Beyond MCQ: An Open-Ended Arabic Cultural QA Benchmark with Dialect Variants

Hunzalah Hassan Bhatti, Firoj Alam
Qatar Computing Research Institute, Qatar
fialam@hbku.edu.qa, hunzalahhassan@gmail.com

Abstract

Large Language Models (LLMs) are increasingly used to answer everyday questions, yet their performance on culturally grounded and dialectal content remains limited across languages and their varieties. We propose a comprehensive method that *(i)* translates Modern Standard Arabic (MSA) multiple-choice questions (MCQs) into English and several Arabic dialects, *(ii)* converts them into open-ended questions (OEQs), *(iii)* benchmarks a range of zero-shot and fine-tuned LLMs under both MCQ and OEQ settings, and *(iv)* generates chain-of-thought (CoT) rationales to fine-tune models for step-by-step reasoning. Using this method, we extend an existing dataset in which QAs are parallelly aligned across language varieties, making it, to our knowledge, the *first* of its kind. A large portion of the resulting test set is further validated through targeted human annotation and native-speaker post-editing. We conduct extensive experiments with both open and closed models. Our findings show that *(i)* models underperform on Arabic dialects, showing persistent gaps in culturally grounded and dialect-specific knowledge; *(ii)* Arabic-centric models perform well on MCQs but struggle with OEQs; and *(iii)* CoT improves judged correctness while yielding mixed n-gram-based metrics.

Keywords: Cultural Knowledge; Everyday Knowledge, Open-Ended Question, Chain-of-Thought

1. Introduction

Cultural information underpins human identity, behavior, and social interaction, encompassing shared beliefs, values, customs, languages, traditions, and collective practices. In today's tightly coupled information-communication ecosystem, hundreds of millions of users interact with LLMs for everyday queries, often asking about local norms, holidays, cuisine, or etiquette, where culturally grounded interpretations are essential (Pawar et al., 2025; Hasan et al., 2025). Yet despite rapid progress in multilingual understanding and reasoning, LLM performance remains uneven across languages, dialects, and culturally specific domains (Wei et al., 2022; Muennighoff et al., 2023). The issue is especially salient for Arabic, where MSA coexists with numerous regional dialects that differ in phonology, morphology, lexicon, and usage (Alwajih et al., 2025a; Sadallah et al., 2025). Beyond modeling challenges, widely used MCQ evaluations can mask deficiencies in reasoning by enabling superficial answer-selection strategies such as label bias or option-guessing, complicating fair cross-lingual and cross-format comparison (Raman et al., 2025; Li et al., 2024b).

A central open problem is how to *measure* and *improve* an LLM's ability to understand and generate responses to such culturally embedded queries, especially in multilingual settings with substantial dialectal variation. Another noteworthy aspect is that MCQs have long been the dominant format for evaluating QA performance in LLMs due to their simplicity, automatic scoring, and structured answer space (Myrzakhan et al., 2024). However, models

Figure 1: Example QA instances shown in two formats: multiple-choice question (MCQ) and open-ended question (OEQ). Flags in parentheses indicate representative countries where each dialect is widely spoken.

can sometimes exploit the test format rather than genuinely understanding the question, leading to a form of selection bias, for instance, consistently favoring certain options (e.g., always choosing "A") regardless of content.

To address these challenges, parallel efforts have emerged to develop culturally aligned language models (Wang et al., 2023) and to enable their efficient deployment in low-compute environments (Hu et al., 2022). At the same time, new culturally relevant datasets, targeted benchmarks, and evaluation protocols are beginning to operationalize the measurement of everyday cultural knowledge (Myung et al., 2024; Li et al., 2024a; Mousi et al.,

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 5215–5231
11–16 May 2026.

2025; Alam et al., 2025a,b). Collectively, these trends demonstrate the need for new resources, evaluations, and models that are grounded in underrepresented dialectal varieties and culturally contextualized content.

To shade a light on the challenges, we introduce a comprehensive method for developing a new resource for under-representative language verities. Starting from an existing MSA MCQ dataset (Alwajih et al., 2025b), we perform the following steps: *(i)* translate the questions into several Arabic dialects and English, which were then manually post-edited *(ii)* convert the MCQs into OEQ that require free-form answers, *(iii)* evaluate a range of zero-shot and fine-tuned LLMs on the resulting benchmark, and *(iv)* create and fine-tune models on chain-of-thought (CoT) annotations to encourage explicit reasoning for OEQ. An example of MCQ, OEQ with CoT is shown in Figure 1.

Our approach allows us to isolate and study the impact of question format, language variety, and reasoning supervision on model performance. We find that OEQ settings present greater challenges than MCQ, especially in dialectal Arabic. Our contributions are as follows:

- We construct a multilingual and multidialectal QA dataset, ***ArabicCulturalQA***, by translating MSA MCQs into English and Arabic dialects. The dataset is publicly available for research use.[1]
- We convert the dataset into OEQs in all language variants, enabling a more rigorous evaluation of model knowledge.
- A substantial portion of the test set is human annotated by native speakers: dialectal MCQs are post-edited, and the conversion from MSA MCQs to MSA OEQs is manually reviewed to ensure linguistic and semantic fidelity.
- We benchmark a range of zero-shot and fine-tuned LLMs under both MCQ and OEQ settings.
- We generate chain-of-thought (CoT) annotations for OEQ and fine-tune models.

This work represents the *first* effort to unify dialectal Arabic QA, open-ended reasoning, and CoT fine-tuning in a single benchmark, offering new insights into LLM performance on culturally rich, linguistically diverse data.

2. Related Work

2.1. General Capabilities of LLMs.

LLMs have shown strong generalization across a broad range of NLP tasks, including text generation, translation, summarization, and reasoning (Abdelali et al., 2024). At sufficient scale, LLMs exhibit *emergent abilities*, such as multi-step inference and commonsense reasoning (Bubeck et al., 2023; Wei et al., 2022). Prompting techniques like few-shot and chain-of-thought (CoT) significantly enhance performance on reasoning-heavy tasks (Kojima et al., 2022; Wei et al., 2022). However, most evaluations focus on English or high-resource languages. Performance often degrades on morphologically rich or low-resource languages such as Arabic, particularly in dialectal contexts (Mousi et al., 2025; Muennighoff et al., 2023).

2.2. Cultural and Everyday Knowledge.

Recent research has highlighted the limitations of LLMs in capturing culturally grounded, everyday knowledge. Myung et al. (2024) introduced BLEnD, a multilingual benchmark comprising 52.6K QA pairs across 13 languages and 16 regions, designed to evaluate models' understanding of daily-life knowledge. Similarly, Hasan et al. (Hasan et al., 2025) developed MultiNativQA, featuring 64K QA pairs covering nine locations in seven languages. Across these studies, results consistently show that LLMs underperform on questions reflecting underrepresented cultures, often reflecting Western-centric norms. In the Arabic context, Sadallah et al. (2025) proposed ArabCulture, a benchmark of 3.5K MSA-based MCQs curated by native speakers from 13 Arab countries to assess culturally specific commonsense reasoning. Likewise, Alwajih et al. (2025a) introduced Palm, a dialect-rich dataset encompassing all 22 Arab countries.

2.3. MCQ to OEQ.

Many evaluation benchmarks use MCQs because they allow straightforward automatic scoring, in which the model selects an option (A/B/C/D) that can be directly compared with the correct answer. However, recent studies show that this format may introduce artificial performance gains and mask a model's actual reasoning ability (Molfese et al., 2025; Chandak et al., 2025; Myrzakhan et al., 2024). For instance, LLMs often display a *selection bias*, favoring certain options (e.g., consistently choosing "A") due to training artifacts. To mitigate these issues, several works propose converting MCQs into OEQs that require the model to generate answers without predefined choices (Myrzakhan et al., 2024). This forces reliance on internal knowledge and reasoning rather than elimination or guessing. Yet, this conversion introduces new challenges: some MCQs become ambiguous once options are removed, and others may yield multiple valid answers unless carefully rephrased. Moreover, evaluating free-form responses is inherently

[1] QCRI/ArabicCulturalQA

harder, as correctness depends on comparing generated text with gold answers that may differ in wording. Prior work addresses this by using LLM-based evaluation pipelines (e.g., GPT-4) to judge open-ended answers against human references with high reliability (Myrzakhan et al., 2024). Overall, shifting from MCQ to open-ended formats holds promise for revealing deeper model understanding, but it demands careful question selection and robust evaluation protocols.

2.4. Chain-of-Thought (CoT) Reasoning.

CoT prompting has emerged as a powerful technique for enhancing reasoning capabilities in LLMs. Instead of producing an answer directly, the model is encouraged to generate an explicit, step-by-step reasoning path before reaching a final conclusion (Wei et al., 2022). By articulating these intermediate steps, models can decompose complex problems into manageable components, leading to substantial gains in accuracy. Remarkably, even without task-specific training, simply prefixing the prompt with *"Let's think step by step"* can induce this behavior in sufficiently large models, a method known as *zero-shot CoT* (Qin et al., 2023). This simple prompting strategy has demonstrated significant improvements across a wide range of reasoning tasks, including mathematical problem solving and commonsense reasoning. Furthermore, Qin et al. (2023) introduced a *self-consistency* mechanism, in which the model generates multiple reasoning chains and selects the most frequent answer, further enhancing performance. While most existing studies emphasize inference-time CoT, recent research has explored *CoT fine-tuning* to transfer reasoning skills to smaller or multilingual models (Puerto et al., 2025). However, to the best of our knowledge, no prior work has applied CoT fine-tuning to Arabic open-ended QA datasets, particularly those covering dialectal varieties, which constitutes a key contribution of our study.

3. Datasets

Our data, ***ArabicCulturalQA***, is based on the **PalmX 2025 - General Culture Evaluation (PalmX-GC)** dataset, which assesses a model's understanding of Arab culture, including customs, history, geography, arts, cuisine, notable figures, and everyday life across the 22 Arab countries. All questions and answers are written in *MSA* and *Manually Verified*, providing a high-quality benchmark for culturally grounded QA (Alwajih et al., 2025b). The dataset comprises 2,000 training, 500 development, and 2,000 test examples, all in MCQ format. We use PalmX-GC as the basis for creating dialectal MCQ and OEQ variants. Figure 2 illustrates the dataset construction process, in which we used LLMs (specifically GPT-4.1) for translation and data conversion. We chose this model based on its reliability and our paid access.

3.1. Dialectal MCQ

To broaden cultural and linguistic coverage beyond MSA, we translate PalmX into four Arabic dialects such as Egyptian, Levantine, Gulf, and Maghrebi and into English using GPT-4.1, followed by quality checking. We selected these dialects because *(i)* they cover the largest speaker populations and broadest geographic span in the Arab world, *(ii)* capture major points on the Arabic dialect continuum, and *(iii)* represent the main language of everyday communication and online discourse. Including English serves two purposes: it provides a shared reference baseline for cross-lingual comparison, helping disentangle language modeling from culture-specific knowledge, and reflects real usage, where users often ask culturally grounded questions in English about Arabic contexts. This design allows us to probe *(a)* format sensitivity (MCQ→OEQ), *(b)* dialect sensitivity (MSA vs. regional varieties), and *(c)* cross-lingual transfer (Arabic↔English) within a single controlled benchmark.

We employed controlled prompting to translate each MSA MCQ into four dialects and *English*. The prompts explicitly enforced semantic equivalence while allowing lexical and stylistic adaptation to dialectal norms. This approach ensured that the dialectal phrasing preserved the original question's intent without causing any semantic drift from its MSA counterpart.

3.2. MCQ to OEQ

We converted the MSA MCQs into OEQs using `GPT-4.1`. Each MCQ was transformed into a natural QA pair by rephrasing the original question and its correct option into a single, self-contained QA instance. The remaining distractors were used only to guide contextual understanding but were excluded from the final prompt. We filtered out QA items where conversion was structurally infeasible, such as questions dependent on visible alternatives, to avoid ill-posed or underspecified open-ended forms. This process ensured that the resulting OEQs were faithful derivations of verified MCQs rather than arbitrary generations.

3.3. Dialectal OEQ

We then translated the OEQs into dialectal variants using similar controlled prompting, encouraging natural dialectal expression while preserving the semantic and pragmatic meaning of the original MSA version. The resulting dataset forms a

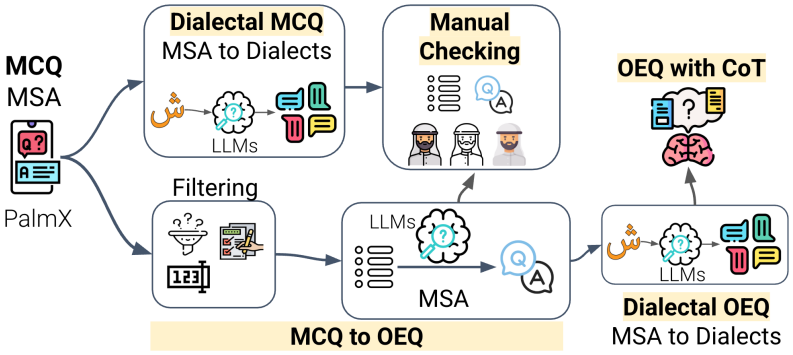

Figure 2: Pipeline for the dataset construction process.

parallel corpus across five Arabic varieties and English, aligned around the same cultural content and verified for equivalence. This structure enables systematic evaluation of dialectal reasoning and cross-variant transfer in generative settings.

3.4. OEQ with CoT

Inspired by prior work (Yu et al., 2025; Zelikman et al., 2022), we transform each OEQ instance $x = (q, a^\star) \in$ dataset D, where q denotes the question and $a^\star$ the gold or reference answer, into one or more CoT training samples using a four-stage pipeline. The pipeline generates multiple reasoning chains without revealing the gold answer, optionally produces gold-conditioned rationalizations, and verifies accepted chains. While generating CoTs, we also prompt the LLM to classify each q as either *factual* or *subjective*. Identifying the question type enables type-specific model development and evaluation. For instance, factual questions may require reference evidence or source attribution for their answers.

Preliminaries. Let $N = \texttt{samples}$ be the number of chain attempts, $T = \texttt{rationalize\_target}$ the minimum number of gold-aligned chains to retain, and $\rho = \texttt{accept\_ratio} \in (0, 1]$ the acceptance threshold. For each attempt $i \in \{1, \ldots, N\}$, we obtain $(\hat{c}_i, \hat{a}_i, \hat{\ell}_i)$, where $\hat{c}_i \in \mathcal{C}$ is a generated chain-of-thought, $\hat{a}_i \in \mathcal{Y}$ is the generated answer, and $\hat{\ell}_i \in \mathcal{L} = \{\text{factual}, \text{subjective}\}$ is a label. Let also denote the collection of attempts as

$$\mathcal{S} = \{(\hat{c}_i, \hat{a}_i, \hat{\ell}_i)\}_{i=1}^{N}, \qquad \mathcal{K} \subseteq \mathcal{S}\ (\textit{accepted subset}).$$

Acceptance is determined via a matching function $\text{match} : \mathcal{Y} \times \mathcal{Y} \to \{0, 1\}$ (see MATCH) against the gold answer $a^\star$, with indicator

$$m_i = \mathbb{I}\{\text{match}(\hat{a}_i, a^\star) = 1\}.$$

We enforce $|\mathcal{K}| \geq T$ and the empirical acceptance ratio $|\mathcal{K}|/N \geq \rho$.

1. CoT generation. Let $\mathcal{G} : \mathcal{Q} \to \mathcal{C} \times \mathcal{Y} \times \mathcal{L}$ denote the rationale-answer-label generator. For each $q \in \mathcal{Q}$, we sample $(\hat{c}_i, \hat{a}_i, \hat{\ell}_i) = \mathcal{G}(q)$. For each attempt, compute the match flag

$$m_i = \mathbb{I}\{\text{match}(\hat{a}_i, a^\star) = 1\},$$

and collect the kept subset

$$\mathcal{K} = \{(\hat{c}_i, \hat{a}_i, \hat{\ell}_i) \in \mathcal{S} : m_i = 1\}.$$

2. CoT rationalize with gold. Let $\mathcal{R} : \mathcal{Q} \times \mathcal{Y} \to \mathcal{C} \times \mathcal{Y} \times \mathcal{L}$ denote the gold-conditioned rationalizer. If $|\mathcal{K}| < T$ (as obtained in Step 1), we draw additional chains via $(\tilde{c}_j, \tilde{a}_j, \tilde{\ell}_j) = \mathcal{R}(q, a^\star)$ and retain those that match the gold answer, with

$$\tilde{m}_j = \mathbb{I}\{\text{match}(\tilde{a}_j, a^\star) = 1\},$$

$$\widetilde{\mathcal{K}} = \{(\tilde{c}_j, \tilde{a}_j, \tilde{\ell}_j) : \tilde{m}_j = 1\}.$$

We then update $\mathcal{K} \leftarrow \mathcal{K} \cup \widetilde{\mathcal{K}}$ until $|\mathcal{K}| \geq T$ (and, if applicable, $|\mathcal{K}|/N \geq \rho$). This stage ensures a sufficient pool of gold-aligned CoT for downstream task.

3. Verification. Let $\mathcal{V} : \mathcal{Q} \times \mathcal{Y} \times \mathcal{C} \times \mathcal{Y} \to [0, 1] \times \{\text{pass}, \text{fail}\} \times \text{Report}$ denote the verifier, where the two $\mathcal{Y}$ components correspond to the gold answer $a^\star$ and the candidate answer a_k, respectively. For each retained item $(c_k, a_k, \ell_k) \in \mathcal{K}$, compute

$$(\sigma_k, \nu_k, r_k) = \mathcal{V}(q, a^\star, c_k, a_k),$$

where $\sigma_k \in [0, 1]$ is a confidence score, $\nu_k \in \{\text{pass}, \text{fail}\}$ is the verdict under a default threshold $\tau = 0.8$ (i.e., $\nu_k = \text{pass}$ iff $\sigma_k \geq \tau$), and r_k is a brief issue report. We then form the verified subset

$$\mathcal{K}_{\text{ver}} = \{(c_k, a_k, \ell_k) \in \mathcal{K} : \nu_k = \text{pass}\},$$

noting that ℓ_k is carried forward but not used by $\mathcal{V}$.

Answer matching *match*. We follow very weak answer matching approach. Given a generated answer $\hat{a}$ and gold $a^\star$, define $\text{match} : \mathcal{Y} \times \mathcal{Y} \to$

$\{0, 1\}$ by match$(\hat{a}, a^\star) = 1$ iff at least one holds: *(i)* exact normalized equality, $\mathrm{norm}(\hat{a}) = \mathrm{norm}(a^\star)$; *(ii)* high token Jaccard,

$$J(P, G) = \frac{|P \cap G|}{\max(1, |P \cup G|)} \geq 0.75,$$

where $P = \mathrm{tokset}(\hat{a})$ and $G = \mathrm{tokset}(a^\star)$; *(iii)* small-set containment,

$$\big(|P| \leq 6 \wedge P \subseteq G\big) \ \vee \ \big(|G| \leq 6 \wedge G \subseteq P\big);$$

(iv) high character similarity,

$$\mathrm{sim}(\mathrm{norm}(\hat{a}), \mathrm{norm}(a^\star)) \geq \tau,$$

with $\tau = 0.88$ and sim computed sequence matching algorithm[2] otherwise, match$(\hat{a}, a^\star) = 0$.

To facilitate the answer matching, we use language-aware normalization $\mathrm{norm}(\cdot)$. For Arabic, we remove diacritics, and drop non-{Arabic letters/digits/_} characters. For non-Arabic, we apply unicode normalization, lowercase, and remove non-{a–z, 0–9} characters. We set $\mathrm{tokset}(s) =$ set$(\mathrm{norm}(s)$ split on spaces$)$.

3.5. Manual Checking and Annotation

3.5.1. Preliminary Annotation

We first conducted a targeted manual evaluation on small samples from each task, including dialectal translation and MCQ→OEQ conversion. For each dialect, one native Arabic speaker who was also fluent in English reviewed the items. Annotators participated on a voluntary basis. Because this initial phase involved only one annotator per dialect, it served mainly as a lightweight quality check, but it still provided an early indication of data quality before larger-scale annotation.

To assess the generated items, we used a set of complementary rubrics covering both linguistic quality and task validity. Specifically, the rubrics evaluate: *(i) dialectal naturalness*, to measure whether the text sounds appropriate and idiomatic in the target dialect; *(ii) meaning preservation*, to check consistency with the source; *(iii) logical coherence*, to identify ill-formed, inconsistent, or contextually inappropriate items; *(iv) question-type appropriateness*, to ensure valid MCQ→OEQ conversion; and *(v) linguistic quality and clarity*, to assess grammar, wording, and readability. The rubrics are defined as follows.

- **Dialectal naturalness:** Do the question and, when applicable, the options sound fluent, idiomatic, and appropriate in the target dialect?
- **Meaning preservation:** Does the OEQ or translation convey the same meaning and intent as the original MCQ or source item?
- **Logical coherence:** Are the question and, when applicable, the options logically consistent, factually sound, and contextually appropriate?
- **Question-type appropriateness:** Is the format suitable for the content (i.e., MCQs are answerable by selection, while OEQs are genuinely open-ended)?
- **Linguistic quality and clarity:** Are grammar, wording, and orthography correct and easy for native speakers to understand?

We rate each dimension on a five-point Likert scale (from 1 to 5), which offers enough granularity to capture meaningful differences, includes a neutral midpoint for ambiguous cases, and supports simple aggregation across annotators and tasks.

Table 1 summarizes the initial manual annotation results for dialectal MCQs and MSA OEQs derived from MCQs. Overall scores are high, with an average of 4.4, indicating strong naturalness, meaning preservation, and linguistic quality. Maghrebi achieved the highest average score of 4.8, while English scored slightly lower at 4.1. The MSA MCQ→OEQ transformation also performed well, with an average of 4.3, suggesting that the generated OEQs generally preserve the meaning and intent of the original MCQs.

Metric	MSA	Lv	Eg	Gf	En	Mg
Dialectal naturalness	4.2	4.4	4.3	4.3	4.2	4.7
Meaning preservation	4.6	4.6	4.3	4.3	4.0	4.7
Logical coherence	4.2	4.4	4.3	4.3	4.0	4.8
Question-type appropriateness	4.1	4.7	4.3	4.4	4.2	4.8
Linguistic quality and clarity	4.2	4.5	4.4	4.4	4.0	4.8
Average	**4.3**	**4.5**	**4.3**	**4.4**	**4.1**	**4.8**

Table 1: Average Likert score from manual annotations on a sample of 50 dialectal MCQs and MSA OEQs derived from MSA MCQs. Lv: Levantine, Eg: Egyptian, Gf: Gulf, En: English, Mg: Maghrebi.

3.5.2. Full Scale Annotation

Following the preliminary annotation, we expanded our annotation task to the test set. Specifically, three independent annotators evaluated all MSA MCQ→MSA OEQ conversions, assessing each converted **Question** across *clarity*, *naturalness in MSA*, *being self-contained*, and *appropriate scope*, and each corresponding **Answer** across *correctness with respect to the original MCQ answer*, *completeness*, *conciseness*, and *fluency*. Beyond Likert ratings, annotators also indicated whether revisions were needed and could provide optional comments when issues arose. This process helps verify that the generated OEQs remain consistent with the source MCQs while forming coherent and meaningful open-ended questions..

[2] https://docs.python.org/3/library/difflib.html

For the **dialectal translation**, we performed full post-editing of the MSA MCQ→Dialectal MCQs. Each dialectal MCQ test set was post-edited by a native speaker of the target dialect, who adapted the content to improve naturalness and linguistic authenticity while preserving the original meaning. This step ensures that the final dialectal MCQs reflect authentic usage patterns rather than literal translations. Detailed annotation guidelines and examples are provided in Appendix 9.1.

4. Experiments

4.1. Models.

For the experiments, we used a range of open and closed-source multilingual and Arabic-centric models, covering capacities from small open frontier to frontier models. The models include *Falcon3-10B-Instruct* (Malartic et al., 2024), *NileChat-3B* (Mekki et al., 2025), *Fanar-1-9B-Instruct* (Team et al., 2025), *Qwen2.5-3B* and *Qwen2.5-7B* (Wang et al., 2024), *GPT-4.1* and *GPT-5* (OpenAI, 2025), and *ALLaM-7B-Instruct-preview* (Bari et al., 2025). This selection covers both high-performing proprietary and open models under 10B parameters, suitable for controlled fine-tuning and reproducible evaluation.

4.2. Benchmarking.

All models were evaluated in a zero-shot setting across multiple language varieties. Prior work on cross-lingual prompting (Kmainasi et al., 2025) has shown that non-native (English) prompts consistently outperform native prompts in reasoning and factual tasks, even for Arabic-centric models, while mixed prompts yield intermediate results. Following these findings, all evaluations in this study were conducted using English prompts. All prompts used are provided in Appendix 9.2.2.

All evaluations were conducted on the automatically generated dataset prior to the human post-editing and annotation stage.

4.3. Training.

We adopt fine-tuning configurations consistent with prior work on Arabic cultural QA tasks, as reported in (Bhatti et al., 2025). Fine-tuning is conducted over 3 epochs using LoRA adapters (Hu et al., 2022), with a maximum sequence length of 512 for MCQ training and 2048 for OEQ training. The learning rate is set to 2×10^{-4}, with a LoRA rank of 64 and $\alpha = 16$. All models are fine-tuned for MCQ evaluation, while only `ALLaM-7B-Instruct-preview` is fine-tuned for the OEQ task.

4.4. Evaluation and Metrics.

For MCQ, we report accuracy, which is an standard metric for MCQ. For OEQ, we employ semantic evaluation using BERTScore (Zhang et al., 2020) and ROUGE-L (Lin, 2004) to assess precision, recall, and overall semantic overlap with the gold answers. Arabic responses are evaluated using `arabert-v2` (Antoun et al., 2020), and English responses with `bert-base-uncased`. This setup allows direct comparability between multilingual and dialectal outputs across all evaluated models. Additionally, for OEQ, we use GPT-4.1 as LLM-as-judge following MT-Bench (Bai et al., 2024), where responses are rated on a 1 to 10 rubric (helpfulness, relevance, accuracy, faithfulness).

5. Results

We compare performance across four conditions: *(i)* MCQ base vs. fine-tuned, *(ii)* OEQ base, *(iii)* OEQ fine-tuned without CoT, and *(iv)* OEQ fine-tuned with CoT. Tables 2, 3, and 4 present the results for the MCQ, OEQ, and OEQ (with vs. without CoT) evaluations, respectively.

Performance Gap for MCQ. As presented in Table 2, the average performance among the Arabic language variants is relatively higher for MSA across open models, followed by Gulf, Egyptian, and others. The average performance for English is higher compared to Arabic across open models, mainly due to the strong performance of non-Arabic-centric models such as Falcon and Qwen. The average performance for Arabic-centric models in the base and fine-tuned (FT) settings is 64.09% and 69.38%, respectively.

The performance of closed models (i.e., GPT*) are higher than closed models in all language variants. The MCQ performance for MSA is highly comparable with the PalmX shared task results where top-system achieved an accuracy of 72.15% (Alwajih et al., 2025b).

Among the smaller open models (i.e., size 3B), in the base setting, `NileChat-3B` achieves the highest average accuracy of 65.43, while `Fanar-1-9B-Instruct` is the best-performing fine-tuned model with an accuracy of 71.01. Among the open models, the fine-tuned `ALLaM-7B-Instruct` performs best for Egyptian and Maghrebi, whereas `Fanar-1-9B-Instruct-FT` achieves the highest performance for MSA, Le, Gf, and En.

Performance Gap for OEQ. Across language variants, we observe a pattern consistent with MCQ results: the average F1 for MSA exceeds that of other Arabic dialects; however, the gap is smaller

Model	MSA	Egyptian	Levantine	Magrebi	Gulf	English	Average
Falcon3-10B-Instruct	46.05	43.95	44.10	42.70	45.15	66.50	48.48
Falcon3-10B-Instruct FT	57.65	55.15	54.25	53.60	55.95	71.90	58.17
NileChat-3B	67.55	64.75	64.65	64.45	66.00	65.15	65.00
NileChat-3B FT	69.20	67.75	67.65	66.90	67.45	69.05	67.76
Fanar-1-9B-Instruct	65.75	62.95	62.40	61.00	61.45	65.30	62.62
Fanar-1-9B-Instruct FT	**72.55**	69.85	**70.55**	69.70	**70.75**	**72.65**	**70.70**
Qwen2.5-3B	59.65	53.70	54.50	52.65	54.85	61.50	55.44
Qwen2.5-3B FT	63.75	62.80	62.80	62.45	62.60	69.55	64.04
Qwen2.5-7B	61.95	60.25	60.65	57.05	60.60	65.15	60.74
Qwen2.5-7B FT	67.50	65.85	65.95	63.25	66.00	71.50	66.51
ALLaM-7B-Instruct-preview	67.25	65.70	64.90	64.35	66.20	62.15	64.66
ALLaM-7B-Instruct-preview FT	71.95	**70.55**	69.85	**69.85**	70.40	67.70	69.67
Avg. Arabic-Centric	66.85	64.47	63.98	63.27	64.55	64.20	64.09
Avg. Arabic-Centric FT	71.23	69.38	69.35	68.82	69.53	69.80	69.38
Avg. Base All	61.37	58.55	58.53	57.03	59.04	64.29	59.49
Avg. FT All	67.10	65.33	65.18	64.29	65.53	70.39	66.14
GPT-4.1	77.42	79.08	78.29	80.24	79.33	78.57	79.10
GPT-5	79.59	79.10	78.88	77.70	79.31	77.17	78.43

Table 2: MCQ accuracy (%) across different language variants. Fine-tuned models (FT) are shaded in light blue, GPT models in gray, and averages for Arabic-centric models (NileChat, Fanar, ALLaM) are highlighted in light green. Bold values indicate the best-performing open model per dialect/language.

Model	MSA		Egyptian		Levantine		Magrebi		Gulf		English		Average	
	F1	RL	F1	RL	F1	RL	F1	RL	F1	RL	F1	RL	F1	RL
Falcon3-10B-Instruct	0.43	0.12	0.41	0.10	0.41	0.09	0.41	0.09	0.41	0.10	0.54	0.23	0.44	0.12
NileChat-3B	0.48	0.17	0.49	0.18	0.50	0.17	0.50	0.18	0.49	0.17	0.49	0.15	0.49	0.17
Fanar-1-9B-Instruct	0.52	0.20	0.50	0.17	0.50	0.16	0.50	0.17	0.51	0.17	**0.53**	0.18	0.51	0.18
Qwen2.5-3B	0.45	0.13	0.43	0.11	0.43	0.10	0.44	0.11	0.44	0.11	0.47	0.11	0.44	0.11
Qwen2.5-7B	**0.55**	0.24	**0.51**	0.20	**0.51**	0.19	**0.53**	0.21	**0.52**	0.20	**0.53**	0.20	**0.53**	0.21
ALLaM-7B-Instruct	0.49	0.20	0.47	0.16	0.47	0.15	0.46	0.15	0.48	0.17	0.52	0.22	0.48	0.17
Avg. Arabic-Centric	0.50	0.19	0.49	0.17	0.49	0.16	0.49	0.17	0.49	0.17	0.51	0.18	0.50	0.17
Avg. Base	0.49	0.18	0.47	0.16	0.47	0.14	0.47	0.15	0.47	0.15	0.51	0.18	0.48	0.16
GPT-4.1	0.55	0.27	0.53	0.24	0.53	0.21	0.54	0.24	0.54	0.24	0.56	0.25	0.54	0.24
GPT-5	0.57	0.28	0.54	0.24	0.54	0.22	0.55	0.24	0.55	0.25	0.54	0.22	0.55	0.24

Table 3: OEQ performance across different language variants. F1: BERTScore F1, RL: Rouge-L. Averages for Arabic-centric models (NileChat, Fanar, ALLaM) are highlighted in light green. GPT models are shaded in gray.

Lang	Base			FT			FT with COT		
	J	F1	RL	J	F1	RL	J	F1	RL
MSA	5.50	0.49	0.20	6.02	0.76	0.56	6.12	0.70	0.48
Eg	4.93	0.47	0.16	5.90	0.71	0.46	6.10	0.66	0.41
Lv	4.95	0.47	0.15	5.93	0.70	0.45	6.13	0.66	0.40
Mg	4.80	0.46	0.15	5.88	0.70	0.45	6.08	0.65	0.39
Gf	4.97	0.48	0.17	5.94	0.70	0.45	6.14	0.66	0.41
En	4.49	0.52	0.22	5.55	0.74	0.57	5.48	0.67	0.43
Avg.	4.94	0.48	0.17	5.87	0.72	0.49	6.01	0.67	0.42

Table 4: Performance on the OEQ across ALLaM-7B base, fine-tuned (FT), and fine-tuned with CoT models. J: LLM-as-a-judge.

than in the MCQ setting (Table 3). Similarly, English achieves higher F1 than the Arabic variants. Notably, for OEQ, the `Qwen2.5-7B` open model outperforms the other open models, including Arabic-centric ones.

Among all base models, `GPT-5` achieves the highest overall performance ($F1 = 0.55$), followed closely by `GPT-4.1` ($F1 = 0.54$). GPT-5 performs best on MSA, while GPT-4.1 shows strong results on both English and MSA.

Did CoT help for OEQ? In Table 4, we report the performance of OEQ with a comparison to the base model, fine-tuning without CoT (FT), and fine-tuning with CoT. Other than F1 and Rouge-L score, we also report LLM-as-a-judge scores. On token-overlap metrics, FT yields the strongest scores (F1/RL), whereas the CoT-tuned model achieves the highest average *LLM-as-a-judge* score. This difference indicates that CoT improves *semantic acceptability* but reduces *lexical overlap* with the references. A manual pass over low-F1 cases shows that the CoT model frequently returns briefer answers that judges deem correct, yet they share fewer n-grams with the (often longer) gold strings, decreasing F1 and RL. Overall, CoT helps on judged correctness but not on n-gram overlap.

This pattern aligns with prior findings that CoT is not uniformly beneficial. For instance, Zhu et al.

(2025) show that adding rationales can sometimes hurt performance, while Li et al. (2025) find that fine-tuning smaller models on lengthy, teacher-generated CoT traces performs no better, or worse, than training without CoT. Together with our results, these observations highlight the need to examine when CoT is advantageous, particularly regarding task type, rationale length, and model size.

6. Conclusions and Future Work

We presented a comprehensive pipeline for converting MCQ into OEQ and extended an existing MCQ dataset across multiple language varieties, including MSA, English, and several Arabic dialects. To our knowledge, ***ArabicCulturalQA*** is the ***first*** Arabic cultural **OEQ resource with parallel dialectal variants** alongside English, providing a foundation for culturally grounded evaluation beyond MSA. Though this dataset is based on PalmX, however, we have extensively extended it: QAs are *parallelly aligned* across all language variants. We have manually checked MCQ to OEQ mapping and post-edited the translated dialectal MCQs by native speakers to ensure naturalness and fidelity. We benchmarked the dataset using a set of open, closed, and fine-tuned models. Fine-tuned models consistently outperform their base counterparts yet still lag behind strong closed models. Performance is generally higher for MSA than for dialects. Arabic-centric models show advantages on Arabic variants for MCQ but smaller gains on OEQ, highlighting the added difficulty of generative, culturally grounded answering. Our initial CoT results improve judged correctness but yield mixed n-gram–based scores. Future work includes annotation and post-editing of the remaining dialectal OEQ test set, variety-aware normalisation and scoring, and extending to additional low-resource languages and modalities.

7. Ethics statement

We do not anticipate ethical concerns arising from this work. We build on publicly available datasets that permit research use, and we comply with their licenses and terms. For the manual annotations, contributors participated after being fully briefed on the task and its purpose. Initial annotators participated on a voluntary basis, while annotators involved in the expanded annotation and post-editing stages were compensated for their work. No personal or sensitive data were collected beyond what is contained in the source datasets.

8. Limitations

Our extensions to publicly available Arabic-dialect datasets rely on LLM-assisted translation and MCQ→OEQ conversion, which may introduce modeling biases (e.g., paraphrase drift, dialectal normalization) and occasional errors. Although we subsequently performed systematic annotation and dialectal post-editing to improve naturalness and quality, the experimental results reported in this paper were computed on the automatically generated versions of the datasets prior to human post-editing. As a result, the reported benchmarks may underestimate the quality and usability of the finalized released datasets.

References

Ahmed Abdelali, Hamdy Mubarak, Shammur Chowdhury, Maram Hasanain, Basel Mousi, Sabri Boughorbel, Samir Abdaljalil, Yassine El Kheir, Daniel Izham, Fahim Dalvi, Majd Hawasly, Nizi Nazar, Youssef Elshahawy, Ahmed Ali, Nadir Durrani, Natasa Milic-Frayling, and Firoj Alam. 2024. LAraBench: Benchmarking Arabic AI with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 487–520, St. Julian's, Malta. Association for Computational Linguistics.

Firoj Alam, Md Arid Hasan, Sahinur Rahman Laskar, Mucahid Kutlu, and Shammur Absar Chowdhury. 2025a. NativQA Framework: Enabling llms with native, local, and everyday knowledge. *arXiv preprint arXiv:2504.05995*.

Firoj Alam, Ali Ezzat Shahroor, Md. Arid Hasan, Zien Sheikh Ali, Hunzalah Hassan Bhatti, Mohamed Bayan Kmainasi, Shammur Absar Chowdhury, Basel Mousi, Fahim Dalvi, Nadir Durrani, and Natasa Milic-Frayling. 2025b. EverydayMMQA: A multilingual and multimodal framework for culturally grounded spoken visual qa. *arXiv preprint arXiv:2510.06371*.

Fakhraddin Alwajih, Abdellah El Mekki, Samar Mohamed Magdy, AbdelRahim A. Elmadany, Omer Nacar, El Moatez Billah Nagoudi, Reem Abdel-Salam, Hanin Atwany, Youssef Nafea, Abdulfattah Mohammed Yahya, Rahaf Alhamouri, Hamzah A. Alsayadi, Hiba Zayed, Sara Shatnawi, Serry Sibaee, Yasir Ech-chammakhy, Walid Al-Dhabyani, Marwa Mohamed Ali, Imen Jarraya, Ahmed Oumar El-Shangiti, Aisha Alraeesi, Mohammed Anwar AL-Ghrawi, Abdulrahman S. Al-Batati, Elgizouli Mohamed, Noha Taha Elgindi, Muhammed Saeed, Houdaifa Atou, Issam Ait Yahia, Abdelhak Bouayad, Mohammed Machrouh, Amal Makouar, Dania

Alkawi, Mukhtar Mohamed, Safaa Taher Abdelfadil, Amine Ziad Ounnoughene, Anfel Rouabhia, Rwaa Assi, Ahmed Sorkatti, Mohamedou Cheikh Tourad, Anis Koubaa, Ismail Berrada, Mustafa Jarrar, Shady Shehata, and Muhammad Abdul-Mageed. 2025a. Palm: A culturally inclusive and linguistically diverse dataset for Arabic LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32871–32894, Vienna, Austria. Association for Computational Linguistics.

Fakhraddin Alwajih, Abdellah El Mekki, Hamdy Mubarak, Majd Hawasly, Abubakr Mohamed, and Muhammad Abdul-Mageed. 2025b. PalmX 2025: The first shared task on benchmarking llms on arabic and islamic culture. In *Proceedings of the Third Arabic Natural Language Processing Conference (ArabicNLP 2025)*, Suzhou, China. Association for Computational Linguistics. Co-located with EMNLP 2025, November 5–9.

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, Marseille, France. European Language Resource Association.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and Wanli Ouyang. 2024. MT-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7421–7454, Bangkok, Thailand. Association for Computational Linguistics.

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham Abdullah Alyahya, Sultan AlRashed, Faisal Abdulrahman Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Saad Amin Hassan, Dr. Majed Alrubaian, Ali Alammari, Zaki Alawami, Abdulmohsen Al-Thubaity, Ahmed Abdelali, Jeril Kuriakose, Abdalghani Abujabal, Nora Al-Twairesh, Areeb Alowisheq, and Haidar Khan. 2025. ALLam: Large language models for arabic and english. In *The Thirteenth International Conference on Learning Representations*.

Hunzalah Hassan Bhatti, Youssef Ahmed, Md Arid Hasan, and Firoj Alam. 2025. Cultranai at palmx 2025: Data augmentation for cultural knowledge representation.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. Technical report, Microsoft Research.

Nikhil Chandak, Shashwat Goel, Ameya Prabhu, Moritz Hardt, and Jonas Geiping. 2025. Answer matching outperforms multiple choice for language model evaluation. *arXiv preprint arXiv:2507.02856*.

Md. Arid Hasan, Maram Hasanain, Fatema Ahmad, Sahinur Rahman Laskar, Sunaya Upadhyay, Vrunda N Sukhadia, Mucahid Kutlu, Shammur Absar Chowdhury, and Firoj Alam. 2025. NativQA: Multilingual culturally-aligned natural query for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14886–14909, Vienna, Austria. Association for Computational Linguistics.

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.

Mohamed Bayan Kmainasi, Rakif Khan, Ali Ezzat Shahroor, Boushra Bendou, Maram Hasanain, and Firoj Alam. 2025. Native vs non-native language prompting: A comparative analysis. In *Web Information Systems Engineering – WISE 2024*, pages 406–420, Singapore. Springer Nature Singapore.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Cheng Li, Damien Teney, Linyi Yang, Qingsong Wen, Xing Xie, and Jindong Wang. 2024a. CulturePark: Boosting cross-cultural understanding in large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 65183–65216.

Wangyue Li, Liangzhi Li, Tong Xiang, Xiao Liu, Wei Deng, and Noa Garcia. 2024b. Can multiple-choice questions really be useful in detecting the abilities of llms? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2819–2834.

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. 2025. Small models struggle to learn from strong reasoners. *arXiv preprint arXiv:2502.12143*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL Workshop*.

Quentin Malartic, Nilabhra Roy Chowdhury, Ruxandra Cojocaru, Mugariya Farooq, Giulia Campesan, Yasser Abdelaziz Dahou Djilali, Sanath Narayan, Ankit Singh, Maksim Velikanov, Basma El Amel Boussaha, et al. 2024. Falcon2-11b technical report. *arXiv preprint arXiv:2407.14885*.

Abdellah El Mekki, Houdaifa Atou, Omer Nacar, Shady Shehata, and Muhammad Abdul-Mageed. 2025. Nilechat: Towards linguistically diverse and culturally aware llms for local communities. *arXiv preprint arXiv:2505.18383*.

Francesco Maria Molfese, Luca Moroni, Luca Gioffré, Alessandro Scirè, Simone Conia, and Roberto Navigli. 2025. Right answer, wrong score: Uncovering the inconsistencies of llm evaluation in multiple-choice question answering. *arXiv preprint arXiv:2503.14996*.

Basel Mousi, Nadir Durrani, Fatema Ahmad, Md Arid Hasan, Maram Hasanain, Tameem Kabbani, Fahim Dalvi, Shammur Absar Chowdhury, and Firoj Alam. 2025. AraDiCE: Benchmarks for dialectal and cultural capabilities in LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4186–4218, Abu Dhabi, UAE. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Aidar Myrzakhan, Sondos Mahmoud Bsharat, and Zhiqiang Shen. 2024. Open-llm-leaderboard: From multi-choice to open-style questions for llms evaluation, benchmark, and arena. *arXiv preprint arXiv:2406.07545*.

Junho Myung, Nayeon Lee, Yi Zhou, Jiho Jin, Rifki Afina Putri, Dimosthenis Antypas, Hsuvas Borkakoty, Eunsu Kim, Carla Perez-Almendros, Abinew Ali Ayele, et al. 2024. BLEnD: A benchmark for llms on everyday knowledge in diverse cultures and languages. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada.

OpenAI. 2025. Gpt-5 technical overview. https://openai.com/research/gpt-5.

Siddhesh Pawar, Junyeong Park, Jiho Jin, Arnav Arora, Junho Myung, Srishti Yadav, Faiz Ghifari Haznitrama, Inhwa Song, Alice Oh, and Isabelle Augenstein. 2025. Survey of cultural awareness in language models: Text and beyond. *Computational Linguistics*, pages 1–96.

Haritz Puerto, Tilek Chubakov, Xiaodan Zhu, Harish Tayyar Madabushi, and Iryna Gurevych. 2025. Fine-tuning on diverse reasoning chains drives within-inference CoT refinement in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3789–3808, Vienna, Austria. Association for Computational Linguistics.

Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2695–2709, Singapore. Association for Computational Linguistics.

Narun Raman, Taylor Lundy, and Kevin Leyton-Brown. 2025. Reasoning models are test exploiters: Rethinking multiple-choice. *arXiv preprint arXiv:2507.15337*.

Abdelrahman Sadallah, Junior Cedric Tonga, Khalid Almubarak, Saeed Almheiri, Farah Atif, Chatrine Qwaider, Karima Kadaoui, Sara Shatnawi, Yaser Alesh, and Fajri Koto. 2025. Commonsense reasoning in Arab culture. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7695–7710, Vienna, Austria. Association for Computational Linguistics.

Fanar Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed Elmagarmid, Mohamed Eltabakh, Masoomali Fatehkia, Anastasios Fragkopoulos, Maram Hasanain, Majd Hawasly, Mus'ab Husaini, Soon-Gyo Jung, Ji Kim

Lucas, Walid Magdy, Safa Messaoud, Abubakr Mohamed, Tasnim Mohiuddin, Basel Mousi, Hamdy Mubarak, Ahmad Musleh, Zan Naeem, Mourad Ouzzani, Dorde Popovic, Amin Sadeghi, Husrev Taha Sencar, Mohammed Shinoy, Omar Sinan, Yifan Zhang, Ahmed Ali, Yassine El Kheir, Xiaosong Ma, and Chaoyi Ruan. 2025. Fanar: An arabic-centric multimodal generative ai platform.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Ping Yu, Jack Lanchantin, Tianlu Wang, Weizhe Yuan, Olga Golovneva, Ilia Kulikov, Sainbayar Sukhbaatar, Jason Weston, and Jing Xu. 2025. Cot-self-instruct: Building high-quality synthetic data for reasoning and non-reasoning tasks. *arXiv preprint arXiv:2507.23751*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. STaR: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with bert. In *Proceedings of ICLR 2020*.

Chiwei Zhu, Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, and Zhendong Mao. 2025. Rationales are not silver bullets: Measuring the impact of rationales on model performance and reliability. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5808–5835, Vienna, Austria. Association for Computational Linguistics.

9. Appendix

9.1. Annotation Guidelines

To ensure the quality and reliability of the generated dataset, we conducted structured human annotation and post-editing using an in-house annotation platform. This appendix provides an overview of the annotation setup and summarizes the instructions provided to annotators.

9.1.1. MCQ→OEQ Annotation

The entire MSA OEQ test set, generated by transforming MSA MCQs into open-ended questions, was reviewed by three independent annotators. Annotators were presented with both the original MCQ (including the question, choices, and correct answer) and the automatically generated OEQ pair consisting of a question and answer.

An example of the annotation interface used for this task is shown in Figure 3. Annotators evaluated the quality of both the transformed question and the generated answer.

Annotators were asked to assign scores on a five-point Likert scale for two aspects:

- **Open-Ended Question Quality (1–5):** evaluating clarity, naturalness in MSA, completeness, and whether the question is self-contained with appropriate scope.
- **Open-Ended Answer Quality (1–5):** evaluating correctness relative to the MCQ answer, completeness, conciseness, and fluency.

Scores were interpreted as follows:

- 1: Poor
- 2: Weak
- 3: Acceptable
- 4: Good
- 5: Excellent

When a score below 4 was assigned, annotators were required to select at least one revision reason explaining the issue. These included categories such as:

- unclear or ambiguous question
- grammatical errors
- not self-contained
- overly broad or overly narrow scope
- incorrect or incomplete answer
- speculative or unsupported content

Annotators could also provide optional comments for particularly difficult or ambiguous cases.

9.1.2. Dialectal Translation Post-Editing

For the dialectal datasets, we performed manual post-editing of the MCQ test sets after automatic translation. Each dialect subset was reviewed by a native speaker of the corresponding dialect who edited the translated question and answer options

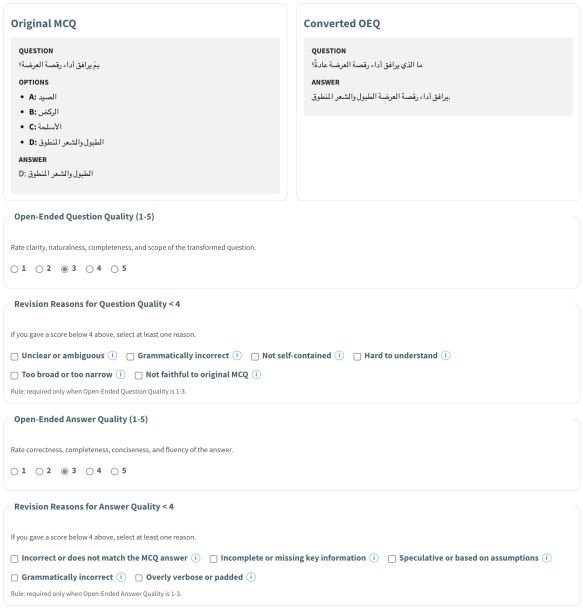

Figure 3: Annotation interface for evaluating MCQ→OEQ transformations. Annotators review the original MCQ alongside the generated open-ended question and answer.

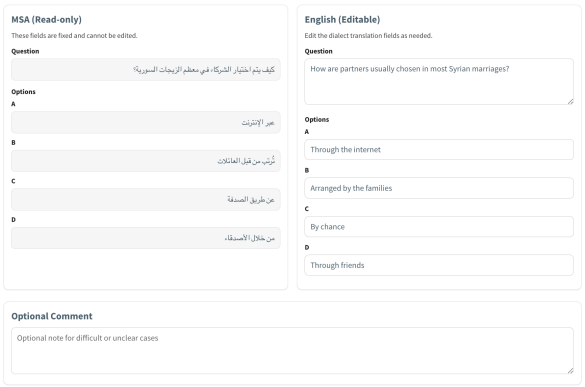

Figure 4: Interface used for dialectal translation post-editing. Annotators review the original MSA question and revise the translated English version.

when necessary to improve naturalness and linguistic authenticity.

The annotation interface for this task is illustrated in Figure 4. Annotators were shown:

- the original MSA content (read-only)
- the translated version (editable)

Their task was to revise only the translated fields to ensure:

- semantic fidelity to the MSA source
- fluent and natural wording in the target variety
- consistent terminology and phrasing across question and options

Annotators were instructed not to modify the MSA fields and to keep the meaning faithful to the source question. Optional comments could be provided for unclear or difficult cases.

Together, these annotation and post-editing steps ensure that both the open-ended question transformations and the dialectal translations maintain semantic fidelity, linguistic quality, and cultural authenticity.

9.2. Prompts

This section provides the prompts used throughout this work, including those for chain-of-thought (CoT) reasoning generation, dataset construction, and benchmarking.

9.2.1. Prompts for CoT Generation

Listings 1–3 include the Solving, Rationalizing, and Verification prompts used to generate CoT reasoning.

9.2.2. Prompts for Dataset Construction and Evaluation

Listings 4–5 cover dataset construction (MCQ-to-MSA and dialectal conversion). Listings 6–7 provide the zero-shot MCQ and OEQ templates. Listings 8–9 contain the LLM-as-a-Judge prompt and output schema.

```
[System Prompt]
You are a culturally knowledgeable assistant. Your task is to answer questions based on
cultural context, history, and common perspectives.
1. First, analyze the question to determine if it's factual or subjective (opinion-based).
2. Provide step-by-step reasoning that explains the answer.
   - For factual questions, use established cultural or historical facts.
   - For subjective questions, explain the common viewpoints or the reasons behind a
   popular opinion, acknowledging its subjective nature.
3. Keep the reasoning concise and relevant.
4. Finally, provide a brief, direct answer.

Return ONLY a strict JSON object:
{"cot": "<step-by-step reasoning>", "final": "<short answer>"}

[User Prompt]
Question:
{question}

Respond as JSON only:
{"cot": "...", "final": "..."}.
```

Listing 1: Solve Prompt

```
[System Prompt]
You are a cultural explainer. Given a question and a known answer, your task is to
construct a step-by-step explanation that coherently leads to that answer.
- The explanation must be grounded in relevant cultural knowledge, history, or common
perceptions.
- For subjective answers, the reasoning should justify why this is a plausible or common
perspective.
- Do not simply state the answer in the first step; build up to it through your
explanation.

Return ONLY a strict JSON object:
{"cot": "<step-by-step reasoning>", "final": "<the provided answer>"}

[User Prompt]
Question:
{question}

Known correct answer:
{gold}

Construct an explanation that leads to the known answer. Restate the given answer in the
'final' field.
Respond as JSON only.
```

Listing 2: Rationalization Prompt

```
[System Prompt]
You are a meticulous cultural context verifier. Your job is to assess a proposed
chain-of-thought (CoT) against a question and a gold answer.

Evaluate the CoT based on the following criteria:
1. Cultural Soundness: Is the reasoning culturally relevant and sound?
2. Factual Accuracy: Are factual claims correct?
3. Subjectivity Handling: If the question is subjective, does the CoT frame it as opinion
rather than objective fact?
4. Logical Support: Does the reasoning support the final answer?

Provide a score from 0.0 to 1.0. The verdict is "pass" if score >= 0.8, otherwise "fail".
Return ONLY JSON:
{"score": <float>, "verdict": "pass|fail", "issues": ["..."]}

[User Prompt]
Question: {question}
Gold answer: {gold}
Proposed chain-of-thought (cot): {cot}
Proposed final answer: {final}

Assess the proposed CoT and final answer.
```

Listing 3: Verification Prompt

```
You are given multiple-choice questions in JSON format. Your task is to convert them into
open-ended Q&A format when possible.

Language: Always write the output in Modern Standard Arabic (MSA).

Rules:
- Rephrase the question so it works without multiple-choice options.
- Rephrase the correct answer as a natural full sentence.
- Use only the correct option for the final answer.
- If the question cannot reasonably be converted, mark it as not_possible.

Output format:
{
 "id": <same as input>,
 "open_question": "<MSA question>",
 "open_answer": "<MSA answer>",
 "status": "ok" | "not_possible"
}
```

Listing 4: MCQ to MSA Open-Ended Conversion Prompt

```
You are a professional linguistic converter specializing in Arabic dialect adaptation.

Convert a Modern Standard Arabic question-answer pair into:
1. Gulf Arabic
2. Egyptian Arabic
3. Levantine Arabic
4. Maghrebi Arabic
5. English

Guidelines:
- Preserve meaning and cultural context
- Use natural dialect phrasing
- Maintain respectful tone
- Do not translate word-for-word

Return strictly JSON:

{
 "id": <same as input>,
 "msa_question": "...",
 "msa_answer": "...",
 "gulf_question": "...",
 "gulf_answer": "...",
 "egyptian_question": "...",
 "egyptian_answer": "...",
 "levantine_question": "...",
 "levantine_answer": "...",
 "maghrebi_question": "...",
 "maghrebi_answer": "...",
 "english_question": "...",
 "english_answer": "..."
}
```

Listing 5: MSA Open-Ended to Dialectal Conversion Prompt

```
You are an AI assistant specialized in answering multiple-choice questions about Arab
culture and society.

Instructions:
- Read the question carefully
- Consider all options
- Respond ONLY with the correct letter (A, B, C, or D)
- Do not include any explanation

Your response must be exactly one letter.
```

Listing 6: MCQ Zero-Shot Prompt

```
You are an expert Arabic culture scholar.

Answer open-ended questions directly and concisely.
Responses should reflect accurate cultural, social, or geographical knowledge.

Return only the answer with no explanation or extra text.
```

Listing 7: Open-Ended Zero-Shot Prompt

```
You are a strict, reference-grounded QA
evaluator. Grade a single predicted
answer to an open-ended question only using the
provided materials
(question, reference answer(s), and optional
evidence/context).
Do NOT use outside knowledge. Judge in the same
language as the
predicted answer (Arabic or English).

Return ONLY a valid JSON object that matches
the schema exactly-no
extra text.

Score each criterion as an integer from 1 to 10
(1 = poor, 10 = excellent):

- Clarity: Is the writing easy to read,
well-structured, and unambiguous?
- Informativeness: Does it cover the important
points the question asks for?
- Plausibility: Is it internally coherent and
reasonable given the question
  and references/context?
- Faithfulness: Are claims supported by the
reference/evidence?
  No hallucinations or contradictions.
- Helpfulness: Does it directly address the
user's need without fluff or evasion?
- Accuracy: Are facts correct relative to the
reference/evidence
  (numbers, entities, relations)?
- Depth and Creativity: Nuance or insightful
distinctions while remaining correct.
- Level of Detail: Appropriate specificity for
the question.

Also compute an overall score (integer 1-10),
typically the rounded mean
of the eight criteria.

Detect and list any unsupported claims, missing
key points, or contradictions.
Be concise in rationales.

{Output_Schema}
```

Listing 8: LLM-as-Judge Prompt for OEQ Evaluation – Instructions

```
Output_Schema:
{
  "scores": {
    "clarity": 1,
    "informativeness": 1,
    "plausibility": 1,
    "faithfulness": 1,
    "helpfulness": 1,
    "accuracy": 1,
    "depth_creativity": 1,
    "level_of_detail": 1,
    "overall": 1
  },
  "rationale": {
    "overall": "1-3 sentences justifying the
    overall score.",
    "faithfulness": "Key supported/unsupported
    points in 1-3 sentences.",
    "accuracy": "Any factual errors vs.
    reference/evidence in 1-3 sentences."
  },
  "findings": {
    "unsupported_claims": ["..."],
    "missing_key_points": ["..."],
    "contradictions": ["..."]
  }
}
```

Listing 9: LLM-as-Judge Prompt – Output Schema

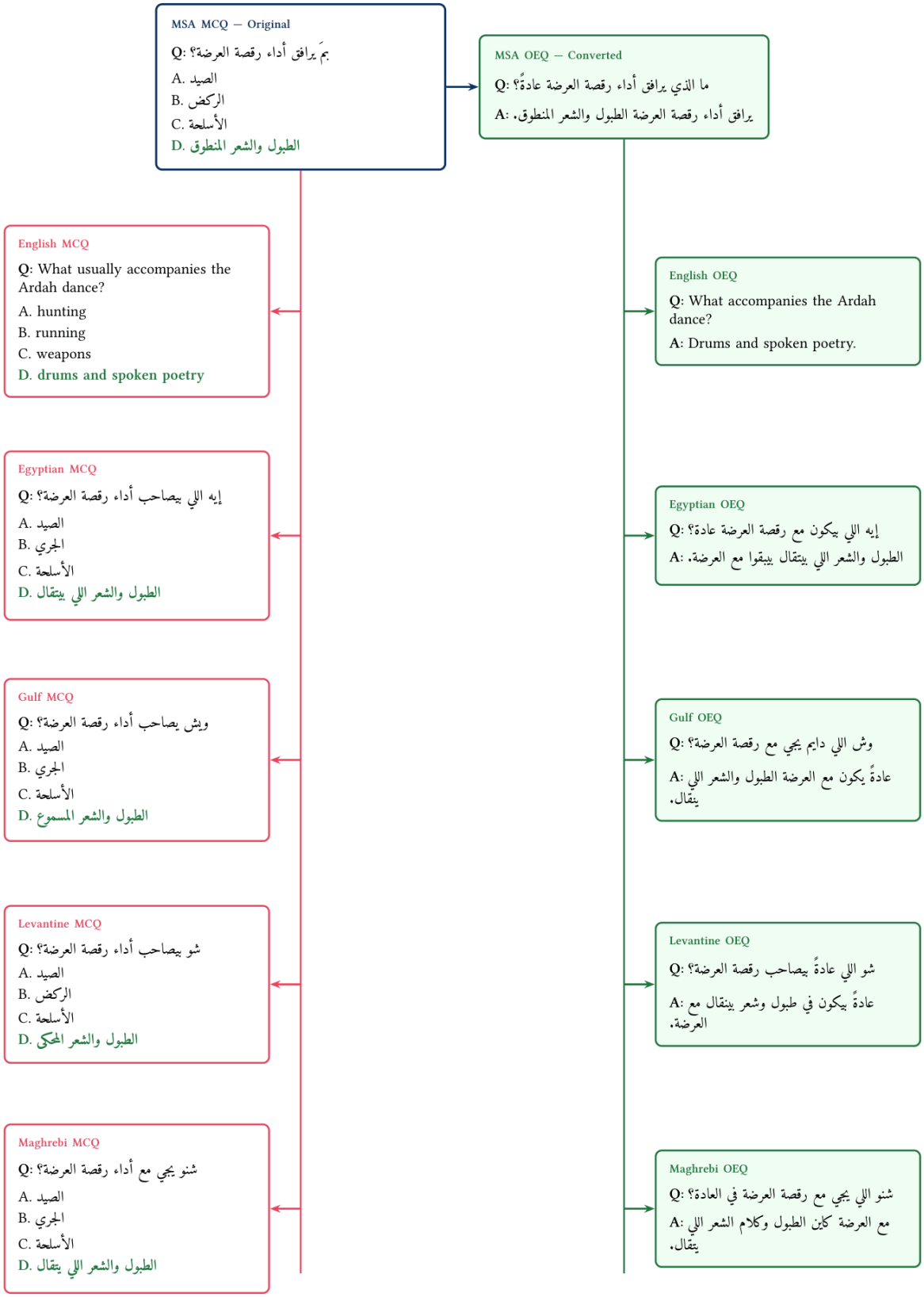

Figure 5: An example from the generated test set before post-editing and annotation. The MSA MCQ is first converted into the target dialects and simultaneously transformed into an MSA OEQ. The resulting MSA OEQ is then further adapted into its corresponding dialectal OEQ variants.