics engineering (ICCCEEE)*. IEEE, 2021, pp. 1–6.

[6] S. Abro, S. Shaikh, Z. H. Khand, A. Zafar, S. Khan, and G. Mujtaba, "Automatic hate speech detection using machine learning: A comparative study," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 8, 2020.

[7] M. Khairy, T. M. Mahmoud, A. Omar, and T. Abd El-Hafeez, "Comparative performance of ensemble machine learning for arabic cyberbullying and offensive language detection," *Language Resources and Evaluation*, vol. 58, no. 2, pp. 695–712, 2024.

[8] A. Al-Hassan and H. Al-Dossari, "Detection of hate speech in arabic tweets using deep learning," *Multimedia systems*, vol. 28, no. 6, pp. 1963–1974, 2022.

[9] W. Yin, J. Hay, and D. Roth, "Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach," *arXiv preprint arXiv:1909.00161*, 2019.

[10] J. Goldzycher and G. Schneider, "Hypothesis engineering for zero-shot hate speech detection," *arXiv preprint arXiv:2210.00910*, 2022.

[11] M. M. Abdelsamie, S. S. Azab, and H. A. Hefny, "A comprehensive review on Arabic offensive language and hate speech detection on social media: methods, challenges and solutions," *Social Network Analysis and Mining*, vol. 14, no. 1, p. 111, May 2024.

[12] F. Shannaq, B. Hammo, H. Faris, and P. A. Castillo-Valdivieso, "Offensive language detection in arabic social networks using evolutionary-based classifiers learned from fine-tuned embeddings," *IEEE Access*, vol. 10, pp. 75 018–75 039, 2022.

[13] F. Shannag, B. H. Hammo, and H. Faris, "The design, construction and evaluation of annotated arabic cyberbullying corpus," *Education and Information Technologies*, vol. 27, no. 8, pp. 10 977–11 023, 2022.

[14] R. Duwairi, A. Hayajneh, and M. Quwaider, "A deep learning framework for automatic detection of hate speech embedded in arabic tweets," *Arabian Journal for Science and Engineering*, vol. 46, pp. 4001–4014, 2021.

[15] N. Albadi, M. Kurdi, and S. Mishra, "Investigating the effect of combining gru neural networks with handcrafted features for religious hatred detection on arabic twitter space," *Social Network Analysis and Mining*, vol. 9, no. 1, p. 41, 2019.

[16] M. T. Alrefaie, N. E. Morsy, and N. Samir, "Exploring tokenization strategies and vocabulary sizes for enhanced arabic language models," *arXiv preprint arXiv:2403.11130*, 2024.

[17] M. Abdelhakim, B. Liu, and C. Sun, "Ar-pufi: A short-text dataset to identify the offensive messages towards public figures in the arabian community," *Expert Systems with Applications*, vol. 233, p. 120888, 2023.

[18] A. Elmadany, C. Zhang, M. Abdul-Mageed, and A. Hashemi, "Leveraging affective bidirectional transformers for offensive language detection," 2020.

[19] 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," 2024. [Online]. Available: https://arxiv.org/abs/2407.02448

[20] F. Plaza del Arco, D. Nozza, D. Hovy *et al.*, "Respectful or toxic? using zero-shot learning with language models to detect hate speech," in *The 7th workshop on online abuse and harms (woah)*. Association for Computational Linguistics, 2023.

[21] J. A. García-Díaz, R. Pan, and R. Valencia-García, "Leveraging zero and few-shot learning for enhanced model generality in hate speech detection in spanish and english," *Mathematics*, vol. 11, no. 24, 2023.

[22] J. Goldzycher, M. Preisig, C. Amrhein, and G. Schneider, "Evaluating the effectiveness of natural language inference for hate speech detection in languages with limited labeled data," *arXiv preprint arXiv:2306.03722*, 2023.

[23] H. B. Zia, I. Castro, A. Zubiaga, and G. Tyson, "Improving zero-shot cross-lingual hate speech detection with pseudo-label fine-tuning of transformer language models," in *Proceedings of the International AAAI conference on web and social media*, vol. 16, 2022, pp. 1435–1439.

[24] F. T. J. Faria, L. H. Baniata, and S. Kang, "Investigating the predominance of large language models in low-resource bangla language over transformer models for hate speech detection: A comparative analysis," *Mathematics*, vol. 12, no. 23, 2024.

[25] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *International conference on machine learning*. PMLR, 2015, pp. 2152–2161.

[26] Y. Meng, J. Huang, Y. Zhang, and J. Han, "Generating training data with language models: Towards zero-shot language understanding," *Advances in Neural Information Processing Systems*, vol. 35, pp. 462–477, 2022.

[27] Y. Belinkov, A. Poliak, S. M. Shieber, B. Van Durme, and A. M. Rush, "Don't take the premise for granted: Mitigating artifacts in natural language inference," *arXiv preprint arXiv:1907.04380*, 2019.

[28] J. H. Martin, *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall, 2009.

[29] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[30] S. Ahmadi, "Klpt–kurdish language processing toolkit," in *Proceedings of second workshop for NLP open source software (NLP-OSS)*, 2020, pp. 72–84.

[31] A. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos, and N. Kourtellis, "Large scale crowdsourcing and characterization of twitter abusive behavior," in *Proceedings of the international AAAI conference on web and social media*, vol. 12, no. 1, 2018.