YallaMorph: A Benchmark for Evaluating Arabic Morphological Generation in Large Language Models

Mahmoud Reda,[1] Salam Khalifa,[1,2] Reham Marzouk,[3] Nizar Habash[1]
Computational Approaches to Modeling Language (CAMeL) Lab
[1]New York University Abu Dhabi,
[2]Stony Brook University, [3]Mohamed bin Zayed University of Artificial Intelligence
{mahmoud.ali,salam.khalifa,nizar.habash}@nyu.edu, Reham.Marzouk@mbzuai.ac.ae

Abstract

Arabic morphology remains challenging for large language models, since fluent generation does not guarantee accurate morphosyntactic control. Existing Arabic evaluations mainly target downstream tasks and do not directly test controlled morphological generation from explicit lexical and feature-based input. We introduce **YallaMorph**, a large-scale benchmark for Arabic morphological generation covering verbs, nouns, adjectives, their cliticized forms, and invalid configurations. We evaluate multilingual and Arabic-oriented LLMs under diacritized and undiacritized settings over 600K benchmark entries. Results show that Arabic morphological generation remains difficult, especially for cliticized, unseen, and morphologically rare forms.

1 Introduction

Arabic morphology presents a major challenge for language generation. The interaction of templatic and concatenative morphology with morphosyntactic and cliticization features creates a large space of productive forms. Our central question is not simply whether Large Language Models (LLMs) can generate plausible Arabic, but whether they can systematically produce appropriate inflected forms for specific morphological features.

Most evaluations of Arabic LLMs focus on downstream tasks or broad generation quality (Nagoudi et al., 2022; Almazrouei et al., 2023; Koto et al., 2024), providing limited insight into whether models can systematically generate correct Arabic word forms from explicit lexical and morphological specifications. Existing morphological inflection benchmarks provide more controlled evaluation settings (Kodner et al., 2022; Goldman et al., 2023), but they do not capture the scale and Arabic-specific feature space required for evaluating modern LLMs.

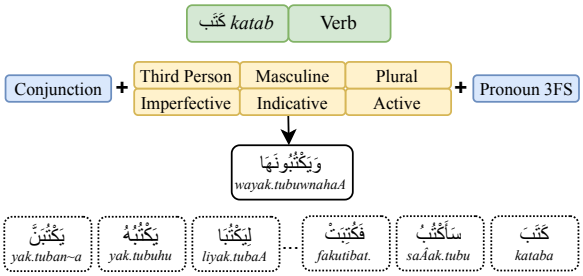

Figure 1: Morphological realization of Arabic lemma كَتَبَ *katab* 'write' as وَيَكْتُبُونَهَا *wayak.tubuwnahaA* 'and they write it' through morphosyntactic features and clitics. Transliterations follow HSB (Habash et al., 2007).

We introduce **YallaMorph**, a large-scale benchmark for controlled Arabic morphological generation.[1] As illustrated in Figure 1, the task maps a lemma, part-of-speech, gloss, and target feature bundle to the corresponding Arabic surface form, or to an indication that the requested configuration is morphologically invalid. The benchmark covers verbs, nouns, adjectives, and their cliticized forms, and contains over 600K evaluation instances sampled across root class, stem complexity, paradigm completeness, and lemma frequency.

We evaluate proprietary, multilingual, and Arabic-oriented instruction-tuned LLMs under both diacritized and undiacritized settings. Results show that controlled Arabic morphological generation remains challenging even for strong modern LLMs, particularly for cliticized, unseen, and morphologically rare forms.

Our contributions are: (a) introducing a **large-scale benchmark** for controlled Arabic morphological generation designed using a linguistically motivated sampling framework; and (b) **evaluating multilingual and Arabic-oriented LLMs** and providing detailed analysis across linguistic and distributional dimensions.

[1] The benchmark data is publicly available: https://github.com/CAMeL-Lab/YallaMorph

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 43299–43315
October 24-29, 2026 ©2026 Association for Computational Linguistics

2 Related Work

Morphological Inflection Morphological inflection is a well-established standalone task in NLP. The SIGMORPHON morphological (re)inflection shared tasks have served as a central benchmark series for this problem (Cotterell et al., 2016, 2017, 2018; McCarthy et al., 2019; Vylomova et al., 2020; Pimentel et al., 2021; Kodner et al., 2022; Goldman et al., 2023). These tasks evaluate systems that map lemmas and morphosyntactic feature bundles to inflected forms, and later editions increasingly emphasized generalization across typologically diverse languages, unseen lemmas, and unseen feature combinations. The 2022 and 2023 shared tasks in particular strengthened the evaluation setup through data splits targeting generalization to unseen lemmas (Kodner et al., 2022; Goldman et al., 2023). This line of work is closely tied to UniMorph (Batsuren et al., 2022), a broad-coverage universal schema for morphological annotation and inflection tables that represent forms through a lemma and a bundle of morphosyntactic features. UniMorph 4.0 covers 182 languages, including Arabic varieties such as Modern Standard Arabic (MSA), Egyptian Arabic, and Gulf Arabic in recent shared-task data. However, while the SIGMORPHON shared tasks yielded a diverse set of baselines and state-of-the-art systems, they were not designed as LLM benchmarks. Moreover, for our purposes, UniMorph does not provide exhaustive lexical coverage, full Arabic paradigm coverage, or cliticized inflected forms. In this work, we instead use Arabic-specific morphological resources that provide broader lexical coverage and richer paradigm generation.

Morphological Generators for Arabic Morphological analyzers and generators have been central to Arabic NLP since its early stages, providing explicit linguistic representations for a morphologically rich and complex language. Early work included finite-state and templatic approaches that modeled Arabic root-and-pattern morphology (Beesley et al., 1989; Kiraz, 1994; Beesley, 1998; Habash and Rambow, 2006; Smrž, 2007). A parallel line of work adopted lexicon- and compatibility-table-based resources, most notably the Buckwalter Arabic Morphological Analyzer and its successors (Buckwalter, 2002; Maamouri et al., 2010). Aragen/ALMORGEANA further extended Buckwalter-style lexical resources toward generation from lexeme-and-feature representations (Habash et al., 2005). More recently, Khairallah et al. (2024) introduced CamelMorph, a comprehensive open-source morphological analyzer and generator for MSA that builds on this tradition, reporting over 100K lemmas with rich morphological features and broad paradigm coverage. In this work, we use the CamelMorph lexicon and database to construct our controlled benchmark set, and we use its generation engine, exposed through CAMeL Tools (Obeid et al., 2020), to generate and validate reference inflected forms.

LLM Evaluation for Morphological Inflection Recently, benchmarking LLMs for morphological generation, particularly inflection, has gained traction as researchers increasingly probe LLMs for linguistic knowledge. Early work in this area used variants of the Wug test (Berko, 1958) to evaluate morphological productivity and generalization in LLMs across typologically diverse languages (Weissweiler et al., 2023; Anh et al., 2024). Ismayilzada et al. (2025) extended this line by evaluating morphological compositional generalization through both generative and discriminative tasks. However, these studies do not focus on Arabic or on controlled generation from explicit Arabic morphosyntactic feature specifications, which is the focus of our work.

Very few studies have focused on Arabic morphological generation in LLMs. IMPACT (Saeed et al., 2025) introduced a multilingual evaluation framework for inflectional morphology across five morphologically rich languages, including Arabic, but its Arabic component targets a narrower set of agreement and inflectional phenomena. The work by Alakeel et al. (2026) focuses solely on Arabic, specifically MSA, evaluating how LLM tokenizers align with Arabic morphological structure and how LLMs perform on productive root–pattern generation. It constructs a controlled test set with real and nonce roots and finds that tokenizer–morpheme alignment is neither necessary nor sufficient for successful morphological generation. In contrast, our work targets controlled Arabic morphological generation over an explicitly structured feature space, focusing on inflected-form realization and systematic sampling rather than root–pattern productivity alone, gender-focused generation, or broad multilingual probing. Furthermore, our benchmark dataset is substantially larger and more comprehensive.

3 Linguistic Background & Terminology

Arabic morphology is characterized by both **richness** and **complexity**. Its richness stems from large inflectional paradigms and productive cliticization involving conjunctions, prepositions, articles, and pronouns, yielding many possible surface forms for a single lemma (Table 2). Its complexity arises from interactions between templatic (root and pattern) and concatenative (affix or clitic) morphemes, often accompanied by orthographic and morphophonological alternations.

Relevant Terminology We briefly summarize the main terminology used throughout the paper, following CamelMorph (Khairallah et al., 2024). A **lemma** represents an abstraction over all inflectional forms of a lexical item (Habash et al., 2022). A **root** is an abstract consonantal sequence encoding core lexical meaning, while a **pattern** specifies the vocalic and templatic structure used to derive grammatical forms. A **stem** is the form produced by combining roots and patterns before affixation. Morphological representation is further organized through functional features such as gender, number, case, and state, as well as **part-of-speech (POS)**, which specifies the grammatical category, and the **gloss**, which provides an English semantic description. Finally, the framework distinguishes between **affixes**, which realize core morphosyntactic features within the **baseword**, and **clitics**, which include conjunctions, prepositions, definite article, and object and possessive pronouns.

Example Consider the noun lemma لَجْنَة *laj.naħ* 'committee' which illustrates multiple types of interactions. It is derived from the sound root *l.j.n* using the singular pattern *1a2.3aħ*. Its plural is the broken plural لِجَان *lijaAn*, which preserves the root while changing the pattern to *1i2aA3*. Clitic attachment further increases surface variation. For example, attaching the enclitic pronoun هَا *haA* 'her' produces لَجْنَتُهَا *laj.n+at+u+haA* 'her committee [nominative]', where ة *aħ* surfaces as تَ *at*. Single proclitic attachment yields forms such as اللِّجَانِ *Al+l∼ijaAni* 'the committees [genitive]' and لِلِجَانِ *li+lijaAnī* 'for committees [genitive]'. Combining both triggers orthographic assimilation, producing لِلِّجَانِ *li+l∼jaAni* 'for the committees [genitive]'.

4 YallaMorph Benchmark Design

In this section, we present the design details of the **YallaMorph** benchmark.

4.1 Design Principles

The design of **YallaMorph** is centered on evaluating Arabic morphological generation at scale while maintaining linguistic balance and interpretability. The benchmark includes over 600K sampled forms covering the major open POS classes of Arabic and capturing both inflectional and cliticization phenomena. To ensure broad linguistic coverage, the dataset is balanced across a wide range of morphophonological dimensions, including root classes, stem complexity, and paradigm completeness. In addition, **YallaMorph** incorporates frequency-balanced sampling over lemmas and morphological categories, enabling controlled evaluation of model generalization across both frequent and long-tail forms.

4.2 Data Source

We construct our benchmark from the lemma inventory and morphological analyses provided by CamelMorph MSA (Khairallah et al., 2024). Specifically, we use the lemma entries, part-of-speech labels, glosses, and morphological features to support sampling, categorization, and paradigm generation. Access to these analyses is handled through CAMeL Tools (Obeid et al., 2020).

4.3 POS Groups & Morphological Features

YallaMorph focuses on the three major open POS classes in Arabic: verbs, nouns, and adjectives. Within the verbal domain, we distinguish five core configurations: active perfective, passive perfective, active imperfective, passive imperfective, and command. The various POS groups differ in inflectional behavior, stem complexity, and compatibility with clitic attachment. To capture both core morphology and cliticization attachment phenomena, the benchmark is organized into two complementary subsets: a *baseword* subset focusing on inflectional morphology, and a *cliticization* subset focusing on interactions between stems and attached clitics.

4.4 Sampling Motivation

Arabic morphology exhibits substantial lexical and inflectional richness. Combining the large lemma

Category	Type	Explanation	POS Group
Lemma Frequency	High	High-frequency lemma	All
	Medium	Medium-frequency lemma	All
	Low	Low-frequency lemma	All
Root Class	Sound	Root with no weak letters, hamza, or gemination	All
	Geminated	Root with identical second and third radicals	All
	Hamzated	Root containing a hamza consonant	All
	Weak Initial	Root whose first radical is weak (*w* or *y*)	All
	Hollow	Root whose second radical is weak	All
	Defective	Root whose third radical is weak	All
	NTWS	Non-templatic word stem	Nouns, Adjectives
Paradigm Completeness	Full	Fully inflecting paradigm	All
	Masculine	Masculine only nominal paradigm	Nouns
	Feminine	Feminine only nominal paradigm	Nouns
Stem Complexity	Regular	No major orthographic or morphological alternations	All
	Defective Family	Stem allomorphs exhibit defective-family alternations	All
	Hamza Family	Stem allomorphs exhibit hamza-related alternations	All
	#N Family	Stem ends with letter ن *n*.	Verbs
	#T Family	Stem ends with letter ت *t*.	Perfective Verbs
	L# Family	Stem begins with letter ل *l*.	Nouns, Adjectives
	Diptote Family	At least one stem allomorph is a diptote ممنوع من الصرف.	Nouns, Adjectives

Table 1: Definitions of sampling categories used for lemma selection.

POS	Lemmas	Hypothetical Maximum - Clitics	Hypothetical Maximum + Clitics
Verbs	9,333	2,183,454	754,417,944
Nouns	19,844	1,071,576	2,730,375,648
Adjectives	6,924	373,896	952,687,008
Total	**36,101**	**3,628,926**	**4,437,480,600**

Table 2: Lemma and maximum instance counts by lexical class, with and without clitics.

inventory of CamelMorph MSA with feature bundles and clitic configurations yields an impractically large number of possible test instances (Table 2). We therefore construct the benchmark through guided sampling rather than exhaustive enumeration. The following subsections describe the linguistic and distributional dimensions underlying this sampling process. Our benchmark comprises over 600K carefully sampled entries.

4.5 Sampling Dimensions

We sample lemmas along linguistically motivated dimensions capturing root class, lemma frequency, paradigm completeness, and stem complexity. Each eligible lemma is assigned categorical labels for the relevant dimensions, and combinations of these labels define the sampling strata. Table 1 summarizes all category dimensions and their possible values, while Table 3 presents representative noun and active perfective verb (PVA) examples illustrating their combinations.

Lemma Frequency Lemma frequency provides a corpus-based estimate of how often each lemma is attested in a reference corpus. We estimate frequency using BAREC-10M (Elmadani et al., 2026), a large, balanced, multi-domain corpus of Arabic that is automatically morphologically tagged and lemmatized.

Our lemma-frequency analysis was conducted in parallel with the BAREC-10M annotation effort. The final BAREC-10M annotations and their additional lemma-cluster resolution procedure were not used because they were not yet available at the time of our analysis. We therefore independently analyzed the raw corpus using the morphological disambiguation component of CAMeL Tools.[2] For each token, we retained the top-ranked morphological analysis.

Lemmas for which our analysis recovered no occurrences were assigned a count of zero. For sampling, we group lemmas into three frequency bands: low, medium, and high. The low band contains lemmas with frequency 0; the medium band contains lemmas with frequencies from 1 to 20; and the high band contains lemmas with frequencies above 20.

[2]We used CAMeL Tools v1.5.5 (Obeid et al., 2020), with BERT-Disambig (Inoue et al., 2022) and the CamelMorph MSA v1.1 database (Khairallah et al., 2024).

Group	Frequency	Root Class	Paradigm	Stem Complexity	Example Lemma		Gloss
Verb	High	Sound	Full	Regular	رَجَع	*rajaς*	return
Verb	High	Hamzated	Full	#Hamza Family	اِبْتَدَأَ	*Aib.tadaÂ*	begin
Verb	Medium	Geminated	Full	#N Family	اِمْتَنَّ	*Aim.tan∼*	be grateful
Verb	Medium	Hollow	Full	Regular	نَاف	*naAf*	exceed
Verb	Low	Sound	Full	Regular	تَبَلْمَر	*tabal.mar*	be polymerized
Verb	Low	Defective	Full	Regular	رَخُو	*raxuw*	be loose
Noun	High	Sound	Full	#Hamza Family	شَرِيك	*šariyk*	partner
Noun	High	NTWS	Masculine	L# Family	لِيتْر	*liyt.r*	liter
Noun	Medium	Hollow	Feminine	#Defective Family	صِينِيَّة	*Siyniy∼aħ*	porcelain
Noun	Medium	Weak Initial	Masculine	#Diptote Family	مَوْشُور	*maw.šuwr*	prism
Noun	Low	Sound	Feminine	L# Family	لَاإِنْقِسَامِيَّة	*laAAin.qisaAmiy∼aħ*	indivisibility
Noun	Low	Geminated	Full	Regular	كَظِيظ	*kaĎiyĎ*	overfilled

Table 3: Representative verb and noun lemmas illustrating sampling categories across lemma frequency, root class, paradigm completeness, and stem complexity. Verb categories are from the perfective active subset.

Root Class Root class captures the phonological type of a lemma's root. We use seven root-class labels. Each lemma is assigned to exactly one root-class label. For lemmas with identifiable roots, overlapping classifications are resolved using the following priority hierarchy: *geminated* > *hamzated* > *weak-initial* > *hollow* > *defective* > *sound*. Lemmas with no identifiable root are treated as non-templatic word stems (NTWS) and assigned to the *NTWS* class. This procedure yields a single, consistent root-class label for each lemma and supports balanced sampling across major root types.

Paradigm Stem Complexity Paradigm stem complexity captures broad stem-related patterns that affect how forms are realized across a lemma's paradigm. We derive this dimension from stem-related information provided in CamelMorph MSA. We assign lemmas to one of seven stem-complexity families and some lemmas may satisfy *more than one* stem-complexity condition. Overlapping classifications are resolved using the following priority hierarchy: *#N Family* > *#T Family* > *L# Family* > *Defective Family* > *Hamza Family* > *Diptote Family* > *Regular*. This procedure yields a single, consistent stem-complexity label for each lemma and supports non-overlapping sampling across stem-complexity families. For example, the lemma شَرِيك *šariyk* 'partner' has a broken plural شُرَكَاء *šurakaA'* which ends in a Hamza (glottal stop). This places the whole lemma in the Hamza Family group. Table 1 defines the various stem-complexity families. Some of these apply to all POS, such as Hamza Family, while others are very specific, e.g., #T Family only applies to perfective verbs and captures the reduced spelling of t-initial suffixes with t-final verbs due to orthographic gemination (Shadda): فُتُّ *fut∼u* (*fut+tu*) 'I entered'.

Paradigm Completeness Paradigm completeness captures the extent to which a lemma realizes the expected range of gender–number combinations in its paradigm. We derive this dimension from stem-related information provided in CamelMorph MSA. Lemmas are assigned to one of three labels: *masculine-only*, *feminine-only* and *full*. Verbal and adjectival paradigms are always *full* in Arabic, whereas nouns can be of any of the three types. The inclusion of *masculine-only* and *feminine-only* paradigms leads to the possibility of Null forms, i.e., feature combinations that cannot be accommodated such as the feminine of مَكْتَب *mak.tab* 'office' is a masculine-only lemma. Simply adding the feminine ending to produce مَكْتَبَة *mak.tabaħ* 'library' yields an incorrect result. Around 13% of all the entries in **YallaMorph** correctly have a *null* gold reference.

4.6 Lemma Sampling Process

After assigning sampling dimension labels to all lemmas, we group them into strata defined by joint combinations of the sampling dimensions across the benchmark POS groups. Since these

strata vary greatly in size, we adopt a fixed per-stratum strategy, sampling up to 10 lemmas from each non-empty stratum while including all lemmas for smaller strata for the baseword subset of the benchmark. This preserves broad linguistic coverage while limiting over-representation of highly productive categories. The cliticization subset is constructed separately due to the large expansion introduced by proclitic and enclitic combinations. Instead of exhaustively enumerating all cliticized forms, we derive a reduced sample from the baseword subset while preserving coverage across lemma frequency and stem complexity, and we only include full paradigm completeness cases. For each relevant category, we sample up to 5 lemmas, yielding a manageable yet linguistically diverse clitic-focused evaluation set.

4.7 Arabic Morphological Generation Entries

We formulate the task as slot-level Arabic morphological generation. Given a lemma, POS, gloss, and a target feature bundle (henceforth, *morphological specification*), the model generates the corresponding inflected surface form. Outputs may consist of a single valid form, multiple valid realizations, or an indication that the requested configuration is invalid (*null* reference).

Multiple-reference cases account for 3.2% of the benchmark entries. In these cases, the gold reference set contains more than one valid realization for the same morphological specification.

Morphological specifications are POS-dependent. Verbal forms are specified using aspect, person, gender, number, voice, and mood, while nouns and adjectives use gender, number, case, and state. Clitic configurations are represented separately using positional proclitic (*prc0–3*) and enclitic (*enc0–1*) slots, with POS-specific constraints. The full list of features is provided in Appendix A. The selected clitic inventory and compatibility restrictions are provided in Appendix A.2.

4.8 Benchmark Statistics

Table 4 summarizes benchmark sizes across POS groups and experimental subsets. Although the cliticization subsets contain fewer lemmas, they are substantially larger in total size due to the multiplicative effect of generating all inflected baseword forms and their cliticized variants. Table 5 shows the frequency distribution of undiacritized target forms based on the CAMeLBERT Frequency List (Khalifa et al., 2021), highlighting the strong long-tail nature of the benchmark, with 59.8% of unseen forms, while an additional 12.9% correspond to null reference entries.

	Lemmas		Inflected Forms	
POS Group	**Baseword**	**Clitics**	**Baseword**	**Clitics**
Active PV	513	75	9,234	45,738
Passive PV	498	70	8,964	8,820
CV	433	60	7,794	6,480
Active IV	435	60	39,150	181,800
Passive IV	435	60	39,150	37,800
Nouns	1,887	75	101,898	67,950
Adj	594	75	32,076	76,950
Subtotal	4,795	475	238,266	425,538
Total	4,795		663,804	

Table 4: Benchmark subset sizes.

Frequency	**Instance Count**	**Percentage**
10M+	43	0.0
1M–10M	721	0.1
100K–1M	5,040	0.8
10K–100K	14,068	2.1
1K–10K	25,699	3.9
101–1K	36,022	5.4
11–100	43,746	6.6
1–10	55,908	8.4
0	397,179	59.8
Null	85,378	12.9
Total	663,804	100

Table 5: Frequency distribution of entries.

5 Evaluation

5.1 Metrics

Our main metric is **Any Match Accuracy**, computed at the instance level by comparing each prediction against its corresponding gold form or set of gold forms. For multi-form outputs, AMA counts a prediction as correct if at least one of the generated forms matches a gold form. Target configurations with a Null reference, representing morphologically invalid lemma–feature combinations, are counted as correct only when the model explicitly predicts a Null output.

We additionally report micro-averaged **Precision**, **Recall**, and **F1** to capture partial correctness in multi-form outputs. We evaluate predictions under three orthographic settings: *diacritized*, *undiacritized*, and *Alif/Ya/Hamza/Ta-Marbuta normalized*. We treat the diacritized setting as primary because it provides the strictest evaluation of Arabic morphological generation. Table 6 reports AMA and F1, while the corresponding Precision and Recall results are provided in Appendix C.

	10-Shot						Zero-Shot					
	Any Match Accuracy			F1 Score			Any Match Accuracy			F1 Score		
Model	**Diac**	**Undiac**	**Norm**	**Diac**	**Undiac**	**Norm**	**Diac**	**Undiac**	**Norm**	**Diac**	**Undiac**	**Norm**
GPT$_{en}$	**51.9**	**67.0**	**67.7**	**54.4**	**70.5**	**71.3**	**46.2**	**61.6**	**62.7**	**49.2**	**66.0**	**67.3**
Gemini$_{en}$	49.7	60.7	61.4	52.4	64.2	64.9	39.1	53.7	54.3	41.5	57.0	57.7
Fanar$_{en}$	21.5	36.5	37.2	22.5	38.3	39.1	5.1	19.4	19.9	5.3	20.3	20.9
Fanar$_{ar}$	15.8	26.8	27.2	16.6	28.1	28.5	2.7	15.5	15.8	2.8	16.1	16.5
Jais-70B$_{en}$	9.1	20.9	21.2	9.5	21.7	22.1	8.2	15.6	15.9	8.8	16.5	16.9
Jais-70B$_{ar}$	8.2	17.5	17.8	8.6	18.3	18.6	3.8	10.7	11.3	4.1	11.4	12.1
ALLaM$_{en}$	4.7	10.7	11.0	4.7	10.9	11.2	4.4	8.6	8.8	3.5	7.2	7.3
ALLaM$_{ar}$	4.5	10.0	10.3	4.1	9.3	9.5	3.1	9.0	9.2	2.8	8.4	8.6
Qwen$_{en}$	4.5	13.0	13.6	4.9	14.4	15.0	2.0	8.7	9.1	2.1	9.2	9.7
Jais-8B$_{en}$	3.4	10.1	10.2	3.3	9.9	10.0	2.7	6.4	6.7	2.3	5.7	6.0
Jais-8B$_{ar}$	2.3	8.2	8.5	2.1	7.6	7.9	2.6	7.7	7.9	2.1	6.4	6.6

Table 6: Overall benchmark results under the 10-shot and zero-shot prompting settings. Subscripts *ar* and *en* indicate the prompt language.

5.2 Compared Models

We evaluate seven instruction-tuned LLMs spanning commercial multilingual, open-weight multilingual, and Arabic-focused model families: **GPT-5.4**, **Gemini 3.1 Flash-Lite**, **Qwen2.5-7B-Instruct**, **Fanar-2-27B-Instruct**, **Jais-2-8B-Chat**, **Jais-2-70B-Chat** and **ALLaM-7B-Instruct-preview**[3]. This selection provides broad coverage of widely used general-purpose systems and models developed specifically for Arabic.

Prompt Design We use POS-specific prompts for verbs, nouns, and adjectives, with separate variants for clitic-aware experiments. Inputs include the lemma, POS, gloss, and the relevant morphological feature bundle. Models are instructed to return outputs in a fixed JSON schema. We evaluate both **zero-shot** and **few-shot** prompting settings, with the few-shot setting including 10 in-context examples. Prompts remain lightweight, with only minimal clarifications for features that showed consistent ambiguity in pilot experiments, such as nominal state distinctions, emphatic verbal moods, and clitic ordering.

We used English and Arabic versions of the prompts (Appendix E). Arabic and English prompts are used for **ALLaM**, **Fanar**, **Jais-70B** and **Jais-8B**; all other models use English prompts only.

Output Processing Model outputs are parsed using a unified post-processing pipeline that extracts candidate Arabic strings, handles minor formatting deviations from the requested JSON schema, and removes duplicate forms.

[3]Abbreviations used throughout: GPT-5.4 (GPT), Gemini 3.1 Flash-Lite (Gemini), Qwen2.5-7B-Instruct (Qwen), Fanar-2-27B-Instruct (Fanar), Jais-2-70B-Chat (Jais-70B), Jais-2-8B-Chat (Jais-8B), and ALLaM-7B-Instruct-preview (ALLaM).

5.3 Results

We first compare overall model performance, then analyze the best-performing model across linguistic and distributional dimensions.

Overall Performance Table 6 shows that GPT substantially outperforms all other models across both Any Match Accuracy and F1 metrics in all evaluation settings, followed by Gemini. In contrast, the Arabic-oriented open models perform considerably worse overall, suggesting that Arabic specialization alone does not guarantee accurate morphological generation. English prompts generally perform better, especially in the 10-shot setting. However, Arabic prompts perform better for ALLaM and Jais-8B on several zero-shot undiacritized and normalized metrics.

Detailed Results We consider the detailed results for GPT, the best-performing model under the English 10-shot setting. Table 7 shows substantial differences between the baseword and cliticization settings, particularly under diacritized evaluation, whereas the differences are smaller in the undiacritized and normalized settings. Across POS groups, adjectives and active perfective verbs perform best in the baseword setting, while nouns achieve the highest cliticized diacritized accuracy and passive perfective verbs lead under undiacritized and normalized evaluation. Lemma frequency has a relatively modest association with performance, whereas root-class differences are more pronounced under cliticization. Full paradigms outperform masculine-only and feminine-only paradigms, partly reflecting the difficulty of Null-reference configurations. Stem-Complexity rankings also vary across settings: #T and #N perform

Category	Group	Base			Clitics		
		Diac	Und.	Norm	Diac	Und.	Norm
All	All	57.4	67.9	68.3	48.8	66.6	67.4
POS Group	Active PV	79.1	**88.9**	**89.2**	58.6	73.1	73.7
	Passive PV	56.7	82.3	83.3	52.4	**86.5**	**88.9**
	CV	45.7	74.6	78.8	21.7	46.9	55.4
	Active IV	60.4	78.2	78.8	37.4	57.7	58.5
	Passive IV	60.5	78.2	78.8	56.5	77.3	78.6
	Nouns	46.2	49.7	49.8	**69.8**	80.4	80.9
	Adj	**82.3**	88.7	89.0	49.5	65.5	65.8
LF Lemma Frequency	High	**59.2**	**69.3**	**69.8**	**49.6**	**67.8**	**68.6**
	Medium	57.4	68.2	68.7	48.6	65.6	66.5
	Low	55.1	65.7	66.1	48.1	66.2	67.0
RC Root Class	Sound	**62.2**	70.5	70.6	55.8	72.0	72.2
	Geminated	57.0	68.2	68.4	46.8	66.7	66.8
	Hamzated	56.9	67.1	68.7	43.3	58.7	61.0
	Weak Initial	54.5	69.8	70.2	50.7	70.4	70.7
	Hollow	59.7	**70.6**	**70.8**	45.2	67.0	67.5
	Defective	56.2	65.8	66.3	53.2	69.9	70.6
	NTWS	49.1	55.6	55.6	**69.2**	**82.5**	**82.5**
PC Paradigm Completeness	Full	**67.8**	**81.9**	**82.5**	48.8	66.6	67.4
	Masculine	33.6	37.1	37.3	–	–	–
	Feminine	39.5	42.2	42.2	–	–	–
SC Stem Complexity	Regular	**62.3**	72.4	72.7	55.5	73.6	73.8
	Defective	53.7	67.5	68.2	46.0	66.8	67.5
	Hamza	55.1	61.3	62.7	44.3	57.0	59.4
	#T	61.7	**79.2**	79.1	39.8	68.2	68.2
	#N	60.0	79.1	**79.6**	42.5	65.7	65.9
	L#	55.1	59.3	59.4	**62.1**	72.3	72.4
	Diptote	49.0	51.2	51.2	61.7	**75.5**	**75.5**

Table 7: Detailed GPT results across benchmark dimensions under the 10-shot prompting setting.

strongly on undiacritized basewords but decline under cliticization, while L# and Diptote families perform comparatively well in the clitic subset.

Performance Across Surface Form Distributions Table 8 shows a strong correlation between frequency and performance. GPT achieves high accuracy on frequent forms, with performance gradually decreasing toward rarer forms. Null-reference configurations are the most difficult category, with 12.5% accuracy, and account for 12.9% of the benchmark. In contrast, performance on attested forms (*>0*) remains considerably higher, suggesting that Arabic morphological generation in LLMs is still strongly tied to exposure frequency and memorization, despite moderate generalization across low-frequency forms.

Performance Across Morphological-Specification Distributions Table 9 shows that performance strongly correlates with the corpus frequency of morphological specifications. GPT performs best on highly frequent feature configurations and degrades steadily toward rarer specifications. Completely unseen specifications are substantially harder, especially in the diacritized setting. Unlike lemma-frequency effects, these results suggest that models are highly sensitive not only to lexical exposure, but also to the distributional frequency of underlying morphosyntactic patterns.

Frequency	Examples	Diac	Undiac	Norm
10M+	43	81.3	88.4	93.0
1M–10M	721	79.5	90.2	90.6
100K–1M	5,040	79.4	88.1	88.3
10K–100K	14,068	77.2	88.0	88.2
1K–10K	25,699	73.8	86.3	86.7
101–1K	36,022	69.5	84.1	84.6
11–100	43,746	68.1	83.4	84.1
1–10	55,908	65.5	81.6	82.3
0	397,179	52.3	71.1	72.0
Null	85,378	12.5	12.5	12.5
> 0	181,247	69.5	83.9	84.5
All	663,804	51.9	67.0	67.7

Table 8: GPT 10-shot Any Match Accuracy (AMA) across surface-form frequency bands derived from CAMeLBERT Frequency.

Feature-Freq Percentile	Examples	%	Diac	Undiac	Norm
75–100%	13,236	2.0	65.2	80.1	80.2
25–75%	59,438	9.0	55.4	69.8	70.0
0–25%	356,131	53.7	55.2	67.9	68.5
0	234,999	35.4	45.2	64.2	65.3
All	663,804	100	51.9	67.0	67.7

Table 9: GPT-5.4 10-shot Any Match Accuracy (AMA) by feature-frequency percentile, computed from BAREC-10M counts of POS-specific morphological specifications excluding the lemma. Zero-count specifications form a separate group; nonzero specifications are ranked by count and divided into the shown percentile bands.

Overall, the results show that controlled Arabic morphological generation remains challenging even for strong modern LLMs, particularly for cliticized, unseen, and morphologically rare configurations. The findings further suggest that current models rely heavily on distributional exposure and struggle to robustly generalize across the full Arabic morphological space.

5.4 Error Analysis

Qualitative Analysis We randomly sampled 100 incorrect GPT outputs from the 10-shot baseword setting and another 100 incorrect outputs from the 10-shot cliticization setting for manual analysis. In the baseword setting, 41% of errors involved generating forms for invalid configurations (instead of *null*), 29% were diacritization-only differences,

Model	LPOS	PGN	CS	VAM	PRC	ENC
GPT (En)	**75.0**	**55.2**	**75.0**	**73.3**	**75.7**	**77.1**
Gemini (En)	68.0	52.8	71.0	67.6	65.7	71.6
Fanar (En)	67.2	43.6	71.9	61.9	51.1	69.2
Fanar (Ar)	70.2	41.5	73.3	60.8	41.7	66.7
Jais-70B (En)	63.4	37.1	69.6	53.1	35.3	55.4
Jais-70B (Ar)	59.1	40.2	66.9	55.6	30.6	48.8
ALLaM (En)	69.0	37.0	69.1	45.2	28.0	49.4
ALLaM (Ar)	64.2	36.8	69.9	60.4	32.6	51.8
Qwen (En)	50.5	25.7	62.0	39.5	38.8	50.4
Jais-8B (En)	68.1	31.8	72.9	47.7	33.8	52.2
Jais-8B (Ar)	70.9	34.1	72.6	53.9	30.4	51.5

Table 10: Comparison of models across morphosyntactic features: lemma-part-of-speech (LPOS), person–gender–number (PGN), case–state (CS), voice–aspect–mood (VAM), proclitics (PRC), and enclitics (ENC).

19% involved incorrect stems, incorrect affixes or both, 10% were missing valid outputs, and 1% was a Tatweel artifact.

In the cliticization setting, 39% were diacritization-only differences, 39% involved incorrect stems, incorrect affixes or both, 13% involved generating forms for invalid configurations, 5% were missing valid outputs, and 4% involved an incorrect clitic sequence form or order.

These results suggest that clitics were rarely the direct source of error. Instead, cliticization mainly increased the difficulty of realizing the host word correctly.

Quantitative Analysis We examine feature-group recovery for all models in the 10-shot setting. This analysis shows which types of morphological information are successfully preserved and where errors are concentrated. Each generated form is normalized using the same procedure as in the main evaluation and then analyzed using the same CAMeL Tools/CamelMorph MSA configuration used for benchmark construction. A group is recovered if at least one analysis matches the target values of all its features. We examine six groups: lemma-part-of-speech (`LPOS`); person, gender, and number (`PGN`); case and state (`CS`); voice, aspect, and mood (`VAM`); proclitics (`PRC`); and enclitics (`ENC`). Table 10 reports the recovery rates across systems.

The feature-group results should be interpreted alongside the output-cardinality distribution reported in Table 17. This is particularly important for Jais-8B, which generates multiple forms for 27.2% and 17.6% of instances under English and Arabic 10-shot prompting, respectively, compared with only 3.2% of the references. Because feature-group recovery assigns credit when at least one generated candidate has an analysis matching the target features, this over-generation increases the probability of recovering the correct lemma and POS. It may therefore partly explain the relatively high `LPOS` recovery of Jais-8B, particularly its score of 70.9% under Arabic prompting, despite its weaker recovery of other feature groups. However, generating additional candidates can also introduce incorrect forms, resulting in lower precision and F1. Feature-group recovery should therefore be interpreted as a measure of morphological coverage rather than exact generation accuracy.

Problematic Output Behavior Inspection of the complete test set revealed three recurring failure modes. First, Jais-8B (En, 0-shot) returned an **unrelated default** كتاب *ktAb* 'book' in 30,085 instances (4.5%), regardless of the requested analysis. Jais-70B did not exhibit this behavior. Second, Jais-70B (Ar) produced **romanized forms** in 3,473 outputs (0.52%), e.g. *ashrafu* and `>a_wa_la_na_athara`. Jais-70B (En) produced only five romanized outputs. Finally, Fanar (0-shot) occasionally generated Quranic material **unrelated to the requested morphological form**, e.g., the isolated sequence الٓمٓ *alif-lam-mim*, which opens Surat al-Baqarah, appeared 2,374 times (0.36%) in Fanar (En) and 3,318 times (0.50%) in Fanar (Ar).

6 Conclusion and Future Work

We presented **YallaMorph**, a large-scale benchmark for controlled Arabic morphological generation covering verbs, nouns, adjectives, cliticized forms, and invalid configurations. The benchmark combines broad linguistic coverage with controlled sampling across frequency, root class, paradigm completeness, and stem complexity. Our evaluation shows that Arabic morphological generation remains challenging for current LLMs, especially in diacritized and cliticized settings. Performance drops substantially for zero-frequency forms and rare morph specifications, suggesting strong reliance on distributional exposure and memorization rather than robust morphological generalization.

Future work will extend **YallaMorph** to additional Arabic varieties and contextual generation settings, while also enabling finer-grained analysis of errors involving agreement, stem selection, diacritization, and cliticization.

Limitations

This work focuses on Modern Standard Arabic as represented in CamelMorph MSA, and therefore does not cover the full diversity of Arabic dialects or mixed-register usage. The benchmark is also constrained by the coverage, analyses, and generation decisions of the underlying morphological resource. Although we include a large number of forms, the benchmark is still based on guided sampling rather than exhaustive enumeration of the full morphological space.

Because our lemma-frequency analysis was conducted in parallel with the BAREC-10M annotation effort, the final released annotations and associated lemma-cluster resolution procedure were not available at the time of our analysis. The reported lemma frequencies therefore reflect our independent analysis pipeline and may differ from frequencies derived from the final BAREC-10M annotations.

Our evaluation uses prompt-based generation with selected LLMs and prompt variants. Results may vary with alternative prompting strategies, decoding settings, or model versions. Finally, while exact-match evaluation is appropriate for controlled generation, it may not capture all cases of acceptable orthographic variation or context-dependent preference among valid forms.

Ethics Statement

This work introduces a benchmark for evaluating Arabic morphological generation. The dataset is derived from existing linguistic resources and does not contain private, personal, or user-generated sensitive information. The benchmark is intended to support more accurate and linguistically informed Arabic NLP systems.

Potential risks include over-reliance on Modern Standard Arabic as a representative form of Arabic and the use of benchmark results to make broad claims about Arabic language competence. Arabic is highly diverse across regions and registers, and performance on **YallaMorph** should not be interpreted as full coverage of Arabic linguistic ability. We encourage responsible use of the benchmark alongside evaluations for dialectal Arabic, downstream tasks, and human-centered applications.

We used AI writing assistance within the scope of "Assistance purely with the language of the paper" described in the ACL Policy on Publication Ethics.

Acknowledgments

This research was conducted using the High Performance Computing resources at New York University Abu Dhabi (NYUAD). We gratefully acknowledge the Center of Interdisciplinary Data Science and AI (CIDSAI) at NYUAD for its generous support. This work was supported in part by Google.org and the Google Cloud Research Credits program through the Gemini Academic Program. We sincerely thank our colleagues at CAMeL Lab, Ossama Obeid, Khalid Elmadani, and Mostafa Saeed, for their helpful conversations and support.

References

Yara Alakeel, Chatrine Qwaider, Hanan Aldarmaki, and Sawsan Alqahtani. 2026. Morphemes without borders: Evaluating root-pattern morphology in Arabic tokenizers and llms. *Preprint*, arXiv:2603.15773.

Ebtesam Almazrouei, Ruxandra Cojocaru, Michele Baldo, Quentin Malartic, Hamza Alobeidli, Daniele Mazzotta, Guilherme Penedo, Giulia Campesan, Mugariya Farooq, Maitha Alhammadi, Julien Launay, and Badreddine Noune. 2023. AlGhafa evaluation benchmark for Arabic language models. In *Proceedings of ArabicNLP 2023*, pages 244–275, Singapore (Hybrid). Association for Computational Linguistics.

Dang Anh, Limor Raviv, and Lukas Galke. 2024. Morphology matters: Probing the cross-linguistic morphological generalization abilities of large language models through a wug test. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 177–188, Bangkok, Thailand. Association for Computational Linguistics.

Mohamed Anwar, Abdelhakim Freihat, George Ibrahim, Mostafa Awad, Abdelrahman Atef Mohamed Ali Sadallah, Gurpreet Gosal, Gokul Ramakrishnan, Sarath Chandran, Biswajit Mishra, Rituraj Joshi, Ahmed Frikha, Etienne Goffinet, Abhishek Maiti, Ali El Filali, Sarah Al Barri, Samujjwal Ghosh, Rahul Pal, Parvez Mullah, Awantika Shukla, and 41 others. 2025. Jais 2: A family of Arabic-centric open large language models. Technical report, IFM.

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, and 7 others. 2025. ALLam: Large language models for arabic and english. In *The Thirteenth International Conference on Learning Representations*.

Khuyagbaatar Batsuren, Omer Goldman, Salam Khalifa, Nizar Habash, Witold Kieraś, Gábor Bella,

Brian Leonard, Garrett Nicolai, Kyle Gorman, Yustinus Ghanggo Ate, Maria Ryskina, Sabrina Mielke, Elena Budianskaya, Charbel El-Khaissi, Tiago Pimentel, Michael Gasser, William Abbott Lane, Mohit Raj, Matt Coler, and 76 others. 2022. UniMorph 4.0: Universal Morphology. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 840–855, Marseille, France. European Language Resources Association.

Kenneth Beesley. 1998. Arabic morphology using only finite-state operations. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages (CASL)*, pages 50–7, Montreal.

Kenneth Beesley, Tim Buckwalter, and Stuart Newton. 1989. Two-Level Finite-State Analysis of Arabic Morphology. In *Proceedings of the Seminar on Bilingual Computing in Arabic and English*.

Jean Berko. 1958. The child's learning of english morphology. *WORD*, 14(2-3):150–177.

Tim Buckwalter. 2002. Buckwalter Arabic morphological analyzer version 1.0. Linguistic Data Consortium (LDC) catalog number LDC2002L49, ISBN 1-58563-257-0.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfverberg, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 1–27, Brussels. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30, Vancouver. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared Task—Morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22, Berlin, Germany. Association for Computational Linguistics.

Khalid N. Elmadani, Adel Mahmoud Wizani, Hanada Taha Thomure, and Nizar Habash. 2026. A large and balanced multi-domain Arabic corpus annotated for morphology, syntax, and readability. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 11761–11775, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

FANAR TEAM, Ummar Abbas, Mohammad Shahmeer Ahmad, Minhaj Ahmad, Abdulaziz Al-Homaid, Anas Al-Nuaimi, Enes Altinisik, Ehsaneddin Asgari, Sanjay Chawla, Shammur Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed Elmagarmid, Mohamed Eltabakh, Asim Ersoy, Masoomali Fatehkia, Mohammed Qusay Hashim, and 18 others. 2026. Fanar 2.0: Arabic generative ai stack. *Preprint*, arXiv:2603.16397.

Omer Goldman, Khuyagbaatar Batsuren, Salam Khalifa, Aryaman Arora, Garrett Nicolai, Reut Tsarfaty, and Ekaterina Vylomova. 2023. SIGMORPHON–UniMorph 2023 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 117–125, Toronto, Canada. Association for Computational Linguistics.

Nizar Habash, Reham Marzouk, Christian Khairallah, and Salam Khalifa. 2022. Morphotactic modeling in an open-source multi-dialectal Arabic morphological analyzer and generator. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 92–102, Seattle, Washington. Association for Computational Linguistics.

Nizar Habash and Owen Rambow. 2006. MAGEAD: A morphological analyzer and generator for the Arabic dialects. In *Proceedings of the International Conference on Computational Linguistics and the Conference of the Association for Computational Linguistics (COLING-ACL)*, pages 681–688, Sydney, Australia.

Nizar Habash, Owen Rambow, and George Kiraz. 2005. Morphological Analysis and Generation for Arabic Dialects. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages (CASL)*, pages 17–24, Ann Arbor, Michigan.

Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, pages 15–22. Springer, Netherlands.

Go Inoue, Salam Khalifa, and Nizar Habash. 2022. Morphosyntactic tagging with pre-trained language models for Arabic and its dialects. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1708–1719, Dublin, Ireland. Association for Computational Linguistics.

Mete Ismayilzada, Defne Circi, Jonne Sälevä, Hale Sirin, Abdullatif Köksal, Bhuwan Dhingra, Antoine Bosselut, Duygu Ataman, and Lonneke Van Der Plas. 2025. Evaluating morphological compositional generalization in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1270–1305, Albuquerque, New Mexico. Association for Computational Linguistics.

Christian Khairallah, Salam Khalifa, Reham Marzouk, Mayar Nassar, and Nizar Habash. 2024. Camel morph MSA: A large-scale open-source morphological analyzer for Modern Standard Arabic. In *Proceedings of the 2024 Joint International Conference*

on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 2683–2691, Torino, Italia. ELRA and ICCL.

Salam Khalifa, Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. Camel Arabic Frequency Lists.

George Kiraz. 1994. Multi-tape Two-level Morphology: A Case study in Semitic Non-Linear Morphology. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 180–186, Kyoto, Japan.

Jordan Kodner, Salam Khalifa, Khuyagbaatar Batsuren, Hossep Dolatian, Ryan Cotterell, Faruk Akkus, Antonios Anastasopoulos, Taras Andrushko, Aryaman Arora, Nona Atanalov, Gábor Bella, Elena Budianskaya, Yustinus Ghanggo Ate, Omer Goldman, David Guriel, Simon Guriel, Silvia Guriel-Agiashvili, Witold Kieraś, Andrew Krizhanovsky, and 11 others. 2022. SIGMORPHON–UniMorph 2022 shared task 0: Generalization and typologically diverse morphological inflection. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 176–203, Seattle, Washington. Association for Computational Linguistics.

Fajri Koto, Haonan Li, Sara Shatnawi, Jad Doughman, Abdelrahman Sadallah, Aisha Alraeesi, Khalid Almubarak, Zaid Alyafeai, Neha Sengupta, Shady Shehata, Nizar Habash, Preslav Nakov, and Timothy Baldwin. 2024. ArabicMMLU: Assessing massive multitask language understanding in Arabic. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5622–5640, Bangkok, Thailand. Association for Computational Linguistics.

Mohamed Maamouri, Dave Graff, Basma Bouziri, Sondos Krouna, Ann Bies, and Seth Kulick. 2010. Ldc standard Arabic morphological analyzer (sama) version 3.1.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244, Florence, Italy. Association for Computational Linguistics.

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

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Tiago Pimentel, Maria Ryskina, Sabrina J. Mielke, Shijie Wu, Eleanor Chodroff, Brian Leonard, Garrett Nicolai, Yustinus Ghanggo Ate, Salam Khalifa, Nizar Habash, Charbel El-Khaissi, Omer Goldman, Michael Gasser, William Lane, Matt Coler, Arturo Oncevay, Jaime Rafael Montoya Samame, Gema Celeste Silva Villegas, Adam Ek, and 39 others. 2021. SIGMORPHON 2021 shared task on morphological reinflection: Generalization across languages. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–259, Online. Association for Computational Linguistics.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Mohammed J. Saeed, Tommi Vehvilainen, Evgeny Fedoseev, Sevil Caliskan, and Tatiana Vodolazova. 2025. IMPACT: Inflectional morphology probes across complex typologies. *Preprint*, arXiv:2506.23929.

Otakar Smrž. 2007. ElixirFM — Implementation of Functional Arabic Morphology. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages (CASL)*, pages 1–8, Prague, Czech Republic. ACL.

Ekaterina Vylomova, Jennifer White, Elizabeth Salesky, Sabrina J. Mielke, Shijie Wu, Edoardo Maria Ponti, Rowan Hall Maudslay, Ran Zmigrod, Josef Valvoda, Svetlana Toldova, Francis Tyers, Elena Klyachko, Ilya Yegorov, Natalia Krizhanovsky, Paula Czarnowska, Irene Nikkarinen, Andrew Krizhanovsky, Tiago Pimentel, Lucas Torroba Hennigen, and 9 others. 2020. SIGMORPHON 2020 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 1–39, Online. Association for Computational Linguistics.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schuetze, Kemal Oflazer, and David Mortensen. 2023. Counting the bugs in ChatGPT's wugs: A multilingual investigation into the morphological capabilities of a large language model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6508–6524, Singapore. Association for Computational Linguistics.

A Morphological Features

A.1 Baseword Morphological Features

The morphological feature space is POS-dependent: target slots for verbs, nouns, and adjectives are specified using different feature sets, as summarized in Table 11. Verb slots are defined by *aspect*, *person*, *gender*, *number*, *voice*, and *mood*.

Noun and adjective slots are defined by *gender*, *number*, *case*, and *state*.

A.2 Clitic Morphological Features

In addition to the core baseword morphological features, the benchmark models *clitic configuration* as a separate source of variation. We represent clitics using positional dimensions for proclitic slots (*prc0*, *prc1*, *prc2*, *prc3*) and enclitic slots (*enc0*, *enc1*). These dimensions are POS-sensitive: not all clitic positions are available for all parts of speech. For example, verbal forms do not use all nominal values of *prc0*, and nouns do not use the verbal enclitic configuration associated with *enc1*.

The selected clitic features and values used in our experiments are summarized in Table 12.

The full clitic space is very large when combined with core morphological feature bundles. We therefore use a restricted subset of clitic configurations rather than enumerating all combinatorially possible configurations. This subset is designed to preserve linguistically informative variation while keeping the evaluation feasible in terms of cost and runtime.

The selected clitics balance representativeness and coverage. For proclitics, we choose representative values, such as `wa_conj` and `li_prep`, whose attachment behavior is shared by other proclitics in their respective positional classes. For enclitics, we select values that cover different person, gender, and number combinations, as well as the main orthographic changes associated with attachment. Thus, while the selected clitics do not exhaust the full inventory, they cover the major attachment behaviors.

In particular, we retain selected values for clitic dimensions across verbs, nouns, and adjectives, and impose additional restrictions on nominal and adjectival configurations through interactions with features such as *state*.

This reduction substantially decreases the number of tested configurations. For example, in active imperfective verbs, the unrestricted clitic space would yield 20 configurations for intransitive verbs, 100 for transitive verbs, and 500 for ditransitive verbs. Under the restricted setup, these numbers are reduced to 8, 40, and 168, respectively. These reductions make the clitic-focused evaluation practically manageable while preserving sufficient variation to analyze clitic-sensitive generation.

Verb Feature	**Values**
person	1st, 2nd, 3rd
gender	masculine, feminine
number	singular, dual, plural
aspect	perfective, imperfective, command
voice	active, passive
mood	indicative, subjunctive, jussive, energetic (light), energetic (heavy)

Noun/Adj Feature	**Values**
gender	masculine, feminine
number	singular, dual, plural
case	nominative, accusative, genitive
state	construct, definite, indefinite

Table 11: Core baseword morphological features and their possible values for verbs, nouns, and adjectives.

Verb Clitic	**Values**
prc1	0, la_rc ,sa_fut ,la_emph , li_jus , li_sub
prc2	0, wa_conj
prc3	0, >a_ques
enc0	0, 1p_dobj, 1s_dobj, 3ms_dobj, 2fs_dobj
enc1	0, 1p_dobj, 1s_dobj, 3ms_dobj, 2fs_dobj

Noun/Adj Clitic	**Values**
prc0	0, Al_det
prc1	0, bi_prep, li_prep
prc2	0, wa_conj
prc3	0, >a_ques
enc0	0, 1p_poss, 1s_poss, 3ms_poss, 2fs_poss

Table 12: Selected clitic features and their possible values for verbs, nouns, and adjectives. The value labels are glossed as follows: `0` = no clitic; `la_rc` = response-conditional *la*; `sa_fut` = future marker *sa*; `la_emph` = emphatic particle *la*; `li_jus` = jussive *li*; `li_sub` = subjunctive *li*; `wa_conj` = conjunction *wa*; `>a_ques` = interrogative particle `>a`; `Al_det` = determiner *Al*; `bi_prep` = preposition *bi*; and `li_prep` = preposition *li*. In the pronominal labels, `1`, `2`, and `3` denote person; `s` and `p` denote singular and plural; `m` and `f` denote masculine and feminine; and `dobj` and `poss` denote direct object and possessive, respectively. Thus, for example, `3ms_dobj` denotes a third-person masculine singular direct object.

Compatibility Restrictions The selected clitic values are further constrained by the morphological

Group	Clitic value	Restriction
Noun/Adj	prc1 = 0	cas = ∗
Noun/Adj	prc1 = bi_prep	cas = g
Noun/Adj	prc1 = li_prep	cas = g
Noun	prc0 = 0	No additional state restriction
Noun	prc0 = Al_det	stt = d
Adj	prc0 = 0	No additional state restriction
Adj	prc0 = Al_det	stt ∈ {d, c}
Noun/Adj	enc0 = 0	stt = ∗
Noun/Adj	enc0 ≠ 0	stt = c
Noun/Adj	enc0 ≠ 0	Cannot co-occur with prc0 = Al_det

Table 13: Case and state based compatibility restrictions for nominal and adjectival clitic configurations. The symbol ∗ indicates that the feature is unrestricted.

Group	Condition	Allowed value
Active_IV	mod = ∗	prc1 = 0
Active_IV	mod ∈ {e, x}	prc1 = la_emph
Active_IV	mod = j	prc1 = li_jus
Active_IV	mod = s	prc1 = li_sub
Active_IV	mod = i	prc1 = sa_fut
Passive_IV	mod = ∗	prc1 = 0
Passive_IV	mod ∈ {e, x}	prc1 = la_emph
Passive_IV	mod = j	prc1 = li_jus
Passive_IV	mod = s	prc1 = li_sub
Passive_IV	mod = i	prc1 = sa_fut
Active_PV	–	prc1 ∈ {0, la_rc}
Passive_PV	–	prc1 ∈ {0, la_rc}
CV	–	prc1 = 0
CV	–	prc3 = 0

Table 14: Verbal-group compatibility restrictions for the selected proclitic configurations. Imperfective prc1 values are conditioned by mood (mod); command verbs (CV) do not admit a prc3 proclitic. The symbol ∗ denotes the underspecified mood value.

features of the baseword. Nominal and adjectival restrictions involve *case* and *state*, whereas verbal restrictions involve the verbal group and, for imperfective verbs, *mood*. Tables 13 and 14 summarize these restrictions.

B Model Configuration and Time & Cost Analysis

All models were evaluated with a temperature of 0 and a maximum output length of 256 tokens. The open-weight models were evaluated locally using NVIDIA A100 and V100 GPUs.

Table 15 reports the aggregate runtime and monetary cost of the complete zero-shot and 10-shot experiments. GPT-5.4 achieves the strongest overall performance among the evaluated LLMs and incurs the highest reported monetary cost, totaling $1,996.60. Gemini-3.1-Flash-Lite incurs a total cost of $1,047.10. The open-weight models have no API usage cost; however, the computational and infrastructure costs associated with running these models locally are not included.

Model	Time (min)	Cost ($)
GPT-5.4	8,104	1,996.60
Gemini-3.1-Flash-Lite	32,542	1,047.10
Fanar-2-27B-Instruct	29,392	0.00
Jais-2-8B-Chat	30,302	0.00
Jais-2-70B-Chat	44,331	0.00
Qwen2.5-7B-Instruct	36,322	0.00
ALLaM-7B-Instruct-preview	40,224	0.00
CamelMorph	233	0.00

Table 15: Aggregate runtime and reported monetary cost across all evaluated systems for the complete zero-shot and 10-shot experiments. Cost is reported in US dollars.

GPT-5.4 records the lowest aggregate runtime among the evaluated LLMs, requiring 8,104 minutes. Reference generation using the CAMeL Tools generator with the CamelMorph MSA database required 233 minutes without an API cost, making it substantially faster than all LLM-based systems. These aggregate runtimes reflect the model-specific configurations evaluated in this study, including the applicable prompting settings and prompt languages, and therefore should not be interpreted as normalized per-instance latency comparisons. Overall, GPT-5.4 provides the strongest LLM performance, whereas CamelMorph requires the least runtime and reported monetary cost.

C Precision and Recall Results

Table 16 reports the micro-averaged Precision and Recall scores corresponding to the F1 results presented in Table 6.

D Output Cardinality Analysis

Table 17 reports how often each model produces no answer, exactly one answer, or multiple answers. Most models predominantly produce one answer, while ALLaM and Jais-8B generate multiple answers more frequently, particularly in the zero-shot setting. This analysis describes output behavior rather than answer correctness.

E Prompts

We use two generalized prompt templates: one in English and one in Arabic. The templates are

	10-Shot						Zero-Shot					
	Precision			Recall			Precision			Recall		
Model	**Diac**	**Undiac**	**Norm**	**Diac**	**Undiac**	**Norm**	**Diac**	**Undiac**	**Norm**	**Diac**	**Undiac**	**Norm**
GPT$_{en}$	**52.6**	**68.2**	**69.0**	**56.3**	**73.0**	**73.8**	**48.8**	**65.6**	**66.8**	**49.7**	**66.5**	**67.7**
Gemini$_{en}$	52.2	63.8	64.5	52.7	64.6	65.3	41.8	57.3	58.0	41.2	56.7	57.4
Fanar$_{en}$	21.4	36.5	37.1	23.7	40.4	41.2	5.0	19.3	19.8	5.6	21.4	22.0
Fanar$_{ar}$	15.8	26.8	27.1	17.4	29.7	30.1	2.6	15.3	15.7	2.9	17.1	17.5
Jais-70B$_{en}$	9.0	20.6	20.9	10.1	23.1	23.5	8.5	16.0	16.4	9.0	17.0	17.4
Jais-70B$_{ar}$	8.1	17.4	17.7	9.1	19.4	19.7	3.9	11.0	11.7	4.2	11.8	12.6
ALLaM$_{en}$	4.4	10.1	10.3	5.2	11.9	12.1	2.8	5.9	6.0	4.7	9.2	9.4
ALLaM$_{ar}$	3.5	8.0	8.2	5.0	11.1	11.4	2.4	7.2	7.4	3.4	10.0	10.2
Qwen$_{en}$	5.0	14.5	15.1	4.9	14.3	14.9	2.0	8.9	9.3	2.2	9.6	10.0
Jais-8B$_{en}$	3.0	8.8	9.0	3.8	11.2	11.3	1.9	4.8	5.0	2.9	7.1	7.4
Jais-8B$_{ar}$	1.8	6.5	6.8	2.5	9.0	9.4	1.7	5.2	5.3	2.9	8.5	8.7

Table 16: Precision and recall under the 10-shot and zero-shot prompting settings. Bolded values are the largest in each column.

	10-Shot			Zero-Shot		
Model	**No Answer**	**Exactly One**	**Multiple**	**No Answer**	**Exactly One**	**Multiple**
Reference	12.9	84.0	3.2	12.9	84.0	3.2
GPT$_{en}$	4.7	93.3	2.0	9.9	87.6	2.4
Gemini$_{en}$	8.5	91.4	0.0	10.6	89.3	0.0
Fanar$_{en}$	0.0	99.8	0.1	0.0	99.5	0.5
Fanar$_{ar}$	0.0	99.8	0.2	0.5	97.7	1.8
Jais-70B$_{en}$	0.0	98.9	1.1	5.1	94.3	0.6
Jais-70B$_{ar}$	2.1	95.7	2.2	4.2	94.8	1.0
ALLaM$_{en}$	0.0	93.5	6.5	9.0	45.4	45.6
ALLaM$_{ar}$	0.0	87.2	12.8	0.2	78.8	21.1
Qwen$_{en}$	11.2	88.5	0.2	6.8	87.8	5.4
Jais-8B$_{en}$	0.0	72.8	27.2	0.6	60.4	38.9
Jais-8B$_{ar}$	1.4	81.0	17.6	1.1	48.1	50.8

Table 17: Distribution of output cardinality under the 10-shot and zero-shot prompting settings. *No Answer* denotes an empty or Null output, *Exactly One* denotes an output containing one answer, and *Multiple* denotes an output containing more than one answer. The reference row reports the cardinality distribution of the gold answers. Values are percentages and may not sum to exactly 100% because of rounding. Subscripts *ar* and *en* indicate the prompt language.

instantiated dynamically according to the part-of-speech, the presence of clitics, and the prompting setting. Text enclosed in braces denotes a placeholder replaced before the prompt is sent to the model.

The 10 in-context examples were randomly selected from a pool disjoint from the benchmark evaluation instances. None of the selected examples appears in the evaluated benchmark set.

The fully instantiated prompts used in the experiments, including the POS-specific rules, clitic rules, and 10-shot examples, are available in the YallaMorph repository (see Footnote 1).

The placeholder `{MORPHOLOGY_TYPE}` is replaced with *Adjectival*, *Nominal*, or *Verbal*, while `{WORD_TYPE}` is replaced with *adjective*, *noun*, or *verb*. The class-specific placeholders contain the features and interpretation rules associated with the selected part-of-speech. The clitic placeholders contain the applicable proclitic and enclitic inventories and their ordering rules; they are replaced with empty strings for baseword generation. The few-shot placeholder contains the relevant demonstrations in the few-shot setting and is empty in the zero-shot setting.

E.1 English Prompt

English Prompt Template

You are an Arabic `{MORPHOLOGY_TYPE}` Morphological Form Generator.

Task
Given an Arabic `{WORD_TYPE}` lemma, POS, gloss, and a specific set of morphosyntactic features, generate the corresponding fully diacritized Arabic `{WORD_TYPE}` form or forms.

Requirements

- Focus only on Arabic {MORPHOLOGY_TYPE} morphology.
- Output only the fully diacritized Arabic form or forms corresponding to the requested feature bundle.
- A feature bundle may yield no valid form, exactly one valid form, or multiple valid forms for the same slot.
- If the feature combination is not realizable for the lemma in Arabic, return an IMPOSSIBLE result.
- Do not provide explanations, glosses, comments, or reasoning.
- Do not invent forms.
- If multiple valid forms exist, return all valid forms.

Input Features
The input contains the lemma, POS, gloss, gender, and number, followed by the features associated with the selected part-of-speech.

{CLASS_SPECIFIC_FEATURES}

{CLITIC_FEATURES}

Critical Interpretation Rules

{CLASS_SPECIFIC_RULES}

{CLITIC_RULES}

Generate the form strictly from the complete feature bundle. If the combination of grammatical features or clitics is not morphologically realizable, return IMPOSSIBLE.

Output Format
For one or more valid forms:

```
{"arabic_forms": ["form1", "form2"],
          "status": "OK"}
```

If the requested form is impossible:

```
{"arabic_forms": null, "status":
          "IMPOSSIBLE"}
```

{N.FEW_SHOT_EXAMPLES}

E.2 Arabic Prompt

Arabic Prompt Template

أنت مولّد للصيغ الصرفية ل {MORPHOLOGY_TYPE_AR} العربيه.

المهمة

عند إعطاء مدخل معجمي {WORD_TYPE_AR} ، وقسم الكلام، والمعنى بالإنجليزية، ومجموعة محددة من السمات الصرفية، قم بتوليد الصيغة أو الصيغ العربية المشكّلة بالكامل المقابلة {TARGET_TYPE_AR} .

المتطلبات

ركّز فقط على تصريف {MORPHOLOGY_DOMAIN_AR} العربية.

أخرج فقط الصيغة أو الصيغ العربية المشكّلة بالكامل التي تطابق حزمة السمات المطلوبة.

قد تؤدي حزمة السمات إلى عدم وجود صيغة صحيحة، أو وجود صيغة صحيحة واحدة، أو وجود أكثر من صيغة صحيحة للخانة نفسها.

إذا كانت تركيبة السمات غير قابلة للتحقق صرفيًا لهذا المدخل المعجمي، فأرجع نتيجة IMPOSSIBLE .

لا تقدّم شروحات أو معاني أو تعليقات أو خطوات تفكير.

لا تخترع صيغًا.

إذا وُجدت عدة صيغ صحيحة، فأرجع جميع الصيغ الصحيحة.

سمات الإدخال

يحتوي الإدخال على المدخل المعجمي، وقسم الكلام، والمعنى بالإنجليزية، والجنس، والعدد، ثم السمات المرتبطة بقسم الكلام المحدد.

{CLASS_SPECIFIC_FEATURES_AR}

{CLITIC_FEATURES_AR}

قواعد تفسيرية مهمة

{CLASS_SPECIFIC_RULES_AR}

{CLITIC_RULES_AR}

ولّد الصيغة بدقة بناءً على حزمة السمات الكاملة. إذا كانت تركيبة السمات النحوية أو السوابق أو اللواحق غير قابلة للتحقق صرفيًا، فأرجع IMPOSSIBLE .

تنسيق الإخراج

عند وجود صيغة صحيحة واحدة أو أكثر:

```
{"arabic_forms": ["form1", "form2"],
          "status": "OK"}
```

إذا كانت الصيغة المطلوبة غير قابلة للتحقق:

```
{"arabic_forms": null, "status":
          "IMPOSSIBLE"}
```

{N.FEW_SHOT_EXAMPLES_AR}

Example Instantiation. The following example illustrates how the feature placeholders in the English and Arabic template are instantiated for a verbal item. It requests the third-person masculine singular imperfective indicative active form of the lemma كَتَبَ.

Baseword Example of the English Prompt

Input to process

LEMMA: كَتَب
POS: Verb
GLOSS: write
GENDER: Masculine
NUMBER: Singular
PERSON: 3rd
ASPECT: Imperfective
MOOD: Indicative
VOICE: Active

Baseword Example of the Arabic Prompt

المدخل المطلوب معالجته

المدخل المعجمي: كَتَب
قسم الكلام: فعل
المعنى بالإنجليزية: write
الجنس: مذكر
العدد: مفرد
الشخص: الغائب
الزمن: مضارع
العلامة الإعرابية: المرفوع
البناء: مبني للمعلوم

F Licenses

We list below the licenses of the data and tools used in this work, all of which are employed in accordance with their intended use.

- BAREC-10M Corpus (Elmadani et al., 2026): Creative Commons Attribution Share Alike 4.0
- CAMeLBERT Frequency List (Khalifa et al., 2021): Creative Commons Attribution Share Alike 4.0
- CAMeL Tools (Obeid et al., 2020) and CamelMorph (Khairallah et al., 2024) : MIT License
- CamelMorph MSA morphological database, lexicon, and specification data (Khairallah et al., 2024): Creative Commons Attribution 4.0 International (CC BY 4.0).
- ALLaM-7B-Instruct (Bari et al., 2025): Apache License 2.0
- Fanar-2-27B-Instruct (FANAR TEAM et al., 2026): Apache license 2.0
- Jais-2-8B-Chat (Anwar et al., 2025): Apache license 2.0
- Jais-2-70B-Chat (Anwar et al., 2025): Apache License 2.0
- Qwen2.5-7B-Instruct (Qwen et al., 2025): Apache License 2.0