# Exploring Data and Parameter Efficient Strategies for Arabic Dialect Identifications

**Vani Kanjirangat**
SUPSI, IDSIA, Switzerland
vanik@idsia.ch

**Ljiljana Dolamic**
armasuisse S+T, Switzerland
Ljiljana.Dolamic@armasuisse.ch

**Fabio Rinaldi**
SUPSI, IDSIA, Switzerland
fabio.rinaldi@idsia.ch

## Abstract

This paper discusses our exploration of different data-efficient and parameter-efficient approaches to Arabic Dialect Identification (ADI). In particular, we investigate various soft-prompting strategies, including prefix-tuning, prompt-tuning, P-tuning, and P-tuning V2, as well as LoRA reparameterizations. For the data-efficient strategy, we analyze hard prompting with zero-shot and few-shot inferences to analyze the dialect identification capabilities of Large Language Models (LLMs). For the parameter-efficient PEFT approaches, we conducted our experiments using Arabic-specific encoder models on several major datasets. We also analyzed the n-shot inferences on open-source decoder-only models, a general multilingual model (Phi-3.5), and an Arabic-specific one(SILMA). We observed that the LLMs generally struggle to differentiate the dialectal nuances in the few-shot or zero-shot setups. The soft-prompted encoder variants perform better, while the LoRA-based fine-tuned models perform best, even surpassing full fine-tuning.

## 1 Introduction

The task of Dialect identification (DI) focuses on classifying the given input utterance to a specific dialect class. Dialectal cues can be quite nuanced, with a lot of fine-grained overlaps (Zampieri et al., 2017, 2018). In recent years, hard prompting has emerged as a simple yet effective and data-efficient approach for leveraging large language models (LLMs) in various natural language processing (NLP) tasks (Brown et al., 2020). Due to their extensive pre-training and advanced reasoning capabilities, LLMs enable zero-shot and few-shot inference, offering data-efficient solutions across a wide range of NLP tasks. While these models have demonstrated impressive performance, much focus has been on English-centric benchmarks (Lai et al., 2023). In the context of dialect identifications (DI),

studies exploring zero-shot and few-shot performance have primarily focused on open-source models, such as GPT and Gemini (Khondaker et al., 2023; Bang et al., 2023; Rane et al., 2024). Supervised fine-tuning allows encoder-based models to acquire domain-specific knowledge in the context of transfer learning (Pan, 2020). Applying this approach to LLMs introduces challenges such as pretrain-finetune discrepancy (Yang et al., 2019), inherited pretraining biases, and the high computational cost of fine-tuning. To mitigate these issues, Parameter-Efficient Fine-Tuning (PEFT) techniques (Lialin et al., 2023) have been introduced, which aim to achieve task adaptability with minimal parameter updates.

In this study, we focus on evaluating data-efficient and parameter-efficient strategies by selecting Arabic as the primary language and Arabic Dialect Identification (ADI) as the target task. Arabic is a language spoken by a large community of approximately 400 million people, which is widely distributed across various countries and regions. The regional Arabic dialects differ from the Modern Standard Arabic (MSA) in lexical, syntactic, and phonetic aspects (MSA: the official language in many Arabic-speaking countries)(Zaidan and Callison-Burch, 2014).

We define the research goals for the proposed work as: *Firstly*, to analyze the data-efficient capabilities of LLMs in zero-shot and few-shot settings for the Arabic Dialect Identification (ADI) task. *Secondly*, to evaluate and compare parameter-efficient fine-tuning methods, specifically LoRA and soft prompting, in the context of ADI tasks, and *thirdly*, to perform a comparative analysis of prompting and fine-tuning strategies across multiple ADI datasets using Arabic-specific LLMs.

## 2 Methods

In this section, we describe the approaches in detail: data-efficient and Parameter-efficient approaches.

# 3 Data-efficient approaches

Although manual engineering of the prompts can be cumbersome, it remains practical and efficient in many applications. Prompting has emerged as a practical approach to infer LLMS without needing full fine-tuning (Brown et al., 2020). To analyze the dialect classification capabilities of LLMs, we focused on using the zero-shot (ZS) and few-shot (FS) inference strategies with LLMs.

We experimented with different ZS prompt variations, such as *vanilla prompt*, *chain-of-thought (CoT)* inspired prompting, and *binary prompting*. For *few-shot*, a general approach was utilized, where some sample input-output pairs from the training set were used to establish the few-shot examples. Further, as a strategy inspired by *Clue And Reasoning Prompting (CARP)* (Sun et al., 2023) was also employed. CARP adopts a progressive reasoning strategy by first prompting the LLM to extract superficial clues (e.g., keywords, tones, semantic relations, references, etc), In our case, we prompted ChatGPT (OpenAI, 2023) to get the specific dialectal vocabularies for each dialect and further used them as clues in the prompt. The templates of different prompting strategies are shown in Figures 3, 4, and 5 in Appendix B.

## 3.1 Parameter-Efficient Fine-Tuning (PEFT)

PEFT includes a set of approaches that enable the efficient adaptation of LLMs in terms of memory and computational performance (Lialin et al., 2023). In this paper, we experimented with two variants of PEFT: *reparameterization-based* and *soft-prompting* methods.

**Low-Rank Adaptation (LoRA)** LoRA (Hu et al., 2022) is a reparameterization method designed to adapt large pre-trained models with minimal additional parameters. Instead of updating the full weight matrices during training, LoRA freezes the original model weights and injects trainable low-rank matrices into specific layers, which significantly reduces memory and computational costs.

**Prefix-Tuning** Prefix-tuning (Li and Liang, 2021) is a soft-prompt approach where a small set of learnable vectors — prefixes — are prepended to the input at each layer of a pre-trained language model (PLM). These prefixes can be interpreted as a sequence of virtual tokens that condition the model's internal representations without altering its original parameters.

**Prompt-Tuning** This soft-prompting (Lester et al., 2021) involves using a PLM without any parameter updates and relies on natural language templates to guide the model's behavior. It introduces soft prompts, which are appended to the input embeddings. These soft prompts are optimized via backpropagation, while keeping the rest of the pre-trained model frozen.

**P-Tuning** P-tuning (Liu et al., 2024) is a soft-prompt designed to enhance the performance of language models like GPTs on Natural Language Understanding (NLU) tasks, where traditional fine-tuning often falls short. Unlike full fine-tuning, which updates all model parameters, P-tuning introduces trainable soft prompts that are prepended to the input. These embeddings are optimized to guide the model's behavior for specific downstream tasks, while the original parameters remain frozen. A distinctive feature of P-tuning is its use of a Long Short-Term Memory (LSTM) network to generate the soft prompts. The LSTM enables the model to capture sequential dependencies within the prompts, allowing more flexible and expressive task conditioning than static embeddings.

**P-TuningV2** P-Tuning v2 (Liu et al., 2022) extends the original P-Tuning approach and is designed to improve performance in both text generation and knowledge probing tasks. Functionally, it can be viewed as an application of Prefix-Tuning to encoder-based models, such as BERT. While earlier methods, such as prompt tuning, appended continuous prompts only at the input layer, P-Tuning v2 introduces deep prompt tuning, where trainable continuous prompts are inserted at every transformer layer. This significantly increases the capacity and expressiveness of the prompts, allowing the model to better capture task-specific nuances without modifying the underlying pre-trained parameters.

# 4 Datasets and Models

**Data:** In this work, we utilize different ADI datasets that cover a wide range of Arabic dialects, with different complexity levels. This includes Vardial ADI, Arabic Online Commentary (AOC), and NADI datasets. The VarDial ADI (Zampieri et al., 2017), focused on five classes - *MSA, Egyptian, Gulf, Levantine, Moroccan, North-African*. AOC (Zaidan and Callison-Burch, 2011) covers

|        | NADI -2022 | NADI -2023 | VarDial -ADI | AOC   |
|--------|-----------|-----------|--------------|-------|
| Train  | 20398     | 18000     | 21001        | 86541 |
| Dev    | 4871      | 1800      | 1566         | 10820 |
| Test   | 4871      | -         | 1492         | 10812 |

Table 1: The size of datasets expressed as the number of utterances.

*MSA* and the dialectal varieties - *Egyptian, Gulf, Levantine, Moroccan*. NADI datasets, available since 2020 (Muhammad et al., 2020), were built upon and extended the MADAR dataset (Bouamor et al., 2019) by introducing a fine-grained, sub-country level dialect identification task. NADI-2022 (Abdul-Mageed et al., 2022) includes approximately 20,000 tweets across 18 dialects. The data statistics of each dataset are presented in Table 1. Figure 2 in Appendix A shows the label distribution of NADI datasets, showing that datasets from 2020 to 2022 are quite unbalanced. In contrast, the NADI-2023 dataset provides 18 dialects with a more balanced distribution (Abdul-Mageed et al., 2024).

**Models:** We compare the following models under full fine-tuning (FFT)- *AraBERT, AraBERT Twitter, CamelBERT, MultiDialectBERT, MARBERTv2*. AraBERT (Antoun et al.) is an Arabic PLM based on BERT, trained with OSCAR unshuffled and filtered, Arabic Wikipedia dump, the 1.5B words Arabic Corpus, the OSIAN Corpus, and Assafir news articles. AraBERT Twitter was trained by continuing the pre-training using the MLM task on 60M Arabic tweets (filtered from a collection of 100M). CamelBERT (Inoue et al., 2021) is a collection of BERT models pre-trained on Arabic texts with different sizes and variants - pre-trained language models for MSA, dialectal Arabic (DA), classical Arabic (CA), and a model pre-trained on a combination of the three. MultiDialectBERT (Talafha et al., 2020) is initialized with the model weights using Arabic-BERT and trained on 10M Arabic tweets from the unlabeled data of the NADI shared task. MARBERT (Mageed et al., 2021) is pre-trained on very large and diverse datasets to facilitate transfer learning on MSA as well as Arabic dialect, along with a large Twitter dataset. It randomly samples 1B Arabic tweets from an extensive in-house dataset of about 6B tweets.

# 5 Experimental Settings, Results & Analysis

## 5.1 Parameter-efficient approach results

Table 2 reports the FFT results across different ADI datasets on various Arabic-specific encoder models. After hyperparameter optimization, we used a dropout rate of 0.3, a learning rate (lr) of 1e-5, a batch size of 8, and 5 epochs. It can be observed that there is a wide discrepancy between the NADI-2022 and 2023 performances, where the latter was a balanced dataset.

In Figure 1, we report the results with PEFT-based approaches with the MARBERTv2 model, since this model presented the best performances across various ADI datasets. For evaluation, we used NADI-2023, considering its complexity due to the inclusion of 18 dialects, while maintaining balanced distributions. We use LoRA with r=8 in parameterizations while keeping the other hyperparameters the same as FFT. In soft-prompting approaches, we compare the different techniques. For P-tuning, we used *SEQ_CLS* task, with lr of 1e-3, a weight_decay of 0.01, and batch_size of 8. We used a similar configuration for prompt-tuning and prefix-tuning with 20 virtual_tokens, token_dim of 768, num_transformer_submodules of 1, 12 attention_heads, and 12 layers. For these soft-prompting experiments, we used the corresponding HuggingFace wrappers. For the P-tuningV2, we adapted (Liu et al., 2022)[1].

It can be observed that all soft-prompting approaches performed similarly except P-tuningV2, presenting a F-score of 83%, improving by 3 points. The best performance was achieved with LoRA, which outperformed even full fine-tuning by 1 point.

## 5.2 Data-efficient approach results

One of the main challenges in ZS and FS inferences is to constrain the generation of LLMs to the predictions or classes we intended. Open-source LLMs are less straightforward than closed-source conversational models, such as ChatGPT or Gemini. For constraining the outputs properly, we relied on the library Skorch [2]. Skorch ensures that the model always predicts the expected labels by intercepting the model predictions (the logits) and forcing them to be one of the labels. We experimented with two open-source multilingual LLMs

---

[1] https://github.com/THUDM/P-tuning-v2
[2] https://skorch.readthedocs.io/en/stable/

| Models | NADI 2022 | NADI 2023 | Vardial ADI | AOC |
|---|---|---|---|---|
| AraBERT V.02 | 0.25 | 0.71 | 0.35 | 0.73 |
| AraBERT Twitter | 0.29 | 0.79 | 0.39 | 0.73 |
| CamelBERT | 0.24 | 0.73 | 0.35 | 0.72 |
| MultidialectBERT | 0.27 | 0.73 | **0.41** | 0.71 |
| MARBERTV2 | **0.30** | **0.84** | 0.40 | **0.79** |

Table 2: Model performance across various Arabic dialect datasets using full fine-tuning


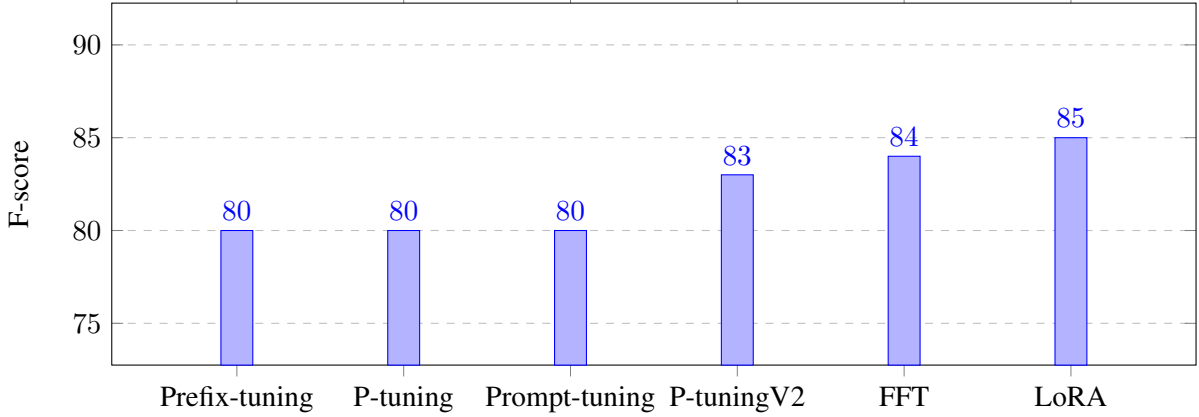
Figure 1: Comparison of F-scores (%) of various PEFT methods with Full Fine Tuning (in NADI-2023 ADI dataset)

- *Phi-3.5-mini* and *SILMA*. SILMA is an Arabic-specific LLM with 9B parameters, based on the Gemma-7B model, and is essentially multilingual. We performed the inferences on the NADI-2023 dataset.

With the Phi-3.5-mini, we achieved a zero-shot (ZS) F-score of only 8%, with a significant bias toward the Egyptian dialect. The situation did not improve with the Arabic-specific SILMA model, which showed a strong bias towards Saudi Arabian dialect. Removing the biased label did not resolve the issue, as the model consistently shifted its bias to a new label at each phase. We also attempted simple binary prompting (Figure 4) to reduce the complexity of the prompt. This resulted in a slight improvement, but not as much as expected. We also analyzed some samples using ChatGPT and Gemini, where the trend again appears to be biased towards Egyptian and Saudi Arabian dialects. The CARP-inspired approach did not yield the intended results. Instead, the clues seemed to act as noise. We could argue that the dialectal features generated by ChatGPT may not be suitable, and we may need to rely on manual curations, which could be expensive. Our experiments on Arabic dialects (NADI-2023) showed that LLMs' zero-shot or few-shot ability for dialectal discrimination is quite limited.

**Observations:** Model prediction may depend heavily on the pre-training data, such as the dialectal variety it has seen, without any specific understanding of the dialectal categorization features. With few shots, the samples may not be as indicative for this specific task as for other NLP tasks, such as sentiment analysis. In the specific use case of dialect classifications, the labels by themselves do not constitute any semantic meaning.

## 6 Conclusion

In this paper, we present the analysis and comparison of the data-efficient prompting strategies and parameter-efficient fine-tuning approaches in the Arabic dialect identification task. We observed that the performance varies across the dialectal datasets based on the complexity and granularity of the dialectal classes. The performance across various Arabic-specific encoder variants shows that the PEFT approaches can be quite effective in these tasks. At the same time, the prompting strategies with LLMs reveal that regardless of the prompt variations, LLMs struggle to understand the nuanced dialectal cues. The challenge increases with the hard classification task, since dialectal varieties can often overlap, and sometimes these can be fine-lined.

# References

Muhammad Abdul-Mageed, Amr Keleg, AbdelRahim Elmadany, Chiyu Zhang, Injy Hamed, Walid Magdy, Houda Bouamor, and Nizar Habash. 2024. NADI 2024: The Fifth Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of The Second Arabic Natural Language Processing Conference (ArabicNLP 2024)*.

Muhammad Abdul-Mageed, Chiyu Zhang, Abdel-Rahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The third nuanced Arabic dialect identification shared task. In *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718.

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The madar shared task on arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 199–207.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104.

Md Tawkat Islam Khondaker, Abdul Waheed, Muhammad Abdul-Mageed, et al. 2023. Gptaraeval: A comprehensive evaluation of chatgpt on arabic nlp. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 220–247.

Viet Lai, Nghia Ngo, Amir Pouran Ben Veyseh, Hiéu Mãn, Franck Dernoncourt, Trung Bui, and Thien Nguyen. 2023. Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13171–13189.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. 2023. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2024. Gpt understands, too. *AI Open*, 5:208–215.

Muhammad Abdul Mageed, Abdelrahim Elmadany, et al. 2021. Arbert & marbert: Deep bidirectional transformers for arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105.

Abdul-Mageed Muhammad, Zhang Chiyu, Bouamor Houda, and H Nizar. 2020. Nadi 2020: The first nuanced arabic dialect identification shared task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 97–110.

OpenAI. 2023. Chatgpt. Version: GPT-4, Mar 14. Available at https://chat.openai.com.

Sinno Jialin Pan. 2020. Transfer learning. *Learning*, 21:1–2.

Nitin Rane, Saurabh Choudhary, and Jayesh Rane. 2024. Gemini versus chatgpt: applications, performance, architecture, capabilities, and implementation. *Journal of Applied Artificial Intelligence*, 5(1):69–93.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. In *Findings*

*of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005.

Bashar Talafha, Mohammad Ali, Muhy Eddin Za'ter, Haitham Seelawi, Ibraheem Tuffaha, Mostafa Samir, Wael Farhan, and Hussein Al-Natsheh. 2020. Multi-dialect arabic bert for country-level dialect identification. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 111–118.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Omar Zaidan and Chris Callison-Burch. 2011. The arabic online commentary dataset: an annotated dataset of informal arabic with high dialectal content. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 37–41.

Omar F Zaidan and Chris Callison-Burch. 2014. Arabic dialect identification. *Computational Linguistics*, 40(1):171–202.

Marcos Zampieri, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. 2017. Findings of the vardial evaluation campaign 2017. In *Proceedings of the fourth workshop on NLP for similar languages, varieties and dialects*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, et al. 2018. Language identification and morphosyntactic tagging. the second vardial evaluation campaign.

## A Dataset details

Figure 2 represents the label distributions of NADI 2020-2022. NADI-2023 has a balanced distribution.

## B Prompt Details

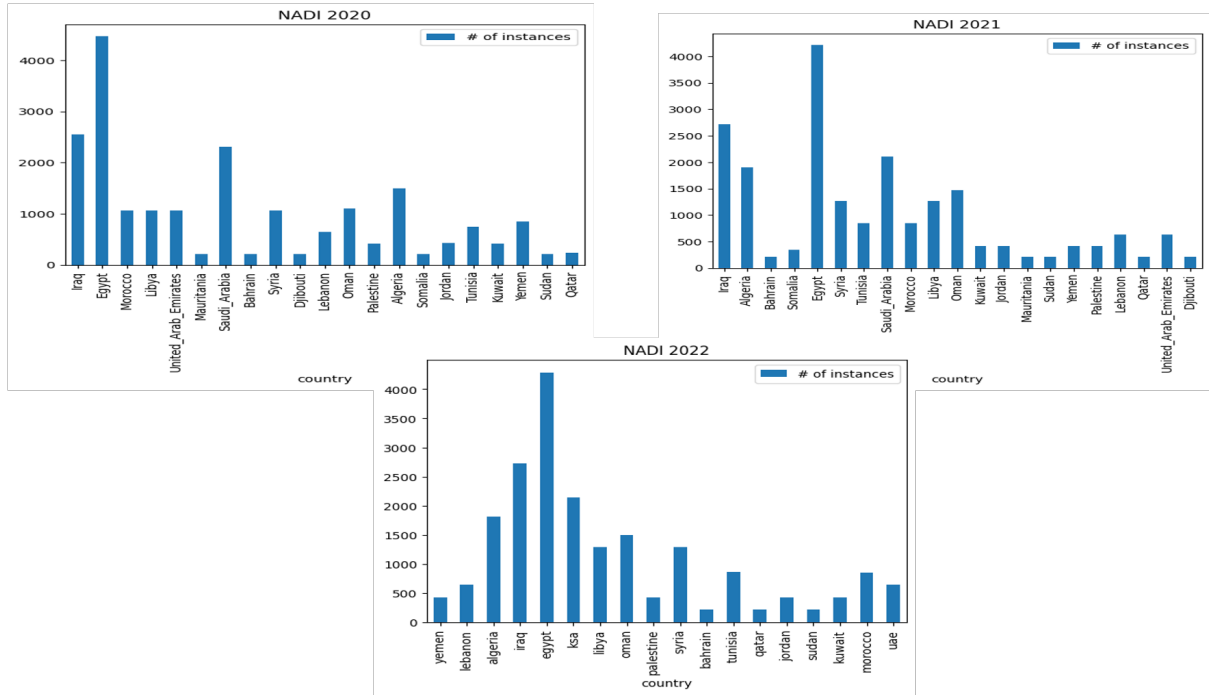The main prompt templates are given in Figures 3, 4, and 5.

Figure 2: Label distributions of NADI 2020-2022

```
system_msg = You are a dialect classification model that is really good at following instructions. Please follow the user's instructions as precisely as you can.

user_prompt =  Your task will be to classify the given Arabic text into one of the following classes: {classes}.

Please respond with a single label that you think fits the text best. Each of the input text belongs to a specific Arabic dialect.
Analyze the dialectal features and then give the prediction. Classify the following list
of Arabic dialectal texts:
text: '{X}'
class:
```

Figure 3: Vanilla zero-shot prompt.

```
user_prompt = Your task will be to classify the given Arabic text and decide whether it belongs to a given dialect or not.
The dialect classes are given: {classes}. Each of the input text belongs to a specific Arabic dialect.
Analyze the dialectal features and then give your prediction. Let us follow a step by step process.

1. Assume yourself as a binary classifier
2. Ask whether the text belongs to each of the dialects in the given
list or not: {classes}.
3. Output the predictions
Arabic Input text: '{X}'
predictions:
"""
```

Figure 4: Binary zero-shot prompt.

```
You are a dialect classifier. List the most important textual features of the Arabic dialects from the given list if you need to classify them.
Please note that classification is based only on the written text. Can you provide a dialectal vocabulary of 20-30 words for each dialect?.

The dialects are:
["Algeria","Bahrain","Egypt","Iraq","Jordan","Kuwait","Lebanon","Libya","Morocco",
"Oman","Palestine","Qatar",
"Saudi_Arabia","Sudan","Syria","Tunisia","UAE","Yemen"].

Please give the answer in list of dictionaries with each dict key corresponding to
the dialect and the values the vocab list. Include only the Arabic words."
```

Figure 5: Few-shot prompt inspired by CARP (Sun et al., 2023).