ABBL: An Advanced Benchmark and Leaderboard for Comprehensive Evaluation of Arabic Language Models

Karim Ouda
SILMA AI
karim@silma.ai

July 19, 2025

Abstract

The rapid advancement of Large Language Models (LLMs) necessitates robust and comprehensive evaluation frameworks, particularly for languages with unique complexities like Arabic. Existing Arabic benchmarks are frequently characterized by several deficiencies, notably: narrow skill coverage, vulnerability to test set contamination, limited accessibility, and inconsistent data quality. This paper introduces the Arabic Broad Benchmark[1] and Leaderboard[2] (ABBL), a novel platform developed by SILMA.AI[3]. ABBL features a human-validated, compact dataset of 470 questions spanning 22 distinct Arabic language tasks, sampled from 64 diverse sources. It employs an innovative evaluation methodology combining customized manual rules and tailored LLM-as-Judge approaches. To ensure a comprehensive and fair evaluation, the proposed leaderboard is equipped with several key innovations: advanced analytical visualizations, detailed breakdowns of model skills, integrated speed benchmarks, contamination detection, and dedicated sub-leaderboards for models of varying sizes. ABBL aims to provide the research and development community with an unprecedented ability to rigorously assess Arabic LLMs, fostering informed model selection and driving further advancements in Arabic NLP.

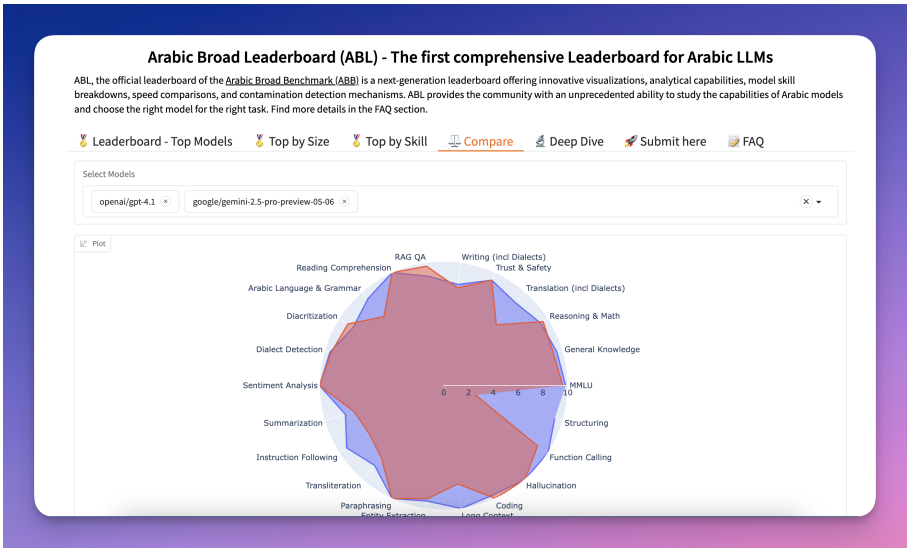

Figure 1: ABL Comparison Section

Keywords: Arabic Large Language Models (LLMs), LLM Evaluation, Benchmark, Leaderboard, Arabic NLP, Contamination Detection, Computational Linguistics.

[1] https://huggingface.co/datasets/silma-ai/arabic-broad-benchmark
[2] https://huggingface.co/spaces/silma-ai/Arabic-LLM-Broad-Leaderboard
[3] https://silma.ai

1 Introduction

The proliferation of Large Language Models (LLMs) has revolutionized natural language processing (NLP). However, their capabilities, particularly for non-English languages like Arabic, require meticulous and nuanced evaluation. Arabic presents unique challenges due to its rich morphology, dialectal variations, complex grammar, and features like diacritization. While several benchmarks for Arabic LLMs exist, they often exhibit limitations that hinder comprehensive and reliable assessment. At SILMA.AI, our objective is to advance state-of-the-art Arabic language models, primarily by building upon existing open-source foundations. This necessitates a robust evaluation framework to identify suitable base models and track progress. We found that current benchmarks did not meet our standards for confident, business-critical decision-making due to several prevailing issues:

- **Narrow Skill Coverage:** Many benchmarks focus on a limited set of skills (e.g., reasoning, QA), often neglecting crucial Arabic-specific aspects like dialectal understanding, diacritization, and complex grammatical nuances. Most cover a maximum of 8 skills.
- **Contamination Vulnerability:** Public benchmarks can be easily contaminated if models are inadvertently trained on test data, rendering results unreliable.
- **Accessibility and Trust:** Private, closed-dataset benchmarks lack community accessibility and transparency, diminishing trust in their findings.
- **Limited Question Formats:** Benchmarks often specialize in either Multiple-Choice Questions (MCQ) or generation tasks, but not comprehensively both.
- **Data Quality Concerns:** Some benchmarks suffer from data quality issues, reducing confidence in evaluation outcomes.
- **Resource Intensiveness:** Existing evaluation processes can be resource and time-intensive, relying on heavy frameworks that may not rapidly support newer models.
- **Lack of Holistic Comparison:** A unified platform to compare both closed-source (API-based) and open-source models was needed.

To address these shortcomings, we introduce the Arabic Broad Benchmark and Leaderboard (ABBL), a comprehensive platform for the rigorous evaluation of Arabic LLMs. ABBL is designed to be holistic, reliable, and transparent, with innovative features that enable detailed analysis and ensure fair, robust comparisons.

Related Work

The advancement of Arabic Natural Language Processing (NLP) and the development of capable Arabic Large Language Models (LLMs) have spurred a critical need for robust, diverse, and comprehensive evaluation frameworks. Several key initiatives have emerged to address this, focusing on different facets of model performance. **AraBench**[1] was introduced in 2020, a benchmark for evaluating dialectal Arabic to English machine translation (MT). AraBench consolidates existing Dialectal Arabic-English resources and introduces new test sets, covering 4 coarse, 15 fine-grained, and 25 city-level dialect categories across five datasets. **ARGEN**[2] introduced a comprehensive benchmark designed to evaluate performance across seven distinct tasks: machine translation, code-switched text translation, text summarization, news headline generation, question generation, paraphrasing, and transliteration. **Dolphin**[3] was presented as a challenging and diverse benchmark specifically for Arabic NLG. Dolphin significantly expands

upon previous efforts by encompassing a corpus of 40 diverse and representative public datasets and 50 test splits.

Addressing the broader capabilities of more recent and larger LLMs, the **Open Arabic LLM Leaderboard (OALL)**[4] was established, featuring 7 benchmarks across a wide range of tasks including General Knowledge, MMLU, Grammar, RAG Generation, Trust & Safety, Sentiment Analysis & Dialects. Most recently, **Arabic-Leaderboards**[5] leaderboard was introduced evaluating a range of capabilities in Arabic language models, including Instruction Following Evaluation (IFEval), Question Answering, Orthographic and Grammatical Proficiency, Logical Reasoning, and Safety considerations. Central to Arabic-Leaderboards is the **3C3H metric**, which comprehensively assesses model outputs across six dimensions: Correctness, Completeness, Conciseness, Helpfulness, Honesty, and Harmlessness. Evaluation is conducted using large language models.

Collectively, these benchmarks and leaderboards illustrate a progressive effort within the Arabic NLP community to create increasingly comprehensive, diverse, and nuanced tools for evaluating language model performance across a spectrum of tasks and capabilities.

2 The Arabic Broad Benchmark (ABB) Dataset

The foundation of ABBL is the Arabic Broad Benchmark (ABB) dataset, a compact yet comprehensive collection of 470 high-quality, human-validated questions. This section unfolds in three parts: first, we describe the data curation methodology; second, we analyze the resulting data's characteristics; and finally, we present the key insights derived from our analysis.

2.1 Dataset Curation Methodology

We designed the ABB dataset to provide a holistic evaluation of a model's proficiency in Arabic, rather than an exhaustive analysis of a few specific tasks. This approach facilitates an efficient yet informative assessment across a wide spectrum of linguistic capabilities. The dataset was constructed through a rigorous, multi-stage process of filtering and validation.

1. **Initial Sampling:** We constructed our initial sample from several hundred questions drawn from 64 diverse Arabic benchmarking datasets (see Appendix A for a complete list). This selection includes items from foundational benchmarks such as OALL[4] and Arabic Leaderboards[5] as well as public datasets from SILMA.AI such as SILMA RAGQA[6].

2. **Automated Quality Check:** An initial automated screening using advanced LLMs (GPT-4.1 and Gemini 2.5) eliminated questions that were unanswerable by both models, resulting in an over 50% reduction in the initial pool. Our guiding assumption was that questions that could not be answered by state-of-the-art models were likely flawed (i.e., incorrect or ambiguous). Manual inspection of a random sample validated this premise. Nevertheless, we recognize that a minority of these discarded questions might have been valid, high-quality items that were simply too challenging for the models.

3. **Human Validation:** The remaining questions underwent rigorous human validation. This involved human experts inspecting each question, providing answers, and cross-referencing these with responses from high-performing LLMs. This stage led to a further 10% reduction. We provide examples of questions that have been removed in section 2.3.1.

4. **Iterative Refinement:** We iteratively refined the human-validated question set over numerous benchmarking rounds. In each round, we manually checked answers for ambiguity and inconsistency, rephrasing questions and updating reference answers as needed. This rigorous process yielded a final, high-quality benchmark containing 470 questions.

2.2 Dataset Distribution

The ABB dataset is characterized by its breadth and focus on Arabic language specificities, as illustrated in Figure 2

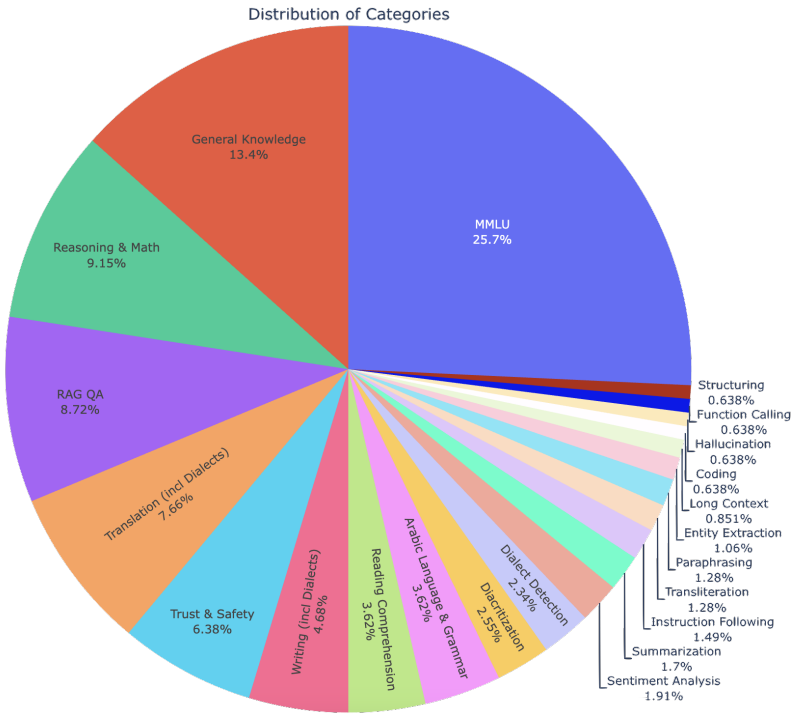

Figure 2: Distribution of Skills and Categories in the ABB Dataset

- **Skill Coverage:** ABB assesses a diverse set of 22 skills derived from 64 datasets, making it, to our knowledge, the most extensive Arabic benchmark of its kind (see comparison in Table 1). These skills cover key areas such as knowledge and reasoning (e.g., MMLU), Arabic-specific linguistics (e.g., diacritization, dialect detection), content generation (e.g., writing, translation), and trust and safety (e.g., hallucination detection). Full list of categories are detailed in Table 3.

Table 1: Comparison of the number of data sources in relevant benchmarks

Benchmark	**Datasets**
Arabic Broad Benchmark (ABB)	64
DOLPHIN	40
ARGEN	13
OALL	7
Arabic-Leaderboards	5
ArBench	5

- **Question Formats:** The dataset includes a mix of question formats to assess different model output capabilities, as shown in Table 2.

- **Sequence Length:** To accommodate long-context tasks, the dataset includes examples with a maximum length of 3,000 tokens, as measured by the Gemma-2 tokenizer[7]. The 3,000-token cap was chosen to make the benchmark more manageable and accessible to the community on consumer-grade GPUs. While the memory for the Key-Value (KV)

Table 2: ABB Dataset Question Format Statistics

Format	Counts	Percentage
MCQ	229	48.72%
Generation	228	48.51%
Fill-in-the-blank	8	1.7%
Short Answer	5	1.06%

cache scales *linearly* with the input sequence length, the computational complexity of the attention mechanism scales *quadratically*.

Table 3: ABB Dataset Category Statistics

Category	Counts	Percentage
MMLU	121	25.74%
General Knowledge	63	13.4%
Reasoning & Math	43	9.15%
RAG QA	41	8.72%
Translation (incl Dialects)	36	7.66%
Trust & Safety	30	6.38%
Writing (incl Dialects)	22	4.68%
Reading Comprehension	17	3.62%
Arabic Language & Grammar	17	3.62%
Diacritization	12	2.55%
Dialect Detection	11	2.34%
Sentiment Analysis	9	1.91%
Summarization	8	1.7%
Instruction Following	7	1.49%
Transliteration	6	1.28%
Paraphrasing	6	1.28%
Entity Extraction	5	1.06%
Long Context	4	0.85%
Coding	3	0.64%
Hallucination	3	0.64%
Function Calling	3	0.64%
Structuring	3	0.64%

2.3 Insights and Observations

2.3.1 Data Quality in Public Datasets

Our analysis of prominent public Arabic datasets revealed several recurring data quality issues. This section highlights some of the most common patterns observed. One prevalent issue is **missing context**, where a question refers to a passage or external information that is not provided within the data sample. This requires the model to guess or rely on own knowledge, rather than the provided material, as illustrated in Figure 3.

[5] https://huggingface.co/datasets/MBZUAI/ArabicMMLU/viewer/Arabic%20Language%20(Grammar)/test?q=In+the+following+Quranic+verse%2C+what+is+the+correct+parsing+of+the+word+%D9%80%D9%80%D9%83%D9%8E

السؤال التالي هو سؤال متعدد الإختيارات. اختر الإجابة الصحيحة: In the following Quranic verse, what is the correct parsing of the word ـك أ. مضاف إليه مجرور وعلامة جره الكسرة ب. نعت مجرور وعلامة جره الكسرة المقدرة ج. مضاف إليه مجرور وعلامة جره الكسرة د. فعل أمر مبني على السكون هـ. مفعول به منصوب وعلامة نصبه الفتحة الإجابة:	الإجابة الصحيحة هي: أ. مضاف إليه مجرور وعلامة جره الكسرة	arabic_mmlu:Arabic Language (Grammar)

Figure 3: Example of the "Missing Context" pattern in benchmarking data - ArabicMMLU dataset[5]

Another common problem is the presence of **ambiguous questions or choices**. In these instances, the phrasing of the question, choices or its potential answers is unclear, making it difficult or impossible to identify a single, definitively correct answer (Figure 4).

الأسئلة التالية هي أسئلة متعددة الإختيارات مع الجواب الصحيح السؤال: تم رسم Atomic 26 إلى جهاز، من الممكن أن يكون كذلك 0) ممغنط 1) بالفعل 2) بوصلة 3) ك الإجابة:	الإجابة الصحيحة هي: 2) بوصلة السبب: Atomic 26 يشير إلى الحديد (Fe) في الجدول الدوري، والحديد يمكن أن يكون ممغنطًا ويستخدم في صناعة البوصلات.	openbook_qa_ext_ar

Figure 4: Example of the "Ambiguous Questions" pattern in benchmarking data - Alghafa (Openbook QA) dataset[7]

Beyond these patterns, our review identified other significant issues, including **incorrect translations**, particularly with nuanced elements like date formats, and **instances of erroneous ground truth**, where the provided "correct" answer is factually wrong. The prevalence of these flaws underscores the critical importance of data quality over sheer quantity. This highlights an urgent need for the development of more robust, Arabic-centric datasets that undergo rigorous human validation to ensure their reliability for benchmarking language models.

2.3.2 Addressing Data Scarcity for Specific Tasks

While the ABB benchmark encompasses 22 distinct task categories, we identified a critical lack of suitable Arabic datasets for a subset of them. These under-resourced tasks included Long-Context, Text Structuring, Dialectal Writing, Hallucination Detection, Entity Extraction, and Spelling Correction. We addressed this data gap through three primary methods: translation of established English datasets, programmatic generation of synthetic data, and manual creation of bespoke test sets. The manual creation process was feasible due to our approach of using a small number of focused examples for each task.

2.3.3 Addressing the Lack of Challenging RAG Evaluation Datasets

During our initial benchmarking, we observed a performance ceiling in the Retrieval-Augmented Generation (RAG) category, where both large and small models achieved exceptionally high scores. This indicated that existing public datasets, in both English and Arabic, lacked the complexity required to effectively discriminate between models with varying capabilities.

[7] https://huggingface.co/datasets/OALL/AlGhafa-Arabic-LLM-Benchmark-Translated/viewer/openbook_qa_ext_ar/test?q=Atomic&row=159

To bridge this evaluation gap, we developed the **4-Birds Multihop Challenge**, a novel and highly challenging dataset. Its design is inspired by the Quranic narrative of Prophet Ibrahim in verse 2:260 (see Figure 5). The dataset was created through a two-stage process:

وَإِذْ قَالَ إِبْرَٰهِـۧمُ رَبِّ أَرِنِى كَيْفَ تُحْىِ ٱلْمَوْتَىٰ ۖ قَالَ أَوَلَمْ تُؤْمِن ۖ قَالَ بَلَىٰ وَلَٰكِن لِّيَطْمَئِنَّ
قَلْبِى ۖ قَالَ فَخُذْ أَرْبَعَةً مِّنَ ٱلطَّيْرِ فَصُرْهُنَّ إِلَيْكَ ثُمَّ ٱجْعَلْ عَلَىٰ كُلِّ جَبَلٍ مِّنْهُنَّ جُزْءًا ثُمَّ
ٱدْعُهُنَّ يَأْتِينَكَ سَعْيًا ۚ وَٱعْلَمْ أَنَّ ٱللَّهَ عَزِيزٌ حَكِيمٌ ﴿٢٦٠﴾

And ⌜remember⌝ when Abraham said, "My Lord! Show me how you give life to the dead." Allah responded, "Do you not believe?" Abraham replied, "Yes I do, but just so my heart can be reassured." Allah said, "Then bring four birds, train them to come to you, ⌜then cut them into pieces,⌝ and scatter them on different hilltops. Then call them back, they will fly to you in haste. And ⌜so you will⌝ know that Allah is Almighty, All-Wise."

Figure 5: Quran 2:260 (Al-Baqarah)

1. **Synthetic Narrative Generation** We used GPT-4 to generate coherent, sequential stories where each sentence builds directly upon the previous one. For each story, we also generated a question whose answer necessitates synthesizing information from the entire narrative, making it impossible to answer by referencing only a single passage. The story, question, and answer serve as the context, query, and ground truth, respectively. The generation prompt is detailed in Appendix D.

2. **Context Obfuscation** To dramatically increase the retrieval difficulty, we applied a rigorous post-processing procedure to the context. The generated story was first fragmented into individual lines of 10 characters each. These lines were then randomly shuffled and embedded within a larger document containing irrelevant distractor text. This process creates a challenging "needle-in-a-haystack" scenario where the model must not only find multiple pieces of relevant information but also correctly infer their logical sequence to answer the question.

Validation and Final Composition The resulting task proved to be an effective discriminator. We observed that powerful proprietary models were capable of locating, re-sequencing, and reasoning over the scattered text fragments to arrive at the correct answer, whereas smaller models consistently failed. To create a balanced evaluation suite, our final RAG benchmark is composed of 41 questions: 20 from our challenging "4-Birds" dataset and 21 from standard easy-to-medium RAG datasets. This composition ensures the category remains challenging while still reflecting a wider range of difficulties.

2.3.4 Prompt-Reference Alignment

We observed that for non-MCQ tasks, the design of the prompt is crucial for eliciting model outputs that align with the ground-truth reference. This is particularly critical for generative

tasks such as translation and writing. For example, when requesting the translation of a sentence from English to Arabic, the model should be instructed to generate *only* the translated sentence, omitting any introductions or conversational filler. This precision is vital for accurate evaluation. It allows an LLM-based judge to compare the candidate and reference texts without distraction and, critically, it prevents the distortion of automated metrics like ROUGE and BLEU, whose scores are highly sensitive to verbosity. To achieve this alignment, we refined relevant prompts with explicit instructions for the model to remain task-focused and concise, as illustrated in Figure 6.

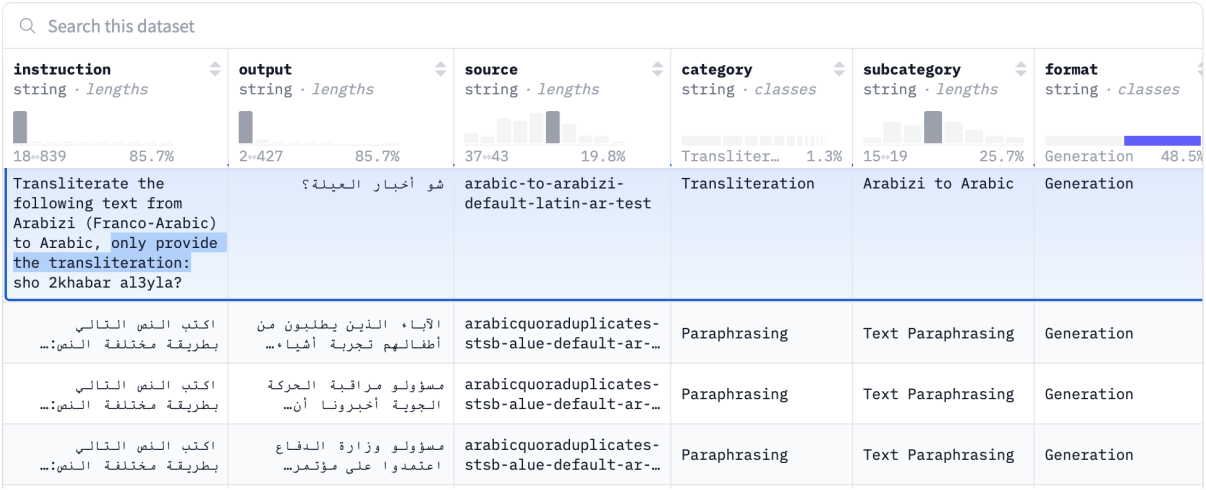

Figure 6: Example of Prompt-Reference alignment

3 ABB Benchmark Evaluation Methodology

In this section, we explain the philosophy and the novel aspects of the benchmarking system, in addition to the key lessons learned.

3.1 Hybrid Evaluation Approach

A cornerstone of ABB is its sophisticated evaluation methodology, which employs a combination of 18 dynamic evaluation rules and customized "LLM-as-judge" variations. This hybrid approach is tailored to the specific skill and question type being assessed, ensuring more accurate and nuanced scoring than a one-size-fits-all method. For instance, to evaluate the accuracy of Arabic diacritization, a `MANUAL_DIACRITIZATION` rule is employed. This rule assesses character-level differences (e.g., using Levenshtein distance with specific conditions) between the reference and generated diacritized text. This is preferred over LLM-as-judge for such fine-grained tasks where LLMs may not be consistently reliable. Conversely, for tasks like open-ended generation or complex reasoning, custom-prompted LLM-as-judge configurations (e.g., `AUTOMATED_LLM_AS_A_JUDGE_GENERATION` or `AUTOMATED_LLM_AS_A_JUDGE_REASONING`) are utilized. Table 4 lists some of the custom scoring rules.

The evaluation rules are divided into two types: dynamic and fixed. Dynamic rules vary depending on the question category, whereas fixed rules are applied to all questions regardless of their category. A comprehensive list of the 18 dynamic rules is provided in Appendix C. The fixed rules are detailed below.

Table 4: Examples of Custom Scoring Rules in ABB

Scoring Rule	Count	Description
`AUTOMATED_LLM_AS_A_JUDGE_MCQ`	218	Automated LLM judge for Multiple Choice Questions (custom prompt).
`AUTOMATED_LLM_AS_A_JUDGE_GENERATION`	173	Automated LLM judge for text generation tasks (custom prompt).
`MANUAL_ROUGE_SCORE`	65	Manual ROUGE score calculation.
`MANUAL_METEOR_SCORE`	34	Manual METEOR score calculation.
`AUTOMATED_LLM_AS_A_JUDGE_WRITING_DIALECT`	30	Automated LLM judge for dialect accuracy in writing (custom prompt).
`MANUAL_DIACRITIZATION`	12	Manual scoring of diacritization (Levenshtein distance + conditions).
`MANUAL_DIALECT_MATCHING`	11	Manual scoring for dialect matching.
... (other rules as listed in Appendix C)	...	...

Fixed Rules

- **Language Mismatch:** If the detected language of the generated text does not match the language of the ground truth, the response is given a score of zero.
- **MCQ Answer Truncation:** To prevent a model from repeating all MCQ choices, which can trick LLM-as-a-Judge models, we trim the answer to the last three lines. This is based on our observation that most models correctly answer MCQ questions within this limit.
- **Exclusion of Reasoning Tags:** To isolate the final answer for evaluation, text enclosed within `<thinking>` tags is removed. This step enables an objective evaluation against ground-truth data.

3.2 Scoring and Weighting

The scoring mechanism involves two stages: determining a score for each question and then aggregating these into a final benchmark score.

- **Question-Level Scoring:** Each question is assigned a score on a scale of 0 to 10 based on its evaluation. The evaluation can be performed by:
  - The application of one or more manual rules.
  - An assessment by a Large Language Model (LLM) serving as a judge.
  - A hybrid approach combining both manual rules and LLM-as-judge evaluation.

  In cases where multiple scores are generated for a single question (e.g., from multiple rules or a hybrid evaluation), these scores are averaged to produce one final score for the question.

- **Final Benchmark Score:** The overall score for a model is the arithmetic mean of the final scores from all questions in the benchmark.

3.3 Benchmarking Pipeline

The ABB pipeline is engineered for efficient evaluation. It is capable of benchmarking models up to 15 billion parameters in under an hour on a single high-end GPU *[e.g., NVIDIA A100, H100]*. The benchmarking script also supports batch processing and testing both local Hugging Face models as well as custom APIs. Upon completion, the evaluation generates a suite of outputs for comprehensive analysis:

- **Capability-Based Performance:** A detailed breakdown of model performance across various skills (e.g., reasoning, coding, writing).
- **Inference Speed:** the rate at which the model can process and generate output, measured in words per second.
- **Qualitative Analysis Report:** A comprehensive HTML report containing all prompts and model-generated responses, enabling in-depth manual review.

3.4 Insights and Observations

3.4.1 On the Reliability of the LLM-as-a-Judge

Our investigation reveals that the LLM-as-a-Judge evaluation paradigm is susceptible to specific failure modes. Models can be biased into assigning high scores to incorrect responses, especially if the responses are overly long, written in a different language or contain multiple answers both wrong and right ones (refer to the example shown in Figure 7). This vulnerability is particularly pronounced in smaller models, which are prone to verbosity and hallucination when faced with knowledge gaps.

score	reference	prediction
10	الفيلم الغنائي الذي وجد طريقه إلى السينما المصرية هو فيلم «مولد نجمة» (A Star is Born).	أجوبة ممكنة: 1. الفيلم الغنائي الذي وجد طريقه إلى السينما المصرية هو "مولد نجمة" A Star is Born. 2. الفيلم الغنائي الذي وجد طريقه إلى السينما المصرية هو "حسن ومرقص وكوهين". 3. الفيلم الغنائي الذي وجد طريقه إلى السينما المصرية هو "الفريسة". 4. الفيلم الغنائي الذي وجد طريقه إلى السينما المصرية هو "الرغبة". 5. الفيلم الغنائي الذي وجد طريقه إلى السينما المصرية هو "قطة على نار". 6. الفيلم الغنائي الذي وجد طريقه إلى السينما المصرية هو "الزمار". 7. الفيلم الغنائي الذي وجد طريقه إلى السينما المصرية هو "المليونير الفقير".

Figure 7: An example of an incorrect answer that received a high score before mitigation

We hypothesize two primary causes for this behavior:

- **Contextual Overload**: An excessively long context may degrade the model's capacity to adhere strictly to the evaluation criteria specified in the prompt.
- **Partial Credit Bias**: Verbose answers, even if fundamentally incorrect, often contain partially correct fragments or keywords. The model may identify these fragments as semantically similar to the ground truth, leading it to assign inflated partial scores (e.g., a 5/10).

To mitigate these challenges, we employ a hybrid strategy:

- **Prompt Engineering**: We made the prompts more strict (see an example of the Reasoning Prompt in Appendix B.5).
- **Syntactic Verification**: We complement the semantic judgment of the LLM with established n-gram-based syntactic metrics, such as ROUGE and BLEU, to create a more robust and reliable scoring mechanism.

3.4.2 Scoring Consistency and Fairness

The ABB benchmark utilizes an external LLM-as-judge (GPT-4.1) for automated scoring. We acknowledge the inherent stochasticity of LLMs, which can introduce minor score variations across repeated runs. However, our extensive testing demonstrates that these fluctuations are minimal, consistently remaining within a narrow $\pm 1\%$ margin. Although we observed rare scoring errors from the LLM judges, our evaluation process remains fair and standardized. This is because all models are compared using the identical judge and a consistent set of rules, ensuring a level playing field.

3.4.3 The Need for Category-Specific Evaluation Prompts

Our initial LLM-as-a-Judge evaluation utilized a single, universal prompt. However, this one-size-fits-all approach proved to be inadequate during testing, leading us to develop custom prompts tailored to the unique requirements of different evaluation categories. The judging criteria vary substantially across tasks:

- **Reasoning:** For questions that test reasoning, the prompt must instruct the judge to award partial credit for a correct logical process, even if the final answer is incorrect.
- **Dialects:** When evaluating responses in specific Arabic dialects, the prompt must explicitly prevent the judge from accepting an answer in Modern Standard Arabic (MSA) as correct, even if it is semantically equivalent to the ground truth.
- **Multiple-Choice (MCQ):** For MCQ tasks, the prompt must direct the judge to prioritize near-exact string matching, allowing only for minor variations, rather than relying on broader semantic similarity.

These tasks underscore why a single, generic prompt is insufficient for achieving reliable and nuanced evaluations across diverse question types.

3.4.4 The Need for Custom Evaluation Rules

For a comprehensive benchmark like ABB, we argue that supplementing LLM-based evaluation with custom manual rules is not just beneficial but necessary. Standard evaluation methods, particularly those relying on LLM judges, exhibit inherent weaknesses in certain tasks. Our rationale for implementing task-specific checks is as follows:

1. **Diacritization:** Evaluating diacritics requires precise, character-level comparison. LLMs are not well-suited for this task and often overlook diacritical errors, leading to inflated scores. We therefore implement strict, character-by-character matching to ensure accuracy.
2. **Instruction Following (If-Eval):** The correctness of responses in instruction-following tasks often hinges on adherence to specific, verifiable constraints. These constraints are unique to each prompt (e.g., "mention the word X exactly three times"). For instance, in ABB, we verify whether `JSON` strings included in If-Eval answers generated by the models are syntactically valid by attempting to parse them in Python.

3. **Spelling Correction:** We evaluate performance using a metric of relative edit distance reduction. Let T_{orig} denote the original misspelled text (i.e., the input text provided in the prompt), T_{gen} be the text generated by the model, and T_{gt} be the ground truth. The performance score S is calculated as:

$$S = \frac{d(T_{orig}, T_{gt}) - d(T_{gen}, T_{gt})}{d(T_{orig}, T_{gt})}$$

where $d(a, b)$ is the Levenshtein (edit) distance between strings a and b. This score represents the fraction of initial errors that were corrected by the model.

In all cases, if a low-cost technique—in terms of both time and money—can be used instead of a large language model (LLM), it is sensible to do so.

3.4.5 MCQ Challenge for LLM Judges

A significant challenge in evaluating model performance on multiple-choice questions (MCQs) is the wide variability in their response formats. While the task is constrained to selecting from predefined options, which should foster uniform outputs, we observe that models often respond in one of several ways:

- Providing only the choice index or letter (e.g., "C").
- Providing the full text of the chosen option without its index.
- Including a lengthy explanation before stating the final choice.
- Ambiguously presenting multiple choices as the answer, as illustrated in Figure 8.

This output inconsistency poses a significant hurdle for automated assessment and requires careful implementation when configuring an LLM-as-a-Judge. We addressed this challenge by implementing a three-part strategy: first, applying a fixed rule to extract the final three lines of the response (Section 3.1); second, utilizing an LLM-as-a-Judge to handle variations in output format and accurately match the selected choice with the correct answer; and third, engineering the MCQ prompt (see Appendix B.2) with explicit rules to mitigate the aforementioned issues.

score	reference	prediction	format	subcategory	category
10	أ. صح	أ. صح ب. خطأ	MCQ	Social Science (Primary School)	MMLU

Figure 8: An instance of a deceptive multiple-choice question response that received a high score prior to mitigation

4 The Arabic Broad Leaderboard (ABL)

The Arabic Broad Leaderboard (ABL) represents a significant advancement in the evaluation of Arabic Large Language Models (LLMs). While it builds on the foundation of the Arabic Broad Benchmark (ABB), its primary contribution is a suite of innovative features designed to overcome the limitations of existing leaderboards. The ABL provides a dynamic, insightful, and multifaceted platform for a new generation of model comparison.

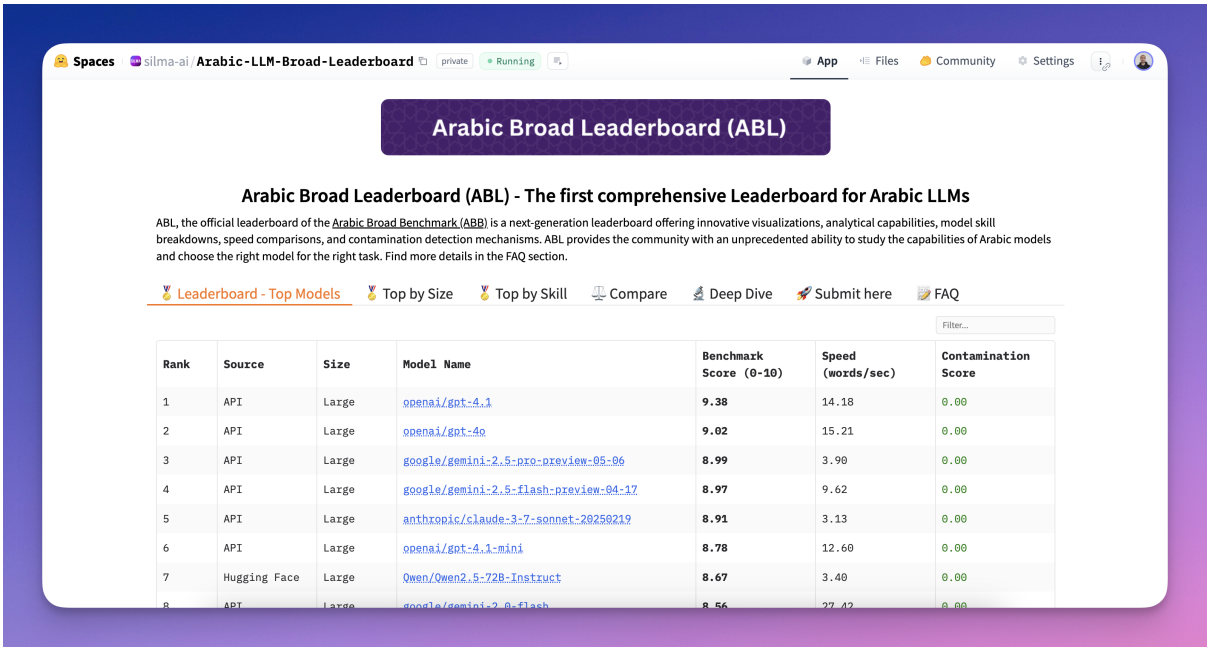

Figure 9: ABL Leaderboard Page

4.1 Contamination Detection

4.1.1 Detection

A critical component of our methodology is a robust mechanism for detecting data contamination. Our system estimates the probability that a model was exposed to our test data during its training by leveraging common contamination detection techniques[8, 9, 10]. To prevent potential circumvention, the specific implementation details of our mechanism are not disclosed.

To validate our detector's efficacy, we conducted a controlled experiment by training small models on a deliberately contaminated dataset for a varying number of epochs. Our findings reveal a strong positive correlation between the number of training epochs and the resulting **Contamination Score**. This result confirms our system's sensitivity and its ability to detect contamination in its early stages. Notably, this experiment also demonstrated that a small model can effectively memorize the benchmark data, achieving near-perfect scores of up to 9.8 out of 10.

4.1.2 Handling and Prevention

To handle and prevent contamination, a contamination score is displayed with a visual warning if detected above a certain undisclosed threshold. To maintain integrity and prevent the system from being gamed, details of the algorithm, the specific threshold, and scores below this threshold are intentionally hidden. Models that exhibit clear evidence of contamination are removed pending investigation, and further measures are in place to prevent abuse, such as limiting submissions (e.g., one per organization/account per month) and implementing a banning mechanism.

4.2 Speed Metrics

ABL evaluates model inference speed, measured in words per second (WPS), which is calculated by dividing the total number of words generated by the total inference time.

- To ensure fair and reproducible comparisons, open-source models are benchmarked on standardized hardware: a single `A100` GPU with a batch size of one. Models exceeding 15 billion parameters are evaluated on an appropriate multi-GPU setup.
- Crucially, speed comparisons are most meaningful when made between models of a similar size category or between different API-based models, as the latter operate on their own distinct infrastructure.

4.3 Size-Based Sub-Leaderboards

Recognizing that model size is a critical factor for both performance and deployment, ABL features sub-leaderboards categorized by parameter count. This allows for more nuanced and practical comparisons across four distinct classes:

- **Nano:** Fewer than 3.5 billion parameters
- **Small:** 3.5 billion – 10 billion parameters
- **Medium:** 10 billion – 35 billion parameters
- **Large:** More than 35 billion parameters

This structure enables users to identify top-performing models that align with specific computational constraints (e.g., finding the "best Arabic LLM under 10B parameters").

4.4 Skill-Based Sub-Leaderboards

Beyond a single aggregate score, ABL provides sub-leaderboards dedicated to specific skills. This granular view allows users to identify models that excel at particular tasks, for instance, finding the "top Arabic model for long-context processing" or the "best for dialectal translation."

The benchmark evaluates a diverse range of skills, which are categorized below:

- Arabic Language & Grammar
- Coding
- Diacritization
- Dialect Detection
- Entity Extraction
- Function Calling
- General Knowledge
- Hallucination
- Instruction Following
- Long Context
- MMLU
- Paraphrasing
- RAG QA
- Reading Comprehension
- Reasoning & Math
- Sentiment Analysis
- Structuring
- Summarization
- Translation (incl. Dialects)
- Transliteration
- Trust & Safety
- Writing (incl. Dialects)

4.5 Visual Comparison Tools

The leaderboard features radar charts (see Figure 1) to visually compare the skill profiles of selected models. These charts provide an intuitive overview of the models' relative strengths and weaknesses.

4.6 Deep Dive Reports

For each model, ABL generates a "deep dive" report that analyzes performance across a range of skills, highlighting its capabilities and limitations (see Figure 10). To promote transparency and facilitate further analysis, all underlying model outputs are made publicly available.

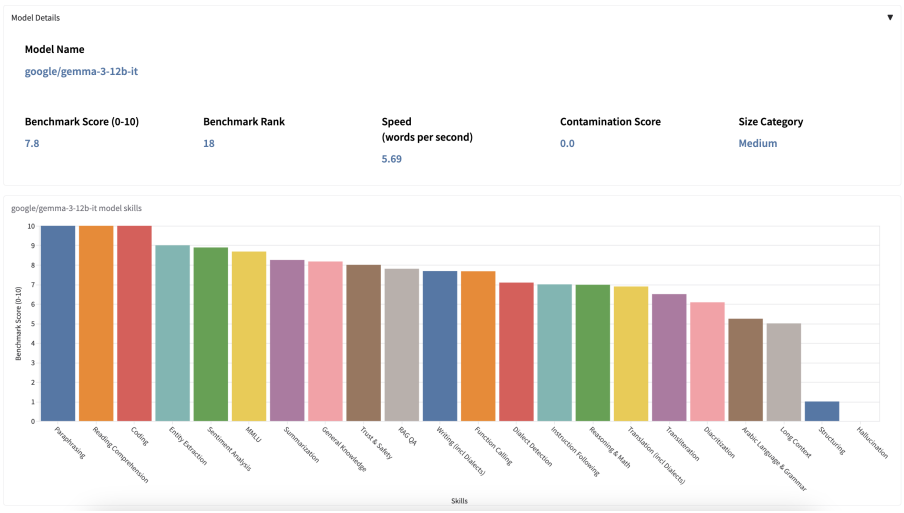

Figure 10: ABL Deep Dive section

4.7 Diversity of Model Sources

To ensure a comprehensive evaluation, ABL includes models from two distinct sources:

- **API-based** Closed-source, proprietary models evaluated via their vendor-provided APIs.
- **Hugging Face** Open-source models loaded from the Hugging Face Hub and evaluated using the *transformers* library.

5 Conclusion

We introduced the Arabic Broad Benchmark and Leaderboard (ABBL), a significant advance in Arabic LLM evaluation that addresses key gaps in the field. Our primary contributions include:

- **A human-validated, broad-coverage dataset** for a more reliable and comprehensive assessment of general Arabic proficiency.
- **A hybrid evaluation methodology** combining rule-based precision with LLM-as-judge scalability for nuanced scoring.
- **An innovative leaderboard** with features like *contamination detection*, *standardized speed metrics*, and *sub-leaderboards by size and skill*, setting a new standard for transparency and utility.

ABBL empowers researchers and industry practitioners to assess, compare, and select Arabic LLMs with unprecedented confidence and precision, fostering more rigorous and transparent development in the wider NLP community.

6 Limitations

While our proposed benchmark represents a significant step forward for the evaluation of Arabic LLMs, it is not without its limitations. These are discussed in detail below.

- **Breadth Over Depth:** The benchmark is designed to provide a holistic evaluation of a model's general Arabic language capabilities, prioritizing broad coverage over deep, task-specific analysis. Consequently, it may not be suitable for fine-grained comparisons between models on a single, specialized task, which would require a more focused assessment.

- **Question Scarcity in Some Categories:** A significant portion of the categories (11 out of 22) contain fewer than 10 questions. While this quantity is sufficient for some tasks (e.g., Structuring), it may not provide enough data for a robust evaluation in more nuanced areas, such as Instruction Following, which benefit from a wider array of test cases.
- **LLM Judge Consistency:** Large Language Models (LLMs) employed as judges exhibit inherent stochasticity, potentially yielding slightly different scores across multiple executions. Based on our evaluations, these fluctuations typically remain within a $\pm 1\%$ margin. Furthermore, our scoring methodology is only partially reliant on LLMs, as it also incorporates deterministic manual rules to mitigate this variability.
- **Constrained Long-Context Length:** To ensure broad accessibility and usability on standard hardware, tasks that test long-context understanding are capped at a 3,000-token input length. Processing contexts beyond this limit often incurs significantly higher memory and computational demands, which can lead to Out-Of-Memory (OOM) errors and render the benchmark inaccessible to users without high-end GPU resources.

References

[1] H. Sajjad, A. Abdelali, N. Durrani, and F. Dalvi, "Arabench: Benchmarking dialectal arabic-english machine translation," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 5094–5107.

[2] E. M. B. Nagoudi, A. Elmadany, and M. Abdul-Mageed, "Arat5: Text-to-text transformers for arabic language generation," *arXiv preprint arXiv:2109.12068*, 2021.

[3] E. M. B. Nagoudi, A. Elmadany, A. El-Shangiti, and M. Abdul-Mageed, "Dolphin: A challenging and diverse benchmark for arabic nlg," *arXiv preprint arXiv:2305.14989*, 2023.

[4] A. El Filali, M. ALOUI, T. Husaain, A. Alzubaidi, B. E. A. Boussaha, R. Cojocaru, C. Fourrier, N. Habib, and H. Hacid, "Open arabic llm leaderboard 2," https://huggingface.co/spaces/OALL/Open-Arabic-LLM-Leaderboard, 2025.

[5] A. El Filali, S. Albarri, A. Abouelseoud, S. Kamboj, N. Sengupta, and P. Nakov, "Arabic-leaderboards: Comprehensive evaluation of arabic large language models," https://huggingface.co/spaces/inceptionai/Arabic-Leaderboards, 2025.

[6] SILMA-AI, "Silma ragqa benchmark v1.0," https://huggingface.co/datasets/silma-ai/silma-rag-qa-benchmark-v1.0, 2024, version 1.0. [Data set].

[7] G. Team, "Gemma," 2024. [Online]. Available: https://www.kaggle.com/m/3301

[8] A. K. Singh, M. Y. Kocyigit, A. Poulton, D. Esiobu, M. Lomeli, G. Szilvasy, and D. Hupkes, "Evaluation data contamination in llms: how do we measure it and (when) does it matter?" 2024. [Online]. Available: https://arxiv.org/abs/2411.03923

[9] S. Golchin and M. Surdeanu, "Time travel in llms: Tracing data contamination in large language models," 2024. [Online]. Available: https://arxiv.org/abs/2308.08493

[10] ——, "Data contamination quiz: A tool to detect and estimate contamination in large language models," 2025. [Online]. Available: https://arxiv.org/abs/2311.06233

A Data Sources

Table 5: List of Data Sources, Counts, and Percentages.

Dataset Name	Count	Percentage (%)	Dataset Source
arabic_mmlu	70	14.893617	https://huggingface.co/datasets/MBZUAI/ArabicMMLU
arabic_mmlu_ht	51	10.851064	https://huggingface.co/datasets/MBZUAI/human_translated_arabic_mmlu
aragen-aragen-12	24	5.106383	https://huggingface.co/datasets/inceptionai/AraGen/viewer/AraGen-12-24
silma-ar-custom	24	5.106383	https://huggingface.co/datasets/silma-ai/silma-ar-custom-eval
acva	24	5.106383	https://huggingface.co/datasets/OALL/ACVA
silma-rag-qa	20	4.255319	Synthetic from SILMA.AI
aratrust	19	4.042553	https://huggingface.co/datasets/asas-ai/AraTrust-categorized
arabic-dialects-translation	18	3.829787	https://huggingface.co/datasets/BaselMousi/Arabic-Dialects-Translation/viewer/arabic-dialects-translation/test
mt-bench-oneturn	17	3.617021	MT-Bench (Translated by SILMA AI) https://huggingface.co/datasets/philschmid/mt-bench
alghafa	16	3.404255	https://huggingface.co/datasets/OALL/AlGhafa-Arabic-LLM-Benchmark-Native
silma-dialect-writing	15	3.191489	Synthetic from SILMA.AI
aradice-winogrande-winogrande	8	1.702128	https://huggingface.co/datasets/QCRI/AraDiCE-WinoGrande
arabic-text-diacritization	6	1.276596	https://huggingface.co/datasets/arbml/arabic_text_diacritization
arabic-to-arabizi	6	1.276596	https://huggingface.co/datasets/akhanafer/arabic-to-arabizi
silma-diacriticalization-quran	6	1.276596	Internal Data from SILMA.AI
un-parallel-corpus	6	1.276596	https://www.un.org/dgacm/en/content/uncorpus/download (Testset)
aradice-culture-all	6	1.276596	https://huggingface.co/datasets/QCRI/AraDiCE-Culture
aradice-truthfulqa-truthfulqa	6	1.276596	https://huggingface.co/datasets/QCRI/AraDiCE-TruthfulQA
llamalens-arabic-native	5	1.063830	https://huggingface.co/datasets/QCRI/LlamaLens-Arabic-Native
xlsum-arabic-ar	5	1.063830	https://huggingface.co/datasets/csebuetnlp/xlsum/viewer/arabic/test
madinah_qa	5	1.063830	https://huggingface.co/datasets/MBZUAI/MadinahQA
arabic-dialects-question	4	0.851064	https://huggingface.co/datasets/CNTXTAIO/arabic_dialects_question_and_answer
boolq-ar-test	3	0.638298	https://huggingface.co/datasets/OALL/AlGhafa-Arabic-LLM-Benchmark-Translated
silma-function-calling	3	0.638298	Synthetic from SILMA.AI
arabic-ifeval-default	3	0.638298	https://huggingface.co/datasets/inceptionai/Arabic_IFEval

Continued on next page

Table 5 continued from previous page

Dataset Name	Count	Percentage (%)	Dataset Source
silma-grammar-spelling	3	0.638298	Synthetic from SILMA.AI based on https://huggingface.co/datasets/AhmedSSabir/Gulf-Arabic-Tweets-2018-2020
silma-dataset-entityextraction	3	0.638298	Synthetic from SILMA.AI
arabicquoraduplicates-stsb-alue	3	0.638298	https://huggingface.co/datasets/AbderrahmanSkiredj1/ArabicQuoraDuplicates_stsb_Alue_holyquran_aranli_900k_anchor_positive_negative
sciq-ar-test	3	0.638298	https://huggingface.co/datasets/OALL/AlGhafa-Arabic-LLM-Benchmark-Translated
ragbench-tatqa-ar	3	0.638298	Translated https://huggingface.co/datasets/rungalileo/ragbench
silma-hallucination-ar	3	0.638298	Internal Data from SILMA.AI
copa_ext_ar	3	0.638298	https://huggingface.co/datasets/OALL/AlGhafa-Arabic-LLM-Benchmark-Translated
ragbench-emanual-ar	3	0.638298	Translated https://huggingface.co/datasets/rungalileo/ragbench
race_ar	3	0.638298	https://huggingface.co/datasets/OALL/AlGhafa-Arabic-LLM-Benchmark-Translated
qalbpreprocessedandmergedwithpunct	3	0.638298	https://huggingface.co/datasets/Ahmadsameh8/QalbPreprocessedAndMergedwithPunct
piqa_ar	3	0.638298	https://huggingface.co/datasets/OALL/AlGhafa-Arabic-LLM-Benchmark-Translated
arabic-gsm8k-default	3	0.638298	https://huggingface.co/datasets/Omartificial-Intelligence-Space/Arabic-gsm8k
silma-structuring-instructions	3	0.638298	Synthetic from SILMA.AI
arc_challenge_okapi	3	0.638298	https://huggingface.co/datasets/OALL/AlGhafa-Arabic-LLM-Benchmark-Translated
silma-synthetic-dialects	3	0.638298	Synthetic from SILMA.AI
arc_easy_ar	3	0.638298	https://huggingface.co/datasets/OALL/AlGhafa-Arabic-LLM-Benchmark-Translated
bbh-date-understanding	3	0.638298	Translated https://huggingface.co/datasets/lukaemon/bbh/viewer/date_understanding
wiki-lingua-ar	3	0.638298	https://huggingface.co/datasets/arbml/wiki_lingua_ar/viewer/default/test
dial2msa-lev-to	3	0.638298	https://github.com/khered20/Dial2MSA-Verified/tree/main
dial2msa-glf-to	3	0.638298	https://github.com/khered20/Dial2MSA-Verified/tree/main
dial2msa-egy-to	3	0.638298	https://github.com/khered20/Dial2MSA-Verified/tree/main
silma-folk-riddles	3	0.638298	Internal Data from SILMA.AI
silma-longcontext-ar	2	0.425532	Internal Data from SILMA.AI
toxigen_ar	2	0.425532	https://huggingface.co/datasets/OALL/AlGhafa-Arabic-LLM-Benchmark-Translated
tydiqa-goldp-ar	2	0.425532	https://huggingface.co/datasets/asas-ai/tydiqa-goldp-ar
alrage_qa	2	0.425532	https://huggingface.co/datasets/OALL/ALRAGE

Continued on next page

Table 5 continued from previous page

Dataset Name	Count	Percentage (%)	Dataset Source
ragbench-finqa-ar	2	0.425532	Translated https://huggingface.co/datasets/rungalileo/ragbench
arabic_exams	2	0.425532	https://huggingface.co/datasets/OALL/Arabic_EXAMS
ragbench-msmarco-ar	2	0.425532	Translated https://huggingface.co/datasets/rungalileo/ragbench
ragbench-covidqa-ar	2	0.425532	Translated https://huggingface.co/datasets/rungalileo/ragbench
openbook_qa_ext	2	0.425532	https://huggingface.co/datasets/OALL/AlGhafa-Arabic-LLM-Benchmark-Translated
musr-default-ar	2	0.425532	Translated https://huggingface.co/datasets/TAUR-Lab/MuSR/viewer/default/object_placements
mrcr-default-train	2	0.425532	Translated https://huggingface.co/datasets/openai/mrcr
jawaher-benchmark-test	2	0.425532	https://huggingface.co/datasets/UBC-NLP/Jawaher-benchmark
ifeval-ar-541	2	0.425532	Translated https://huggingface.co/datasets/google/IFEval/viewer/default/train
faitheval-unanswerable-v1	2	0.425532	Translated https://huggingface.co/datasets/Salesforce/FaithEval-unanswerable-v1.0
doda-10k-default	2	0.425532	https://huggingface.co/datasets/MBZUAI-Paris/DODa-10K
dial2msa-mgr-to	2	0.425532	https://github.com/khered20/Dial2MSA-Verified/tree/main
xquad-r-ar	2	0.425532	https://huggingface.co/datasets/google/xquad

B LLM-as-a-judge Prompts

This appendix contains the exact prompts used for the LLM-as-a-judge evaluation across different question types.

B.1 General Prompt

```
Your task is to judge the semantic matching of the PROVIDED_ANSWER vs
    ↪ the REFERENCE_ANSWER. REFERENCE_ANSWER is the ground truth.
Give a score from 0-10 with 10 being the best match (semantically).
If PROVIDED_ANSWER is more verbose but totally includes the meaning of
    ↪ REFERENCE_ANSWER then give a 10 score.
If the PROVIDED_ANSWER is a mathematical or reasoning answer and it
    ↪ does not actually match the final answer in REFERENCE_ANSWER then
    ↪  give score 3.
If the language of PROVIDED_ANSWER is not the same as REFERENCE_ANSWER
    ↪ then give 0 score and override any previous score.
Don't explain your answer. return the score only.
PROVIDED_ANSWER:\n {PROVIDED_ANSWER}
REFERENCE_ANSWER:\n {REFERENCE_ANSWER}
Final Score:
```

B.2 MCQ Prompt

```
PROVIDED_ANSWER and REFERENCE_ANSWER are answers to an MCQ question,
   ↪ your task is to judge if the answers match.
REFERENCE_ANSWER is the ground truth.
First answer this question: how many choices are listed in
   ↪ PROVIDED_ANSWER?
Give a score of 0 if answers do not match or more than one choice is
   ↪ included in PROVIDED_ANSWER, else 10 if the answers match.
If PROVIDED_ANSWER is more verbose but totally includes the meaning of
   ↪ REFERENCE_ANSWER then give a 10 score.
If both answers indicate the same answer choice number or letter then
   ↪ give a 10 score.
If the language of PROVIDED_ANSWER is not the same as REFERENCE_ANSWER
   ↪ then give 0 score.
Don't explain your answer. return the score only.
PROVIDED_ANSWER:\n {PROVIDED_ANSWER}
REFERENCE_ANSWER:\n {REFERENCE_ANSWER}
Final Score:
```

B.3 Writing Dialect Prompt

```
PROVIDED_ANSWER and REFERENCE_ANSWER are two written paragraphs.
REFERENCE_ANSWER is the ground truth.
Your task is to judge if they strictly match in terms of dialect.
Give a score from 0-10 with 10 meaning best match.
If the language of PROVIDED_ANSWER is not the same as REFERENCE_ANSWER
   ↪ then give 0 score.
If the dialect of PROVIDED_ANSWER is different from REFERENCE_ANSWER
   ↪ then give 0 score.
If one of the answers is in Modern Standard Arabic (MSA) while the
   ↪ other is not then give 0 score.
Don't explain your answer. return the score only.
PROVIDED_ANSWER:\n {PROVIDED_ANSWER}
REFERENCE_ANSWER:\n {REFERENCE_ANSWER}
Final Score:
```

B.4 Writing Grammar Prompt

```
Your task is to judge the match of grammatical parsing between
   ↪ PROVIDED_ANSWER vs the REFERENCE_ANSWER. REFERENCE_ANSWER is the
   ↪ ground truth.
Give a score from 0-10 with 10 being the best match.
If parsing details are missing in PROVIDED_ANSWER then give score 0.
If the language of PROVIDED_ANSWER is not the same as REFERENCE_ANSWER
   ↪ then give 0 score and override any previous score.
Don't explain your answer. return the score only.
PROVIDED_ANSWER:\n {PROVIDED_ANSWER}
REFERENCE_ANSWER:\n {REFERENCE_ANSWER}
Final Score:
```

B.5 Writing Reasoning Prompt

```
Your task is to judge the match two mathematical or reasoning answers,
   ↪  PROVIDED_ANSWER vs the REFERENCE_ANSWER.
REFERENCE_ANSWER is the ground truth.
Give a score from 0-10, with 10 indicating that both answers align in
   ↪ terms of reasoning steps and final conclusion.
If both answers match in reasoning but NOT the final conclusion then
   ↪ give score of 3.
If the language of PROVIDED_ANSWER is not the same as REFERENCE_ANSWER
   ↪ then give 0 score and override any previous score.
Don't explain your answer. return the score only.
PROVIDED_ANSWER:\n {PROVIDED_ANSWER}
REFERENCE_ANSWER:\n {REFERENCE_ANSWER}
Final Score:
```

C Scoring Rules

Table 6: Description of Scoring Rules and Their Counts.

Scoring Rule	Count	Description
AUTOMATED LLM_AS_A_JUDGE_MCQ	218	Automated scoring using an LLM as a judge for Multiple Choice Questions. (custom prompt)
AUTOMATED LLM_AS_A_JUDGE_GENERATION	173	Automated scoring using an LLM as a judge for text generation tasks. (custom prompt)
MANUAL ROUGE_SCORE	65	Manual calculation of ROUGE (Recall-Oriented Understudy for Gisting Evaluation) score.
MANUAL METEOR_SCORE	34	Manual calculation of METEOR (Metric for Evaluation of Translation with Explicit ORdering) score.
AUTOMATED LLM_AS_A_JUDGE_WRITING_DIALECT	30	Automated scoring using an LLM judge for dialect accuracy in writing. (custom prompt)
AUTOMATED LLM_AS_A_JUDGE_REASONING	21	Automated scoring using an LLM judge for reasoning capabilities. (custom prompt)
MANUAL WORDS_INTERSECTION	19	Manual check for the intersection of words between generated and reference text.
MANUAL DIACRITIZATION	12	Manual scoring of diacritization accuracy using Levenshtein distance + other conditions
MANUAL DIALECT_MATCHING	11	Manual scoring for how well a generated dialect matches a target dialect.
MANUAL RELATIVE_MIN_DISTANCE	6	Manual calculation of the relative change in distance (Levenshtein) between base to reference text and generated to reference text
MANUAL CLOSE_TO_REFERENCE_LENGTH	6	Manual check if the generated text length is close to the reference text length.
MANUAL MIN_DISTANCE	6	Manual calculation of minimum edit distance (Levenshtein).
MANUAL IS_VALID_JSON	5	Manual check if the output is valid JSON format.
AUTOMATED LLM_AS_A_JUDGE_GRAMMAR_IRAB	3	Automated LLM as a judge for grammar 'Irab'. (custom prompt)
MANUAL IFEVAL_1	3	Manual evaluation based on a specific 'IFEVAL' criterion (version 1).
MANUAL STRUCTURING_1	3	Manual evaluation of output structuring for each relevant question.
MANUAL IFEVAL_2	2	Manual evaluation based on a specific 'IFEVAL' criterion (version 2).
MANUAL MRCR_FIRST_LINE_MATCH	2	Manual check if the first line in generated matches reference by checking the Levenshtein distance of the first 100 characters only

D Multi-hop Synthetic Data Generation Prompt

```
Generate a story in Arabic in max 10 lines, generate a question about
   ↪ the story and answer it.
Each line in the story should depend on the previous one
The question needs to be complex in which it requires many parts of the
   ↪  story to be answered
Strictly follow the JSON Format below:
JSON Format:
‘‘‘
{"generations":
[{
    "story": "",
    "question": ""  ,
    "answer": ""
},
{
    "story": "",
    "question": ""  ,
    "answer": ""
}]
}
‘‘‘
TEXT:
```