AraReasoner: Evaluating Reasoning-Based LLMs for Arabic NLP

Ahmed Hasanaath[1], Aisha Alansari[1], Ahmed Ashraf[1], Chafik Salmane[3],

Hamzah Luqman[1,2], Saad Ezzini[1]

[1]King Fahd University of Petroleum and Minerals

[2]SDAIA–KFUPM Joint Research Center for Artificial Intelligence

[3]Mohammed VI Polytechnic University

Correspondence: `hluqman@kfupm.edu.sa`

ABSTRACT

Large language models (LLMs) have shown remarkable progress in reasoning abilities and general natural language processing (NLP) tasks, yet their performance on Arabic data, characterized by rich morphology, diverse dialects, and complex script, remains underexplored. This paper presents a comprehensive benchmarking study of multiple reasoning-focused LLMs, with a special emphasis on the newly introduced DeepSeek models, across a suite of fifteen Arabic NLP tasks. We experiment with various strategies, including zero-shot, few-shot, and fine-tuning. This allows us to systematically evaluate performance on datasets covering a range of applications to examine their capacity for linguistic reasoning under different levels of complexity. Our experiments reveal several key findings. First, carefully selecting just three in-context examples delivers an average uplift of over 13 F1 points on classification tasks—boosting sentiment analysis from 35.3 % to 87.5 % and paraphrase detection from 56.1 % to 87.0 %. Second, reasoning-focused DeepSeek architectures outperform a strong GPT o4-mini baseline by an average of 12 F1 points on complex inference tasks in the zero-shot setting. Third, LoRA-based fine-tuning yields up to an additional 8 points in F1 and BLEU compared to equivalent increases in model scale. The code is available at Project Repository
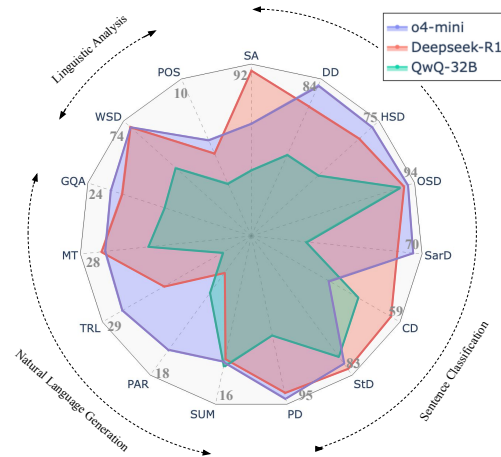
Figure 1: Zero-shot performance of reasoning-based models on 15 Arabic tasks, grouped into Sentence Classification, Linguistic Analysis, and Natural Language Generation.

1 Introduction

Arabic is spoken by more than 400 million people across 22 countries, making it one of the most widely used languages in the world. Despite its global significance, Arabic has traditionally been underrepresented in natural language processing (NLP) research (Alturayeif et al., 2022). This gap is partially due to the inherent complexity of the language, which features a rich morphological system, multiple dialects, and a non-Latin script. For instance, modern standard Arabic (MSA) differs considerably from various colloquial dialects in syntax and vocabulary, increasing the difficulty for models to generalize across tasks such as sentiment analysis, summarization, and translation. Moreover, diacritization and transliteration tasks pose unique challenges, as they require a nuanced understanding of phonological rules and script variations. These factors underscore the importance of dedicated research efforts to improve Arabic NLP capabilities.

Large-scale language models (LLMs) have demonstrated outstanding performance across a wide range of NLP tasks (Chang et al., 2024). Models from the GPT (Radford et al., 2018), Gemini (Team et al., 2023), Deepseek (Bi et al., 2024), and LLaMA (Touvron et al., 2023) families have shown exceptional capabilities in different NLP tasks. These models exhibit a strong understanding of natural language and express fluently in hu-

man language (Alyafeai et al., 2023a). This performance can be attributed mainly to their model architecture and the pre-training on large-scale corpora. Recently, the newly introduced reasoning-based models, such as DeepSeek R1 and GPT-4o, have demonstrated impressive performance in several NLP tasks (Cheng et al., 2025). Although these models showed a good performance in English and other high-resource languages, their adaptability to Arabic remains an open question as arabic remains a morphologically rich yet low-resource language (Antoun et al., 2020).

This study aims to systematically evaluate reasoning-based LLMs on fifteen critical Arabic NLP tasks. These tasks have been categorized into sentence classification (**SC**), natural language generation (**NLG**), and linguistic analysis (**LA**) tasks. The SC cluster involves sentiment analysis, dialect, hate speech, offensive speech, sarcasm, claim, stance, and paraphrase detection. The NLG cluster involves generation-based tasks, such as summarization, paraphrasing, transliteration, machine translation, and question answering. The LA tasks are word sense disambiguation (WSD) and part-of-speech (PoS) tagging. These tasks have been evaluated under various learning paradigms, such as zero-shot, in-context learning, and fine-tuning. Moreover, we compare the performance of the reasoning-based models with other recent LLMs. Through this comprehensive evaluation, we seek to identify both the potential and the shortcomings of these advanced reasoning-based models in handling Arabic's linguistic complexities, ultimately guiding further research and development in this area.

2 Related Work

Evaluating the performance of these LLMs requires robust, realistic, and task-specific benchmarks to ensure fair and meaningful comparisons. Due to the unique characteristics of the Arabic language, its complex morphology, rich diacritic system, and numerous dialects, Arabic-specific benchmarks have been developed to better capture these challenges. For instance, diacritization (AlKhamissi et al., 2020), the task of restoring diacritical marks to Arabic text, is a crucial task specific to Arabic. Transliteration (Shazal et al., 2020a) is another Arabic-specific task, often involving converting Arabizi (romanized Arabic script) into standard Arabic script. Additional tasks and benchmarks

tailored to Arabic are discussed in more detail in Section 3.1. Prior studies have examined the performance of LLMs across multiple Arabic NLP tasks. For instance, (Elmadany et al., 2022) introduced recently ORCA, a challenging benchmark for Arabic language understanding, consisting of 60 different datasets across seven natural language understanding tasks. They used ORCA to offer a comprehensive comparison between 16 multilingual and Arabic pretrained language models. However, their evaluation is restricted to BERT-based models, focusing solely on understanding tasks and excluding generative capabilities and newer architectures such as decoder-only LLMs.

Taqyimitw (Alyafeai et al., 2023b) assessed the performance of GPT-3.5 and GPT-4 across seven Arabic NLP tasks: sentiment analysis, translation, transliteration, paraphrasing, PoS tagging, summarization, and diacritization, proving the outperformance of GPT-4 on six tasks. They also provide an evaluation pipeline that facilitates the evaluation of such models. However, their study is limited to just two closed-source models from the same family, leaving out a broader range of open-source and multilingual models. Abdelali et al. introduced LAraBench (Abdelali et al., 2023) for evaluating Arabic NLP and speech processing, comparing multiple models, including GPT-3.5, GPT-4, BLOOMZ, Jais-13b-chat, Whisper, and USM, using zero-shot and few-shot learning techniques to address 33 distinct tasks across 61 publicly available datasets. They found that state-of-the-art (SOTA) models generally outperformed the evaluated models in both zero-shot and few-shot settings, with a few exceptions in few-shot prompting.

Although prior studies such as ORCA, Taqyim, and LAraBench have significantly advanced the evaluation of Arabic NLP, there remains a notable gap in assessing the reasoning capabilities of more recent LLMs. Prior efforts have either focused on NLU using BERT-based models or a limited set of closed-source generative models. In this paper, we address this gap by evaluating a new set of SOTA reasoning-oriented models including DeepSeek-V3 (Liu et al., 2024a), DeepSeek-R1 (Liu et al., 2024a), GPT-4o, Qwen (Bai et al., 2023), and QwQ (Qwen-Team, 2025) on a diverse suite of Arabic NLP tasks.

3 Benchmark

Figure 2 shows the pipeline of AraReasoner. We selected fifteen Arabic NLP tasks to evaluate how well reasoning-based LLMs perform across a diverse range of language understanding challenges. A prompt instruction selection process was followed for each task to identify the most effective prompt (see Appendix C). Subsequently, extensive evaluation experiments were conducted under various prompting strategies, including zero-shot and in-context learning. In addition, we compared the reasoning-based models not only against each other but also against non-reasoning LLMs to highlight the impact of reasoning abilities on task performance.

3.1 Tasks and Datasets

In this study, we evaluate reasoning-based models on diverse Arabic NLP tasks. Based on their task type, we group these tasks into three categories: SC, NLG, and LA. The dataset statistics for each task are provided in the Appendix (Table 6).

Sentence classification. This category encompasses a diverse range of sentence-level classification NLP tasks. Eight tasks have been involved in this work: sentiment analysis (**SA**), dialect detection (**DD**), sarcasm detection (**SarD**), hate speech detection (**HSD**), offensive speech detection (**OSD**), claim detection (**CD**), stance detection (**StD**), and paraphrase detection (**PD**). The SA task involves determining a sentence's emotional tone or opinion, usually as positive, negative, or neutral. We use the AJGT Corpus, which comprises tweets in the Jordanian dialect (Alomari et al., 2017), and we used the split proposed by (Nagoudi et al., 2022). The DD task distinguishes MSA from dialectal Arabic, whereas the SarD task identifies whether a given tweet conveys sarcasm. For both DD and SarD tasks, we utilize the ArSarcasm corpus (Farha and Magdy, 2020), which includes annotated labels for both sarcasm and dialect. The HSD is the task of identifying language that expresses hatred, discrimination, or hostility toward individuals or groups based on traits like race, gender, religion, or ethnicity. The OSD task captures rude or disrespectful language that may insult but is not necessarily individual or group-targeted. For HSD and OSD tasks, we employ the dataset proposed by (Mubarak et al., 2020). The CD task involves checking the factuality of a claim. The StD task detects the writer's stance towards a par-

ticular target or topic. The goal is to detect whether the writer agrees, disagrees, discusses, or has no opinion on a given topic. For CD and StD tasks, we used the dataset presented in (Khouja, 2020) as it provides labels for both tasks. PD checks whether a pair of sentences conveys the same meaning, and we use the dataset introduced by (Seelawi et al., 2019) for this task, which consists of Arabic sentence pairs annotated for semantic equivalence.

Natural language generation. This category includes five principal tasks in **NLG**. It involves summarization (**SUM**), paraphrasing (**PAR**), transliteration (**TRL**), machine translation (**MT**), and generative question answering (**GQA**) tasks. The SUM task involves generating a concise and coherent version of a longer text while preserving its key information. For this task, we use the AraSum (Kahla et al., 2021) as a training set and EASC as a testing set (El-Haj et al., 2010). This approach was chosen because AraSum provides clear and concise summaries, which enhance fine-tuning quality. The PAR task focuses on rephrasing a given text while maintaining its original meaning. For this task, we utilize the Arabic Paraphrased Parallel Synthetic dataset (APPSD) dataset (Al-Shameri and Al-Khalifa, 2024), excluding Arabic Paraphrasing Benchmark (APB) (Alian et al., 2019), which we use for testing. The TRL task involves converting text from one script to another while preserving its pronunciation. It enables non-Arabic speakers to approximate the pronunciation of Arabic words by representing them using Latin letters. We use the BOLT dataset (Shazal et al., 2020b) for this task. We also evaluate the reasoning-based models on the MT task using the Arabic-English split of the United Nations Parallel Corpus v1.0 (UNv1) (Ziemski et al., 2016). Moreover, we evaluate these models on the generative QA task that involves generating natural language answers to questions based on a given context. For this task, we utilize the XTREME benchmark (Siddhant et al., 2020).

Linguistic analysis. This category encompasses POS tagging and (**WSD**) tasks. The PoS tagging task (**PoS**) assigns a grammatical category (e.g., noun, verb, adjective) to each word in a sentence. We use the full Universal Dependencies v2.3 corpus (Nivre et al., 2018) for PoS tagging evaluation experiments. The WSD task identifies the correct meaning of a word with multiple possible senses based on its context. We utilize the Arabic WSD benchmark (El-Razzaz et al., 2021) for this task.
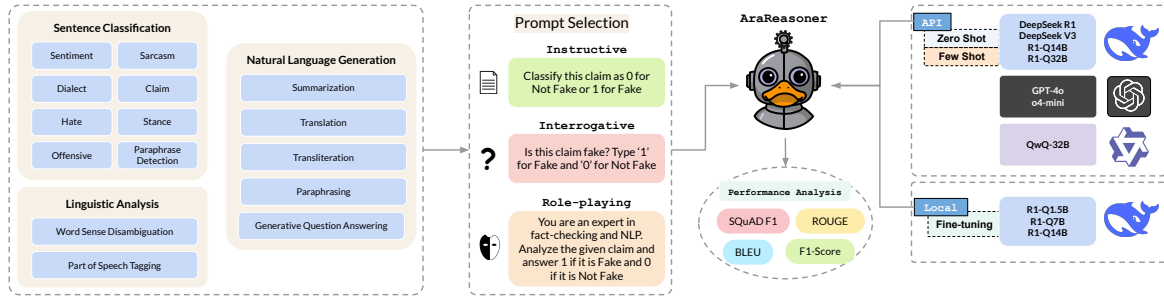
Figure 2: The AraReasoner pipeline.

3.2 Model Selection

This study aims to include a wide range of reasoning-oriented LLMs to comprehensively evaluate their performance on Arabic tasks. Accordingly, we selected a diverse set of models known for their reasoning capabilities, including OpenAI's o4-mini, DeepSeek R1, and four Qwen-based distilled variants: R1-Q1.5B, R1-Q7B, R1-Q14B, and R1-Q32B. We also included QwQ-32B, a reasoning-focused variant from the Qwen family. For comparison, two non-reasoning models, GPT-4o and DeepSeek v3, were also evaluated, which enables us to contrast reasoning and non-reasoning approaches on the same tasks.

Specifically, we employed GPT-4o, o4-mini, DeepSeek-R1, DeepSeek-V3, R1-Q14B, R1-Q32B, and QwQ-32B for both zero-shot and few-shot prompting to evaluate their out-of-the-box reasoning capabilities under minimal supervision. These models were chosen for their scale and diverse architectural foundations, allowing for a broad comparison across reasoning and non-reasoning paradigms. For fine-tuning experiments, we selected R1-Q1.5B, R1-Q7B, and R1-Q14B, balancing model size and training feasibility. This selection enabled us to explore how different capacities within the same model family respond to task-specific supervision, offering insight into the scalability of fine-tuned reasoning performance.

3.2.1 OpenAI GPT Models

We include two proprietary models from OpenAI in our evaluation: *GPT-4o* and *OpenAI-o4-mini* (also referred to as GPT-4-mini or o4-mini). These models represent SOTA commercial LLMs and serve as important baselines for evaluating the capabilities of open-source alternatives.

GPT-4o is OpenAI's flagship model, optimized for performance across a wide spectrum of reasoning, coding, and multimodal tasks. It combines the strengths of GPT-4 with a more efficient architecture, offering faster inference and improved latency. In our experiments, it serves the purpose of providing a point of comparison with non-open-source models.

OpenAI-o4-mini is a smaller, cost-effective reasoning-based variant of GPT-4. Despite its reduced size, it retains strong performance on core reasoning and alignment benchmarks. In our analysis, o4-mini provides a valuable point of comparison for distilled models like DeepSeek-R1-Q32B, which helps to contextualize the effectiveness of scaling and distillation approaches.

3.2.2 DeepSeek

We evaluated the reasoning-based models of the DeepSeek (Guo et al., 2025a), particularly DeepSeek-R1 and its distilled Qwen-based variants. These models aim to advance reasoning in language models through reinforcement learning (RL) and effective distillation.

DeepSeek-R1-Zero is trained purely via large-scale RL without supervised fine-tuning. This model demonstrated emergent reasoning behaviors such as self-verification and reflection (Guo et al., 2025b). However, it struggles with repetition, readability, and language mixing.

DeepSeek-R1 addresses the limitations of DeepSeek-R1-Zero by introducing cold-start supervised fine-tuning before RL. This resulted in improving stability and performance across math, code, and reasoning tasks (Guo et al., 2025b).

DeepSeek-V3 DeepSeek-V3 is a 671B-parameter mixture-of-experts (MoE) model with multi-head latent attention and DeepSeekMoE for efficient training. It uses an auxiliary-loss-free multi-token prediction and load balancing for better performance. Trained on 14.8 trillion tokens with supervised fine-tuning and reinforcement learning, it matches top models with stable training (Liu et al.,

2024b).

Qwen Based Distilled DeepSeek Models We focus on the distilled variants of DeepSeek-R1 that are based on the Qwen2.5 series, including *1.5B*, *7B*, *14B*, and *32B* models. These models transfer reasoning capabilities from DeepSeek-R1 through a distillation process that preserves core performance characteristics while significantly reducing model size. Among them, *DeepSeek-R1-Q32B* has demonstrated competitive performance, reportedly outperforming OpenAI's o1-mini model on several standard benchmarks (Guo et al., 2025b). However, it is important to note that their effectiveness in Arabic NLP tasks remains unexplored.

3.2.3 Qwen

QwQ-32B (Qwen-Team, 2025) is a 32B parameter reasoning model from the Qwen series, developed to tackle complex tasks via advanced RL techniques. This model showed strong reasoning capabilities, achieving competitive results on math, coding, and general benchmarks (Qwen-Team, 2025). Furthermore, it performed on par with much larger models such as DeepSeek-R1 (671B, 37B active) and OpenAI-o1-mini.

4 Evaluation

4.1 Research Questions

In this study, we aim to address the following research questions:

- **RQ1:** How do reasoning-focused LLMs, particularly the DeepSeek family of models, perform on Arabic NLP tasks compared to established models like GPT-4o?

- **RQ2:** What impact do different prompting strategies (zero-shot, 3-shot, and 5-shot) have on model performance across various Arabic NLP tasks?

- **RQ3:** How does model performance vary across different categories of Arabic NLP tasks, particularly between classification tasks and generation tasks?

- **RQ4:** To what extent can model size and architecture influence performance on Arabic NLP tasks?

4.2 Experimental Setup

In our experiments, we accessed the LLMs via APIs and conducted all fine-tuning locally. For fine-tuning, we focused on select models from the DeepSeek family. Appendix B lists the prompt strategies followed in this work, and Appendix D gives more information on inference settings, fine-tuning configurations, and hardware specifications.

4.2.1 Data Split

The datasets were divided into *Train*, *Dev*, and *Test* splits. The *Dev* set was used for prompt selection. *Train* was used for fine-tuning, while *Test* served for final evaluation. See Appendix A for more dataset details.

4.2.2 Evaluation Metrics

In this study, each task is evaluated using well-established metrics that align with the nature and goals of the task. This ensures a comprehensive and fair assessment of model performance across different linguistic and generative tasks.

Sentence Classification tasks are evaluated using the F1-Score, a widely-used metric that balances precision and recall. This is especially appropriate in scenarios where class imbalance is present.

Natural Language Generation tasks are evaluated using metrics that measure the fluency, coherence, and semantic fidelity of the generated text.

For the SUM task, we use the ROUGE-L metric, which evaluates the quality of summaries by comparing the longest common subsequence (LCS) between the generated output and reference summaries. ROUGE-L is particularly effective at capturing sentence-level structure and ensuring that the generated summary retains key content and ordering found in the reference, making it suitable for extractive and abstractive summarization.

For PAR, TRL, and MT tasks, we employ the BLEU score. BLEU measures n-gram overlap between the generated output and reference texts, thereby assessing how closely the system output matches human-written references. While it does not account for semantic equivalence, BLEU is robust at scale and provides reliable corpus-level evaluations, making it especially valuable for comparing multiple model outputs over a large test set.

Finally, for the GQA task, we use the SQuAD-style F1-Score. This metric measures the token-level overlap between the predicted answer and the ground truth, combining both precision and recall. It is especially well-suited for extractive and generative QA settings, ensuring that partial correctness and exact matches are both rewarded.

Table 1: Performance comparison across different tasks using different prompts. Highlighted cells indicate the best performance within the respective row. **INS:** Instructive prompt. **INT:** Interrogative prompt. **RP:** Role-playing. **F1 Score** is used for sentence classification and linguistic analysis; **ROUGE-L** for SUM; **SQuAD-F1** for GQA; **BLEU** for MT, TRL, and PAR.

Task	English			Arabic		
	P1 (INS)	P2 (INT)	P3 (RP)	P4 (INS)	P5 (INT)	P6 (RP)
Sentence Classification						
SA	59.20	35.62	90.66	**90.98**	88.97	88.65
DD	48.53	51.48	48.99	50.43	**69.63**	49.50
HSD	58.73	86.28	86.99	**89.33**	87.66	89.33
OSD	54.99	**85.47**	56.76	83.97	57.15	84.56
SarD	65.89	46.3	47.74	**71.17**	68.16	68.16
CD	27.73	37.74	37.18	**52.87**	37.01	32.56
StD	43.37	29.88	48.59	**63.62**	33.45	63.44
PD	58.01	56.07	58.06	56.01	**83.49**	58.42
Natural Language Generation						
SUM	17.39	**17.41**	15.77	16.24	15.76	15.60
PAR	**13.64**	12.37	8.176	10.22	9.847	4.151
TRL	0.955	1.302	0.443	1.199	1.305	**1.445**
MT	5.31	3.458	5.134	**6.03**	4.813	5.663
GQA	12.4	**17.62**	15.23	11.71	15.66	15.17
Linguistic Analysis						
WSD	35.61	34.96	36.94	**58.43**	38.65	57.99
POS	3.536	3.689	**3.733**	3.363	3.744	3.159

Linguistic Analysis tasks, such as PoS tagging and WSD are evaluated using the F1-Score. This metric provides a balanced evaluation of both accuracy and coverage by integrating precision and recall. It is critical for these tasks where success depends not only on correct identification but also on comprehensive and consistent tagging across the entire dataset.

4.2.3 Prompts Selection

We started the evaluation process by selecting the appropriate prompt for each task using the DeepSeek R1-Q14B model. We designed six prompt variants for each task by crossing three styles with two languages (English and Arabic) as shown in Appendix C. The three prompt styles can be classified as *Instructive*, *Interrogative*, and *Role-playing*. Instructive prompts use an imperative tone, Interrogative prompts take the form of a question, and Role-playing prompts present the instruction as if the model is performing a task in a specific role.

Table 1 presents the performance of the DeepSeek R1-Q14B model in a zero-shot learning setup, using three prompt types written in both English and Arabic. The results highlight the importance of tailoring prompt styles to each task. As shown in the table, the best performance for

most tasks, particularly in SC and LA tasks, was achieved using Arabic instructive prompts. For NLG and some LA tasks, English prompts, especially instructive ones, outperformed the others. Additionally, the lowest performance across nearly all tasks, except for the TRL and PoS tagging tasks, was observed when using role-playing prompts, regardless of language Based on these findings, all subsequent experiments were conducted using the best-performing prompt types(INS, INT, RP) for each task, combined with different few-shot configurations (zero-shot, 3-shot, and 5-shot) to assess the impact of few-shot prompting on task performance.

4.3 Results

This section outlines the performance evaluations of the selected LLMs across a variety of Arabic NLP tasks. The findings are detailed in several tables. Firstly, Table 2 presents the performance data for 15 tasks, comparing leading reasoning models such as R1-671B, QWQ-32B, and GPT 04-mini, under various prompting strategies. Following this, Table 3 illustrates the performance variations across different model sizes within the DeepSeek R1 family for the chosen tasks, also employing diverse prompting strategies. Furthermore, Table 4 offers a comparative analysis of the fine-tuning performance of different R1 model sizes on the selected tasks. Finally, Table 5 contrasts the performance of non-reasoning models, specifically Deepseek V3-685B and GPT-4o, against the reasoning R1-671B model in a zero-shot setting. These tables collectively provide a detailed view of how model choice, size, and prompting techniques influence outcomes on the benchmarked Arabic NLP tasks. Visual representations of these results can be found in Appendix E.

4.3.1 Model Performance Across Task Categories

As outlined in Table 2, GPT o4-mini demonstrated strong capabilities across most Arabic NLP tasks, particularly excelling in SA (90.18%), DD (81.53%), HSD (71.14%), and PD (92.45%) tasks. However, this model struggled most with CD and StD, achieving a maximum F1-score of only 30.63% with zero-shot prompting for CD.

For NLG tasks, DeepSeek models showed competitive performance in MT (BLEU score of 26.39 with 5-shot prompting) but were outperformed by GPT-o4-mini on PAR tasks. The ROUGE-L scores

Table 2: Reasoning models across tasks. DS: Deepseek; ZS: zero-shot; 3S: 3 shots; 5S: 5 shots.

Task	DS R1-671B			QwQ-32B			GPT o4-mini		
	ZS	3S	5S	ZS	3S	5S	ZS	3S	5S
Sentence Classification									
SA	88.59	87.56	87.52	35.30	85.07	87.50	60.24	**90.18**	89.61
DD	69.42	76.76	77.79	43.49	50.20	49.01	80.69	80.50	**81.53**
HSD	63.56	62.46	64.40	39.57	39.58	61.46	**71.14**	66.69	70.11
OSD	87.95	88.72	59.82	85.55	85.50	83.26	**90.28**	88.90	87.90
SarD	59.86	65.81	67.33	22.54	36.46	38.53	66.31	**70.96**	70.32
CD	55.43	63.11	57.72	42.46	62.75	**63.81**	30.63	24.68	19.60
StD	79.36	**95.16**	88.12	72.02	87.12	82.42	76.14	85.37	75.85
PD	88.56	89.32	91.12	56.12	56.60	86.99	91.77	91.85	**92.45**
AVG	74.09	**78.61**	74.23	49.63	62.91	69.12	70.90	74.89	73.42
Natural Language Generation									
SUM	11.70	11.79	12.07	12.37	12.93	**13.85**	12.00	13.20	12.40
PAR	4.76	8.84	10.40	7.37	9.66	9.43	14.77	16.77	**16.80**
TRL	16.99	18.21	18.86	5.58	7.27	7.52	25.18	**28.46**	28.42
MT	24.60	26.05	**26.39**	16.86	25.24	17.01	24.02	25.03	25.00
GQA	19.03	26.60	**29.83**	12.76	17.01	19.30	20.65	23.98	27.77
AVG	15.42	18.30	19.51	10.99	14.42	13.42	19.32	21.49	**22.08**
Linguistic Analysis									
WSD	70.30	71.43	**72.26**	43.84	66.69	44.71	69.83	69.71	70.76
POS	5.27	18.41	**24.07**	3.36	8.81	11.33	6.12	7.64	10.63
AVG	37.79	**44.92**	48.17	23.60	37.75	28.02	37.98	38.68	40.70

Table 3: Performance across different model sizes.

Task	R1-Q14B			R1-Q32B			DS R1-671B		
	ZS	3S	5S	ZS	3S	5S	ZS	3S	5S
Sentence Classification									
SA	82.32	86.93	87.54	86.04	87.38	84.91	**88.59**	87.56	87.52
DD	33.16	70.22	70.87	33.68	48.50	48.98	69.42	76.76	**77.79**
HSD	40.45	59.15	61.49	41.02	62.20	62.23	63.56	62.46	**64.40**
OSD	57.11	55.67	81.95	85.95	85.05	56.31	87.95	**88.72**	59.82
SarD	**69.74**	69.12	46.88	42.14	42.17	64.96	59.86	65.81	67.33
CD	27.43	59.20	56.11	39.97	60.23	41.19	55.43	**63.11**	57.72
StD	24.68	67.58	70.45	71.50	79.99	75.06	79.36	**95.16**	88.12
PD	41.55	52.43	87.51	59.09	87.31	88.43	88.56	89.32	**91.12**
AVG	47.05	65.03	70.35	57.42	69.10	65.25	74.09	**78.61**	74.22
Natural Language Generation									
SUM	9.43	11.37	11.98	10.57	**12.43**	12.01	11.70	11.79	12.07
PAR	7.45	**11.04**	9.18	8.38	10.80	10.30	4.76	8.84	10.40
TRL	1.61	1.86	2.07	3.12	4.08	4.41	16.99	18.21	**18.86**
MT	9.62	9.93	9.68	13.04	15.92	13.41	24.60	26.05	**26.39**
GQA	10.29	14.31	15.14	11.94	14.45	15.92	19.03	26.60	**29.83**
AVG	7.68	9.70	9.61	9.41	11.53	11.21	15.41	18.29	**19.51**
Linguistic Analysis									
WSD	40.43	43.26	43.74	63.47	43.45	66.07	70.30	71.43	**72.26**
POS	3.40	6.10	7.60	3.90	6.00	7.64	5.27	18.41	**24.07**
AVG	21.91	24.68	25.67	33.68	24.725	36.85	37.78	44.92	**48.16**

for SUM were relatively consistent across prompting strategies, indicating that this capability might be more inherent to the model's pre-training rather than influenced by in-context examples.

4.3.2 Impact of Prompting Strategies

Introducing in-context examples as a prompting strategy consistently improves model performance, particularly on classification tasks, though the benefit expands, and in some cases reverses, when too many examples are provided. For instance, as shown in Table 2, QwQ-32B's average F1 on sentence classification tasks leaps from 49.6 % under zero-shot prompting to 62.9 % with three examples and further to 69.1 % with five examples. The most dramatic improvements occur on semantically rich tasks: SA rises from 35.3 % to 87.5 %, PD from 56.1 % to 87.0 %, and CD from 42.5 % to 63.8 % as the number of examples increases. These gains underscore how curated examples help the model internalize task intent and disambiguate complex decisions.

NLG tasks exhibit smaller but reliable benefits from few-shot prompting. DeepSeek R1-671B's SUM performance (ROUGE-L) increases modestly from 11.70 to 12.07, and its MT quality (BLEU) climbs from 24.60 to 26.39 when moving from zero- to five-shot prompts. PAR and GQA follow similar trends, suggesting that while generative models can leverage examples to refine outputs, the magnitude of improvement is constrained by the open-ended nature of these tasks.

In contrast, LA tasks benefit only marginally from additional examples. WSD sees a slight uptick from 70.3 % to 72.3 % F1, and PoS tagging, despite jumping from a mere 5.3 % to 24.1 % F1 with five-shot prompting, still reflects significant room for growth in capturing Arabic morphological nuance. These results indicate that models struggle to generalize fine-grained linguistic phenomena solely through few-shot prompting.

Crucially, our findings reveal diminishing—and sometimes negative—returns from larger context windows. Certain tasks, such as StD and DD, experience performance declines when examples increase from three to five. We hypothesize that an excess of examples may introduce conflicting patterns or over-specify the task, impeding the model's flexibility.

4.3.3 Model Size and Architecture Effects

As demonstrated in Table 3, larger model sizes generally correlate with better performance. The largest DeepSeek R1 variant (671B) consistently outperforms smaller versions across most tasks, with an average performance improvement of 25.17 percentage points compared to the 14B variant in the zero-shot setting. However, the relationship between model size and performance is not strictly linear, as evidenced in Table 2 by the QwQ-32B model underperforming by 10.62 percentage points

Table 4: Comparing finetuned versions of R1-Q models.

Task	1.5B	7B	14B
Sentence Classification			
SA	71.89	86.52	**93.87**
DD	72.88	76.28	**79.92**
HSD	53.72	64.47	**68.75**
OSD	72.63	83.34	**90.59**
SarD	54.52	65.03	**73.60**
CD	51.35	67.47	**68.81**
StD	59.35	86.37	**98.02**
PD	92.71	95.48	**97.21**
AVG	66.13	78.12	**83.85**
Natural Language Generation			
SUM	11.16	12.48	**19.40**
PAR	**25.85**	20.51	20.48
TRL	47.13	52.82	**57.77**
MT	10.48	13.73	**16.97**
GQA	2.14	4.03	**13.40**
AVG	19.35	20.71	**25.60**
Linguistic Analysis			
WSD	76.25	78.15	**86.27**
POS	70.37	87.72	**95.99**
AVG	73.31	82.94	**91.13**

Table 5: Reasoning vs. non-reasoning models in zero-shot mode.

Task	R1-671B	V3-685B	GPT-4o
Sentence Classification			
SA	**88.59**	87.13	86.71
DD	**69.42**	65.80	55.75
HSD	63.56	65.78	**66.21**
OSD	**87.95**	84.93	87.07
SarD	59.86	**70.76**	64.84
CD	55.43	38.34	**59.98**
StD	79.36	**80.75**	35.15
PD	88.56	**89.99**	89.29
AVG	**74.09**	72.94	68.13
Natural Language Generation			
SUM	11.70	12.92	**13.60**
PAR	4.76	5.39	**13.62**
TRL	16.99	20.94	**25.58**
MT	24.60	**25.40**	25.23
GQA	19.03	19.89	**20.36**
AVG	15.42	16.91	**19.68**
Linguistic Analysis			
WSD	**70.30**	44.17	69.05
POS	**5.27**	4.74	3.85
AVG	**37.79**	24.46	36.45

compared to the DeepSeek R1-Q32B despite similar parameter counts.

This suggests that architectural design choices and training methodology play significant roles alongside raw parameter count in determining Arabic language capabilities. The Qwen-based DeepSeek R1-32B variant, for instance, showed stronger performance than might be expected based solely on its parameter count, potentially due to architectural advantages.

4.3.4 Reasoning vs. Non-Reasoning Models

In table 5, when comparing the reasoning R1-671B model to non-reasoning models including GPT-4o and V3-685B, we observe mixed results across different tasks. GPT-4o demonstrated superior performance on most NLG tasks, while R1-671B outperforms the non-reasoning models and achieves competitive results on sentence classification and linguistic analysis with an exception on SarD task.

These results suggest that while GPT-4o maintains advantages in most generation tasks, the reasoning R1-671B model show particular promise for Arabic classification and linguistic analysis tasks, potentially due to differences in pretraining data

composition or architectural design choices optimized for such tasks.

5 Conclusion

In this work, we have delivered the first in-depth benchmark of reasoning-oriented LLMs on 15 Arabic NLP datasets, demonstrating how prompt engineering, model architecture, and scale interact to shape performance across classification, generation, and LA tasks. By showing that just three well-chosen examples can yield double-digit F1 improvements, that DeepSeek's reasoning-focused design outstrips a top GPT o4-mini baseline by 12 points in zero-shot inference, and that LoRA fine-tuning can add up to 8 additional points over sheer scaling, we provide a clear roadmap for maximizing results on morphologically rich, dialectally varied Arabic data. Our comparative analysis also quantifies the remaining 10–15 point gap between reasoning and non-reasoning models, underscoring where future architectures must evolve. Future work will focus on evaluating additional open-source models, exploring methods to enhance performance on specific tasks, and expanding both the range of datasets and the diversity of tasks considered.

6 Limitations

Despite the strengths of this study, several limitations should be acknowledged. First, the evaluation was limited to a subset of available datasets and target tasks, which may not fully capture the linguistic richness and variability of the Arabic language, including its diverse dialects, each with unique characteristics and challenges. Second, only a subset of LLMs was considered, potentially limiting the generalizability of the findings across different model architectures and training paradigms. Third, prompt selection was performed using a single LLM, which may introduce bias. Exploring the impact of prompts on other LLMs is important for a comprehensive evaluation of their performance. Finally, the prompting strategy was limited to three types of prompts. Although this was sufficient to establish baseline comparisons, it may not fully leverage the capabilities of each model. More advanced prompt engineering techniques, such as chain-of-thought prompting or opinion-based prompts, could lead to improved performance and deserve further exploration in future work. Furthermore, while we used a standardized prompt format across all models for consistency, we acknowledge that different models may be optimized for different prompting styles; and that prompt optimization per model is a potential direction for future work.

Acknowledgments

References

Ahmed Abdelali, Hamdy Mubarak, Shammur Absar Chowdhury, Maram Hasanain, Basel Mousi, Sabri Boughorbel, Yassine El Kheir, Daniel Izham, Fahim Dalvi, Majd Hawasly, et al. 2023. Larabench: Benchmarking arabic ai with large language models. *arXiv preprint arXiv:2305.14982*.

Noora Al-Shameri and Hend Al-Khalifa. 2024. Arabic paraphrased parallel synthetic dataset. *Data in Brief*, 57:111004.

Marwah Alian, Arafat Awajan, Ahmad Al-Hasan, and Raeda Akuzhia. 2019. Towards building arabic paraphrasing benchmark. In *Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems*, pages 1–5.

Badr AlKhamissi, Muhammad N ElNokrashy, and Mohamed Gabr. 2020. Deep diacritization: Efficient hierarchical recurrence for improved arabic diacritization. *arXiv preprint arXiv:2011.00538*.

Khaled Mohammad Alomari, Hatem M ElSherif, and Khaled Shaalan. 2017. Arabic tweets sentimental analysis using machine learning. In *International conference on industrial, engineering and other applications of applied intelligent systems*, pages 602–610. Springer.

Nora Saleh Alturayeif, Hamzah Abdullah Luqman, and Moataz Aly Kamaleldin Ahmed. 2022. Mawqif: A multi-label arabic dataset for target-specific stance detection. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 174–184.

Z Alyafeai, MS Alshaibani, B AlKhamissi, H Luqman, E Alareqi, and A Fadel. 2023a. Taqyim: Evaluating arabic nlp tasks using chatgpt models. arxiv 2023. *arXiv preprint arXiv:2306.16322*.

Zaid Alyafeai, Maged S Alshaibani, Badr AlKhamissi, Hamzah Luqman, Ebrahim Alareqi, and Ali Fadel. 2023b. Taqyim: Evaluating arabic nlp tasks using chatgpt models. *arXiv preprint arXiv:2306.16322*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.

Fengxiang Cheng, Haoxuan Li, Fenrong Liu, Robert van Rooij, Kun Zhang, and Zhouchen Lin. 2025. Empowering llms with logical reasoning: A comprehensive survey. *arXiv preprint arXiv:2502.15652*.

Mahmoud El-Haj, Udo Kruschwitz, and Chris Fox. 2010. Using mechanical turk to create a corpus of arabic summaries.

Mohammed El-Razzaz, Mohamed Waleed Fakhr, and Fahima A Maghraby. 2021. Arabic gloss wsd using bert. *Applied Sciences*, 11(6):2567.

AbdelRahim Elmadany, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2022. Orca: A challenging benchmark for arabic language understanding. *arXiv preprint arXiv:2212.10758*.

Ibrahim Abu Farha and Walid Magdy. 2020. From arabic sentiment analysis to sarcasm detection: The arsarcasm dataset. In *The 4th Workshop on Open-Source Arabic Corpora and Processing Tools*, pages 32–39. European Language Resources Association (ELRA).

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025b. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. 2024. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*.

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.

Mram Kahla, Zijian Győző Yang, and Attila Novák. 2021. Cross-lingual fine-tuning for abstractive Arabic text summarization. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 655–663, Held Online. INCOMA Ltd.

Jude Khouja. 2020. Stance prediction and claim verification: An arabic perspective. In *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*, pages 8–17.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024b. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Hamdy Mubarak, Kareem Darwish, Walid Magdy, Tamer Elsayed, and Hend Al-Khalifa. 2020. Overview of osact4 arabic offensive language detection shared task. In *Proceedings of the 4th Workshop on open-source arabic corpora and processing tools, with a shared task on offensive language detection*, pages 48–52.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022. AraT5: Text-to-text transformers for Arabic language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 628–647, Dublin, Ireland. Association for Computational Linguistics.

Vu-Linh Nguyen, Mohammad Hossein Shaker, and Eyke Hüllermeier. 2022. How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning*, 111(1):89–122.

Joakim Nivre, Mitchell Abrams, Željko Agić, and Bengt Ahrenberg. 2018. Universal dependencies 2.3. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Qwen Qwen-Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Haitham Seelawi, Ahmad Mustafa, Hesham Al-Bataineh, Wael Farhan, and Hussein T Al-Natsheh. 2019. Nsurl-2019 shared task 8: Semantic question similarity in arabic. *arXiv preprint arXiv:1909.09691*.

Ali Shazal, Aiza Usman, and Nizar Habash. 2020a. A unified model for arabizi detection and transliteration using sequence-to-sequence models. In *Proceedings of the fifth arabic natural language processing workshop*, pages 167–177.

Ali Shazal, Aiza Usman, and Nizar Habash. 2020b. A unified model for arabizi detection and transliteration using sequence-to-sequence models. In *Proceedings of the fifth arabic natural language processing workshop*, pages 167–177.

Aditya Siddhant, Junjie Hu, Melvin Johnson, Orhan Firat, and Sebastian Ruder. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. In *Proceedings of the International Conference on Machine Learning 2020*, pages 4411–4421.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jingbo Zhu, Huizhen Wang, Benjamin K Tsou, and Matthew Ma. 2009. Active learning with sampling by uncertainty and density for data annotations. *IEEE Transactions on audio, speech, and language processing*, 18(6):1323–1331.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534.

A Data statistics

Table 6 provides an overview of the datasets used in our study, including the task type and the number of examples in each data split. A notable characteristic of many of these datasets is the absence of an official development set. Since prompt-based methods require a separate development set for prompt selection and hyperparameter tuning, we addressed this by randomly sampling 300 examples from the training set to serve as a validation set whenever one was not provided. In addition to prompt selection, the training splits were also used for fine-tuning language models when applicable. For datasets with official validation and test splits, we adhered to the provided divisions. For those without, we ensured that the random sampling for the dev set was done without replacement, preserving a clean separation between training, validation, and test data. This procedure ensured consistency across tasks and datasets, enabling a fair and comparable evaluation setup for both prompt-based and fine-tuned approaches.

B Prompt Strategies

We used zero-shot and in-context learning setups to evaluate all models in our selection pool, including GPT-4o, DeepSeek R1, Qwen-based distilled variants, and QwQ-32B. This comprehensive approach allowed us to assess both the reasoning capabilities of full-scale models and the adaptability of the distilled variants under different prompt conditions. Moreover, we fine-tuned three distilled variants of DeepSeek R1: R1-Q1.5B, R1-Q7B, and R1-Q14B. These fine-tuned models were selected due to their open availability and because larger models like the full DeepSeek R1 presented challenges in fine-tuning with our available computational resources.

Zero-shot learning. In the zero-shot setting, models were evaluated using the default parameters provided by the API. No task-specific examples or demonstrations were included in the prompt. This setup served as a baseline to assess how well the models could perform without prior exposure or task conditioning.

In-Context Learning. For in-context learning, we evaluated models using both 3-shot and 5-shot configurations. Default model parameters from the API were used in this setting as well. To select the samples involved in the in-context learning experiments, we first estimated uncertainty scores using

AraBERT's (Antoun et al., 2020) representations over the training data. The rationale behind using uncertain samples was to expose the model to more informative and challenging instances, which are more likely to enhance reasoning capabilities during inference (Nguyen et al., 2022; Zhu et al., 2009). From the top 20 most uncertain samples, we manually selected 5 examples. These examples were then used for both 3-shot and 5-shot prompts. Manual curation ensured that examples exhibiting high uncertainty due to noise or ambiguity were excluded, focusing instead on those that were difficult yet coherent. From the top 20 most uncertain samples, we manually selected 5 examples. These examples were then used for both 3-shot and 5-shot prompts.

C Prompt Selection

Prompt Template. We standardized our prompt formatting across tasks using a consistent block structure. Each prompt was composed of the following segments in order: a *System Prompt* block, an *Instruction* block, an optional *Example* block (used in few-shot settings), a *Query* block containing the input question or data for the model, and a final *Response* block that the model is expected to complete. These blocks were clearly delimited with '###' to facilitate easy parsing during evaluation. For instance, once an LLM generated its output, we could extract the relevant part by splitting on '### Question'. Our templates are illustrated in Figure 3.

Instruction Selection. For each task, we designed a total of six prompt variants by crossing three styles with two languages (English and Arabic). The three prompt styles were: (1) **Instructive**, where the prompt uses an imperative tone (e.g., *"Classify the sentiment of the following sentence."*); (2) **Interrogative**, where the prompt takes the form of a question (e.g., *"What is the sentiment expressed in this sentence?"*); and (3) **Role-playing**, which frames the instruction as if the model is performing a task in a role (e.g., *"You are a sentiment analysis expert. Analyze the following sentence and determine its sentiment."*). Each of these three variants was written in both English and Arabic, resulting in six total prompts per task. To determine the most effective instruction style for each task, we evaluated performance using either the official dev set (if available) or a 300-example subset randomly sampled from the training set (see

Table 6: Summary of Datasets and Tasks. Most datasets do not include an official validation set; in such cases, we randomly sampled 300 examples from the training set to serve as a development set for the prompt selection stage.

Dataset	Task(s)	Train size	Dev size	Test size	Full size
AJGT	SA	1,440	-	360	1800
ArSarcasm	DD, SarD	8,437	-	2,110	10,547
OSACT4	HSD, OSD	7,000	-	1,000	8,000
ANS-claim	CD	3,185	987	456	4,628
ANS	StD	2,652	756	379	3,787
NSURL-2019	PD	11,997	-	3,715	15,712
AraSum, EASC	SUM	39,683	-	152	39,835
APPSD	PAR	15,769	-	1,010	16,779
BOLT	TRL	60,479	-	6,630	67,109
UNv1	MT	99,999	-	4,000	103,999
XTREME	QA	14,805	-	921	15,726
Universal Dependencies v2.3 corpus	POS	6,075	909	680	7,664
Arabic WSD Benchmark	WSD	24,878	-	6,220	31,098

Figure 3: Prompt templates used in finetuning as well as in context learning setup. Zero-shot templates were used in the fine-tuning experiments as well.

Appendix A). The prompt with the highest score was selected for use in all downstream experiments, including both in-context learning and fine-tuning. See Figure 4 and Figure 5 for the complete list of the instructions we evaluated.

Instruction Evaluation Results. Table 1 summarizes the performance of the six instruction prompts across different tasks. Each row reports the evaluation score for a particular task using each of the six prompt variants: three styles in English (P1–P3) and three in Arabic (P4–P6). As shown, performance varied across both prompt style and language. Arabic prompts frequently outperformed their English counterparts in sentence classification tasks, especially when using an instructive tone (e.g., P4). In general, instructive and interrogative prompts performed better than role-playing prompts in most settings. These findings informed our prompt design decisions in the main experiments.

Selecting Examples for Few-Shot Prompts. To select representative and informative examples for few-shot prompting, we adopted an uncertainty-based sampling strategy. Using AraBERT (Antoun et al., 2020), we computed uncertainty scores across the training set of each task. Prior work has shown that exposing LLMs to more uncertain and challenging examples can improve performance by encouraging more robust reasoning (Nguyen et al., 2022; Zhu et al., 2009). From the top 20 most uncertain samples, we manually selected 5 high-quality examples. These were used in both 3-shot and 5-shot prompt settings. Manual inspection allowed us to discard noisy or ambiguous examples, ensuring that the selected samples were difficult yet coherent and informative.

D Implementation Details

In our experiments, we leveraged two APIs to access the aforementioned LLMs. Specifically, we used the Novita API[1] to access both the DeepSeek and QwQ models, and employed the official OpenAI platform[2] for ChatGPT. All models were queried using their default inference settings, except for the maximum token limit, which we set to 2048 tokens.

For fine-tuning experiments, we focused on a subset of models from the DeepSeek family: R1-Q1.5B, R1-Q7B, and R1-Q14B. We fine-tuned the

models using the Low-Rank Adaptation (LoRA) technique (Hu et al., 2022). In our setup, LoRA was applied to an extended set of projection layers within the Transformer architecture, including the query, key, value, and output projections in the attention mechanism, as well as the gating and feedforward components (specifically, q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, and down_proj). This broader application is motivated by findings from (Hayou et al., 2024), which show that incorporating LoRA into MLP components improves adaptation performance while maintaining parameter efficiency. A LoRA rank of 4 was used. The maximum sequence length was set to 2048 tokens, with a batch size of 2 and gradient accumulation steps of 2, effectively simulating a batch size of 4. Fine-tuning was performed on an NVIDIA RTX A6000 GPU with 48 GB of VRAM. The training ran for up to 100,000 steps using the 8-bit AdamW optimizer with a cosine learning rate scheduler and a learning rate of 2e-4. Mixed-precision training was enabled via FP16 or BF16, depending on hardware support.

E Evaluation Results

Figure 6 compares the normalized performance of the models across four experimental settings: zero-shot, 3-shot, 5-shot, and fine-tuned settings. Figure (a) presents the zero-shot performance when the models are evaluated without being provided with any task-specific examples. The performance varies immensely across tasks, with the largest models generally performing the best due to their better pre-training.

Figure (b) illustrates the 3-shot setting. Providing the model with a few demonstrations improves the performance on the majority of tasks, with greater benefit seen in models with low parameters. This implies that few-shot learning is effective in improving generalization.

Figure (c) presents the 5-shot results. Generally, the trends from the 3-shot remain, with better performance observed in some tasks, particularly those involving classification or structured reasoning. Nonetheless, performance on other tasks degrades relative to 3-shot results.

Figure (d) shows the performance after fine-tuning over a task-specific set. Fine-tuning yields the optimal overall performance for the distilled models, minimizing differences among model sizes and demonstrating the benefit of task adaptation.

Sentence Classification

Sentiment Analysis

P1 (INS): Classify the sentiment of this sentence as 0 for Negative or 1 for Positive

P2 (INT): What sentiment does this text express? Type '1' for Positive and '0' for Negative:

P3 (RP): You are an expert in sentiment analysis and natural language processing. Analyze the given text and answer 1 if the sentiment is Positive and 0 if the sentiment is Negative

✅ **P4 (INS):** صنف مشاعر هذه الجملة كـ0 إذا كانت سلبية و 1 إذا كانت ايجابية

P5 (INT): هل تعبر هذه الجملة عن مشاعر ايجابية ام سلبية؟ اكتب '1' إذا كانت ايجابية و'0' إذا كانت سلبية

P6 (RP): أنت خبير في تحليل المشاعر ومعالجة اللغة الطبيعية. قم بتحليل النص المعطى وأجب بـ 1 إذا كانت المشاعر ايجابية و0 إذا كانت المشاعر سلبية

Dialect Binary

P1 (INS): Classify this text as '0' for Modern Standard Arabic (MSA) or '1' for Dialectal Arabic (DA)

P2 (INT): Is this text written in Modern Standard Arabic (MSA) or Dialectal Arabic (DA)? Type '0' for MSA or '1' for DA

P3 (RP): You are an expert in Arabic dialect identification and natural language processing. Analyze the given text and answer '0' for MSA and '1' for DA.

P4 (INS): صنف هذا النص كـ 0 إذا كان بالفصحى (العربية الفصحى) و 1 إذا كان بالعامية (العربية الدارجة)

✅ **P5 (INT):** صنف هذا النص كـ 0 إذا كان بالفصحى (العربية الفصحى) و 1 إذا كان بالعامية (العربية الدارجة):

P6 (RP): أنت خبير في تصنيف اللهجات العربية ومعالجة اللغة الطبيعية. قم بتحليل النص المعطى وأجب بـ 0 إذا كان النص مكتوب بالعربية الفصحى و 1 إذا كان باللهجة العامية

Hate Speech

P1 (INS): Classify this text as 0 for Not Hate Speech or 1 for Hate Speech

P2 (INT): Does this text contain hate speech? Type '1' for Hate Speech and '0' for Not Hate Speech.

P3 (RP): You are an expert in hate speech detection and natural language processing. Analyze the given text and answer 1 if it contains Hate Speech and 0 if it does not.

✅ **P4 (INS):** صنف هذا النص كـ 0 إذا لم يكن يحتوي على خطاب كراهية و 1 إذا يحتوي على خطاب كراهية

P5 (INT): هل يحتوي هذا النص على خطاب كراهية؟ اكتب '1' إذا كان يحتوي على خطاب كراهية و'0' إذا لم يكن يحتوي على خطاب كراهية

P6 (RP): أنت خبير في اكتشاف خطاب الكراهية ومعالجة اللغة الطبيعية. قم بتحليل النص المعطى وأجب بـ 1 إذا يحتوي على خطاب كراهية و 0 إذا لم يكن يحتوي على خطاب كراهية

Offensive

P1 (INS): Classify this text as 0 for Not Offensive or 1 for Offensive.

✅ **P2 (INT):** Does this text contain offensive language? Type '1' for Offensive and '0' for Not Offensive.

P3 (RP): You are an expert in offensive language detection & natural language processing. Analyze the given text and answer 1 if it contains Offensive Language and 0 if it does not.

P4 (INS): صنف هذا النص كـ 0 إذا لم يكن هجومياً و 1 إذا كان هجومياً.

P5 (INT): هل يحتوي هذا النص على لغة هجومية؟ اكتب '1' إذا كان هجومياً و'0' إذا لم يكن هجومياً

P6 (RP): انت خبير في اكتشاف النصوص الهجومية ومعالجة اللغة الطبيعية. قم بتحليل النص المعطى وأجب بـ 1 إذا كان يحتوي على نص هجومي و 0 إذا لم يكن يحتوي على نص هجومي

Sarcasm

P1 (INS): Classify this text as 0 for Not Sarcastic or 1 for Sarcastic

P2 (INT): Does this text contain sarcasm? Type '1' for Sarcastic and '0' for Not Sarcastic.

P3 (RP): You are an expert in sarcasm detection and natural language processing. Analyze the given text and answer 1 if it contains Sarcasm and 0 if it does not.

✅ **P4 (INS):** صنف هذا النص كـ 0 إذا لم يكن ساخراً و 1 إذا كان ساخراً

P5 (INT): هل يحتوي هذا النص على سخرية؟ اكتب '1' إذا كان ساخراً و'0' إذا لم يكن ساخراً

P6 (RP): أنت خبير في اكتشاف السخرية ومعالجة اللغة الطبيعية. قم بتحليل النص المعطى وأجب بـ 1 إذا كان ساخراً و 0 إذا لم يكن ساخراً!

Claim

P1 (INS): Classify this claim as 0 for Not Fake or 1 for Fake

P2 (INT): Is this claim fake? Type '1' for Fake and '0' for Not Fake.

P3 (RP): You are an expert in fact-checking and natural language processing. Analyze the given claim and answer 1 if it is Fake and 0 if it is Not Fake.

✅ **P4 (INS):** صنف هذا الادعاء كـ 0 إذا لم يكن زائفا و 1 إذا كان زائفا

P5 (INT): هل هذا الادعاء زائف؟ اكتب '1' إذا كان زائفا و'0' إذا لم يكن كذلك.

P6 (RP): أنت خبير في التحقق من المعلومات ومعالجة اللغة الطبيعية. قم بتحليل الادعاء المعطى وأجب بـ 1 إذا كان زائفا و 0 إذا لم يكن كذلك

Stance

P1 (INS): Given two sentences, classify their stance using the following labels: 0 for Disagree, 1 for Agree, and 2 for Unclear or Unrelated.

P2 (INT): Determine the stance between the given two sentences. Choose one: (0) Disagree, (1) Agree, (2) Unclear/Unrelated.

P3 (RP): You are an expert in stance detection & natural language processing. Analyze the given two sentences & classify their stance using the provided labels: 0 for Disagree, 1 for Agree, and 2 for Unclear or Unrelated.

✅ **P4 (INS):** بناءً على الجملتين المعطيتين، صنف موقفهما باستخدام التصنيفات التالية: 0 للاختلاف، 1 للاتفاق، و 2 لغير الواضح أو غير المرتبط

P5 (INT): حدد الموقف بين الجملتين المعطيتين. اختر أحد التصنيفات التالية: (0) اختلاف، (1) اتفاق، (2) غير واضح/غير مرتبط

P6 (RP): أنت خبير في اكتشاف المواقف ومعالجة اللغة الطبيعية. قم بتحليل الجملتين المعطيتين وصنف موقفهما باستخدام التصنيفات التالية: 0 للاختلاف، 1 للاتفاق، و 2 لغير الواضح أو غير المرتبط

Paraphrase Detection

P1 (INS): Determine if the two given sentences are paraphrases. Type '1' if they are paraphrased and '0' if they are not

P2 (INT): Are these two sentences paraphrased? Answer '1' if they are paraphrased and '0' if they are not.

P3 (RP): You are an expert in paraphrase detection & natural language processing. Analyze the two sentences and classify them as '1' if they are paraphrases and '0' if they are not.

P4 (INS): قم بالإجابة بـ 1 إذا تمت إعادة صياغة إحدى الجملتين لتكون مكافئة للأخرى و اجب بـ 0 إذا لم يتم اعادة صياغتهما.

✅ **P5 (INT):** هل تمت إعادة صياغة إحدى الجملتين لتكون مكافئة للأخرى؟ أجب بـ '1' إذا كانتا معادتي الصياغة و'0' إذا لم تكونا كذلك.

P6 (RP): أنت خبير في كشف إعادة الصياغة ومعالجة اللغة الطبيعية. قم بتحليل الجملتين وصنفهما بـ '1' إذا كانتا معادتي الصياغة و'0' إذا لم تكونا كذلك.

Figure 4: Listing of experimented instructions for sentence classification tasks. INS stands for instructive, INT stands for interrogative, and RP stands for role playing. Ticked instruction indicates the final instruction prompt after instruction comparison.

Linguistic Analysis

		Word Sense Disambiguation

P1 (INS): Classify the sentiment of this sentence as 0 for Negative or 1 for Positive

P2 (INT): What sentiment does this text express? Type '1' for Positive and '0' for Negative:

P3 (RP): You are an expert in sentiment analysis and natural language processing. Analyze the given text and answer 1 if the sentiment is Positive and 0 if the sentiment is Negative

✅ **P4 (INS):** صنف مشاعر هذه الجملة كـ0 اذا كانت سلبية و 1 اذا كانت ايجابية

P5 (INT): هل تعبر هذه الجملة عن مشاعر ايجابية ام سلبية؟ اكتب '1' اذا كانت ايجابية و '0' اذا كانت سلبية

P6 (RP): أنت خبير في تحليل المشاعر ومعالجة اللغة الطبيعية. قم بتحليل النص المعطى وأجب بـ 1 إذا كانت المشاعر إيجابية و0 إذا كانت المشاعر سلبية

		Part of Speech Tagging

P1 (INS): Tag each word in the given sentence with its correct part of speech. Choose from: ['NOUN', 'PUNCT', 'ADP', 'NUM', ...].

P2 (INT): What are the correct part-of-speech tags for each word in this sentence? Assign one of the following labels: ['NOUN', 'PUNCT', 'ADP', 'NUM', ...]

P3 (RP): You are an expert in Part-of-Speech (POS) tagging and natural language processing. Analyze the given sentence and assign the correct POS tag to each word from the following set: ['NOUN', 'PUNCT', 'ADP', 'NUM', ...].

P4 (INS): قم بوضع وسم نحوي لكل كلمة في الجملة حسب نوعها الصرفي المناسب. اختر من القائمة التالية: ['NOUN', 'PUNCT', 'ADP', 'NUM', ...].

✅ **P5 (INT):** ما هو النوع الصرفي الصحيح لكل كلمة في هذه الجملة؟ حدد الوسم المناسب لكل كلمة من بين الخيارات التالية: ['NOUN', 'PUNCT', 'ADP', 'NUM', ...']

P6 (RP): أنت خبير في التحليل النحوي ومعالجة اللغة الطبيعية. قم بتحليل الجملة المعطاة وحدد الوسم النحوي المناسب لكل كلمة من القائمة التالية: ['NOUN', 'PUNCT', 'ADP', 'NUM', ...].

Natural Language Generation

		Summarization

P1 (INS): Summarize the following text in one sentence

✅ **P2 (INT):** Can you summarize the following text in one sentence?

P3 (RP): You are a professional text summarizer with expertise in extracting key information. Read the given text and generate a one sentence concise and coherent summary that preserves the main ideas and important details

P4 (INS): لخص النص التالي في جملة واحدة

P5 (INT): هل يمكنك تلخيص النص التالي في جملة واحدة؟

P6 (RP): أنت متخصص في تلخيص النصوص ولديك خبرة في استخراج المعلومات الأساسية. اقرأ النص المقدم وأنشئ ملخصًا موجزًا ومتماسكًا في جملة واحدة يحافظ على الأفكار الرئيسية والتفاصيل المهمة.

		Paraphrasing

✅ **P1 (INS):** Paraphrase the following text while keeping the meaning intact

P2 (INT): Can you paraphrase the following text while preserving the meaning?

P3 (RP): You are a language expert skilled in rewriting text while maintaining its original meaning. Rewrite the following passage using different words and sentence structures while keeping the meaning intact.

P4 (INS): أعد صياغة النص التالي مع الحفاظ على المعنى كما هو.

P5 (INT): هل يمكنك إعادة صياغة النص التالي دون تغيير معناه؟

P6 (RP): أنت خبير لغوي ماهر في إعادة صياغة النص مع الحفاظ على معناه الأصلي. أعد كتابة المقطع التالي باستخدام كلمات وتركيبات جمل مختلفة مع الحفاظ على المعنى

		Transliteration

P1 (INS): Convert the following text written in Arabizi into Arabic script

P2 (INT): Can you convert the following text into Arabic script?

P3 (RP): You are an expert in Arabizi transliteration. Convert the following text from Arabizi into Arabic script.

P4 (INS): حوّل النص التالي المكتوب بالحروف اللاتينية وفقا لأسلوب العربيزي إلى الحروف العربية

P5 (INT): هل يمكنك تحويل النص التالي إلى الحروف العربية؟

✅ **P6 (RP):** أنت خبير في تحويل النصوص المكتوبة بالأحرف اللاتينية وفقا لأسلوب العربيزي. حوّل النص التالي إلى الحروف العربية

		Translation

P1 (INS): Translate the following English sentence into Arabic

P2 (INT): What is the translation of this sentence from English to Arabic?

P3 (RP): You are a professional translator fluent in both English and Arabic. Translate the following sentence into Arabic while maintaining proper grammar and context:

✅ **P4 (INS):** ترجم الجملة الإنجليزية التالية إلى اللغة العربية

P5 (INT): ماهي ترجمة هذه الجملة باللغة الانجليزية الى العربية؟

P6 (RP): أنت مترجم محترف تتقن اللغتين الإنجليزية والعربية. ترجم الجملة التالية إلى اللغة العربية مع الالتزام بالقواعد اللغوية الصحيحة والحفاظ على السياق السليم.

		Question Answering

P1 (INS): Generate an informative answer for the following question

✅ **P2 (INT):** What is the answer for the following question?

P3 (RP): You are an advanced knowledge-based AI trained in answering general questions across multiple domains. Provide an accurate, well-structured, and informative response to the following question.

P4 (INS): قم بإنشاء إجابة توضيحية للسؤال التالي

P5 (INT): ماهي الاجابة لهذا السؤال؟

P6 (RP): أنت ذكاء اصطناعي متقدم قائم على المعرفة ومدرب على الإجابة على الأسئلة العامة عبر مجالات متعددة. قدم إجابة دقيقة، ومنظمة للسؤال التالي.

Figure 5: Listing of experimented instructions for Linguistic Analysis and Natural Language Generation tasks. INS stands for instructive, INT stands for interrogative, and RP stands for role playing. Ticked instruction indicates the final instruction prompt after instruction comparison.
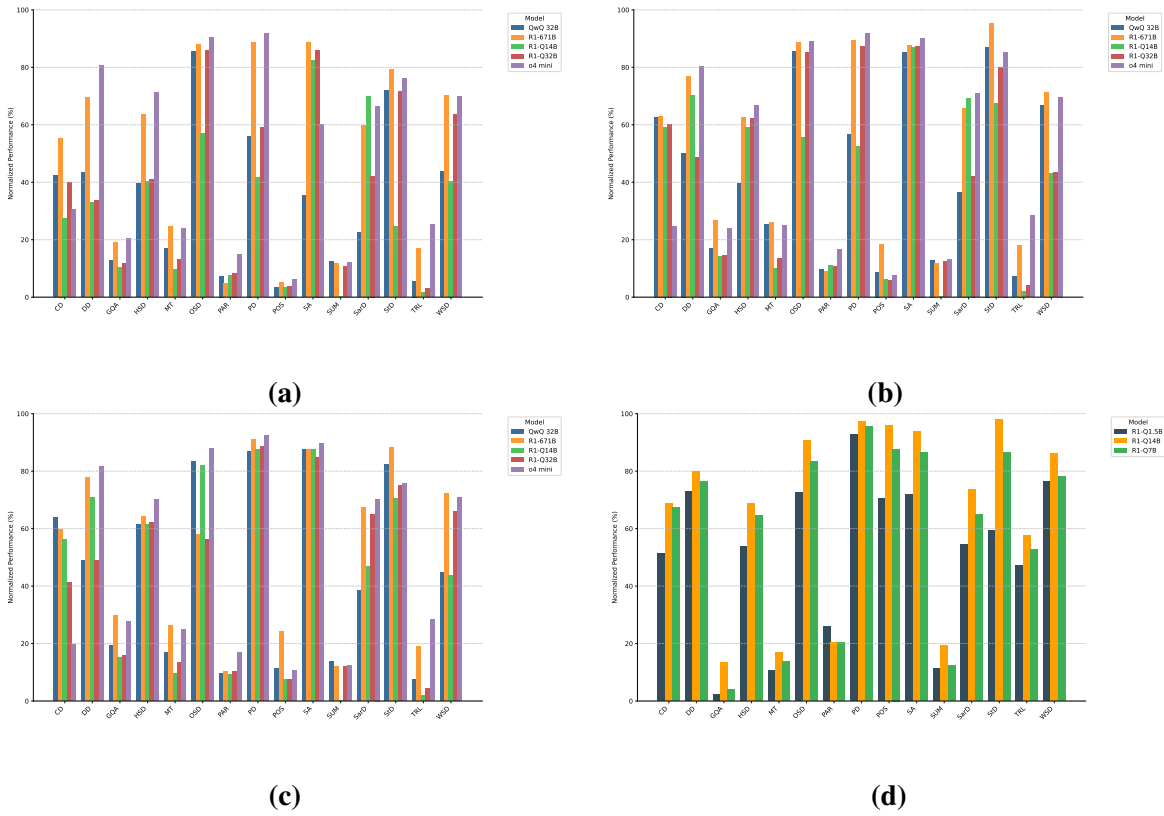
(a)

(b)

(c)

(d)

Figure 6: Normalized performance of the models across different settings: (a) zero-shot, (b) 3-shot, (c) 5-shot, and (d) fine-tuned.