Proper Noun Diacritization for Arabic Wikipedia: A Benchmark Dataset

Rawan Bondok,[1] Mayar Nassar,[1,2] Salam Khalifa,[1,3] Kurt Micallef,[1,4] Nizar Habash[1]
Computational Approaches to Modeling Language (CAMeL) Lab
[1]New York University Abu Dhabi, [2]Ain Shams University
[3]Stony Brook University, [4]Department of Artificial Intelligence, University of Malta
{rawan.bondok,nizar.habash}@nyu.edu, mayar.nassar@art.asu.edu.eg
salam.khalifa@stonybrook.edu, kurt.micallef@um.edu.mt

Abstract

Proper nouns in Arabic Wikipedia are frequently undiacritized, creating ambiguity in pronunciation and interpretation, especially for transliterated named entities of foreign origin. While transliteration and diacritization have been well-studied separately in Arabic NLP, their intersection remains underexplored. In this paper, we introduce a new manually diacritized dataset of Arabic proper nouns of various origins with their English Wikipedia equivalent glosses, and present the challenges and guidelines we followed to create it. We benchmark GPT-4o on the task of recovering full diacritization given the undiacritized Arabic and English forms, and analyze its performance. Achieving 73% accuracy, our results underscore both the difficulty of the task and the need for improved models and resources. We release our dataset to facilitate further research on Arabic Wikipedia proper noun diacritization.[1]

Figure 1: Four Arabic Wikipedia entries: (a) عمان *ςmAn* 'Oman', (b) عمان *ςmAn* 'Amman', (c) لندن *lndn* 'London', and (d) نخجوان *nxjwAn* 'Nakhchivan'. All titles lack diacritics. Lead sentences do not consistently use diacritics: (a) *ς**u**mAn*, (b) *ς**a**m∼**a**An*, and (c) *l**a**nd**a**n*; but (d) lacks diacritics, allowing multiple readings.

1 Introduction

Arabic Wikipedia, like other language editions, has been a valuable resource for both its readers and NLP research. In this paper, we focus on a particular limitation rooted in Arabic's abjad orthography, where diacritics are typically omitted (Elgamal et al., 2024) except for children's books and religious texts. This omission leads to ambiguity in pronunciation and interpretation, especially for proper nouns. Some Arabic Wikipedia articles address this issue by providing partial or full diacritization in their lead sentences. For instance, عمان *ςmAn*[2] can refer to either عُمَان *ςumaAn* 'Oman' or عَمَّان *ςam∼aAn* 'Amman' depending on the diacritization (Figure 1). But more often than not, these diacritics are missing. In our dataset we found 99.45% of all entries had no diacritics. Our intention is to solve this limitation.

The work presented in this paper lies at the intersection of three commonly but often independently studied Arabic NLP tasks: *transliteration*, *diacritization*, and *lemmatization*.

Transliteration is the mapping of words, primarily proper nouns, from one script to another, usually in the context of machine translation (Beesley, 1997; Benites et al., 2020; Chen et al., 2018). It poses challenges due to misalignments between scripts, differences in representing phonology and

[1]https://github.com/CAMeL-Lab/CamelProp
[2]Arabic HSB Romanization (Habash et al., 2007).

Proceedings of the 2nd Workshop on Advancing Natural Language Processing for Wikipedia (WikiNLP 2025), pages 31–44
August 1, 2025

morphology, and historical ad hoc conventions.

Diacritization, or diacritic restoration, aims at recovering omitted diacritics in languages that rely on them for disambiguation (Alqahtani et al., 2019; Darwish et al., 2017; Abandah et al., 2015). While both transliteration and diacritization have been well studied for Arabic, they are typically treated in isolation. An exception is the work of Mubarak et al. (2009), which considers both in the context of Arabic to English proper noun transliteration.

Lemmatization maps inflected words to their base forms. This is particularly important for morphologically rich languages such as Arabic (Roth et al., 2008). In the context of Wikipedia entries, providing the lemmas is useful to readers as it gives them a grounding on how to interpret and later inflect the word forms properly.

More concretely, we focus here on mapping pairs of undiacritized Arabic proper nouns and their English glosses to fully diacritized and lemmatized Arabic forms. The task can be viewed as partial transliteration, where Roman-script vowels help infer (or transliterate into) Arabic diacritical marks. For example, نخجوان *nxjwAn* 'Nakhchivan' (from Figure 1) should ideally be mapped to نَخْجِوَان *nax.jiwaAn*, rather than incorrect alternatives like نِخْجَوَان *nix.jawaAn* or نُخْجِوَان *nux.jiwaAn*.

We present a new dataset of 3,000 unique Arabic Wikipedia proper nouns annotated with gold lemma-level diacritizations. Each entry is paired with its English Wikipedia equivalent, enabling the study of joint diacritization and transliteration. We benchmark GPT-4o (OpenAI et al., 2024), which shows promising results but struggles with spelling variants and ambiguity. The dataset covers a range of named entities (people, places, and organizations) and includes 3,362 total pairs to reflect multiple valid diacritizations based on the gloss.

Our contributions are:

- A publicly available gold-standard dataset of Arabic Wikipedia proper nouns with English equivalents.[1]
- A GPT-4o benchmark and detailed error analysis for Arabic proper noun diacritization.

The remainder of this paper is structured as follows. Section 2 outlines Arabic linguistic aspects. Section 3 reviews related work. Sections 4 and 5 describe our dataset and annotation process. Section 6 presents evaluation results and error analysis.

Diacritic	Example		
Fatha	بَ	*ba*	/ba/
Damma	بُ	*bu*	/bu/
Kasra	بِ	*bi*	/bi/
Shadda	بّ	*b∼*	/bb/
Sukun	بْ	*b.*	/b/
Dagger Alif	بٰ	*bá*	/ba:/
Shadda + Fatha	بَّ	*b∼a*	/bba/
Shadda + Damma	بُّ	*b∼u*	/bbu/
Shadda + Kasra	بِّ	*b∼i*	/bbi/
Long vowel /a/	بَا	*baA*	/ba:/
Long vowel /u/	بُو	*buw*	/bu:/
Long vowel /i/	بِي	*biy*	/bi:/
Shadda + Long vowel /a/	بَّا	*bbaA*	/bba:/
Shadda + Long vowel /u/	بُّو	*bbuw*	/bbu:/
Shadda + Long vowel /i/	بِّي	*bbiy*	/bbi:/
Glide w	بَوْ	*baw.*	/baw/
Glide y	بَيْ	*bay.*	/bay/

Table 1: Examples of Arabic diacritics, their transliterations, and phonological values. We exclude nunation diacritics as they are not used in our lemmas.

2 Linguistic Background

2.1 Arabic Diacritization

Arabic orthography follows an *Abjad* system (Daniels, 2013), where letters encode consonants and diacritical marks represent short vowels, nunation (case endings), gemination, and vowel absence. Diacritic clusters are typically limited to a Shadda (◌ّ ∼) followed by a short vowel or nunation diacritic. Three letters, ا *A*, و *w*, and ي *y* (henceforth *AWY*), encode long vowels when preceded by a matching short vowel and not followed by any diacritic: ◌َا *aA* (/a:/), ◌ُو *uw* (/u:/), and ◌ِي *iy* (/i:/). These letters are often used with foreign name transliterations to mark the vowel quality independent of length, e.g., بين *byn* 'Ben' or 'Bean'. The letters و and ي also serve as glides (/w/ and /y/) when preceded by ◌َ *a* and followed by a sukun (◌ْ .). The letter ا *A* functions as a carrier for initial short vowels (*Alif Wasla*, ٱ *Ä*). Additionally, Arabic

Input Arabic		Gloss	Lemma Arabic		Transformation
الست	*Alst*	Al-Sit	سِتّ	*sit∼*	DET → ϕ
الواس	*AlwAs*	Elvas	إِلْوَاس	*Ǎil.waAws*	Bare Alif → Alif Hamza
العجم	*Alςjm*	Al-Ajam	عَجَم	*ςajam*	DET → ϕ
الغظاة	*AlγĎAħ*	Al-Ghadhah	غَظَاة	*γaĎaAħ*	DET → ϕ
فنزويليون	*fnzwylywn*	Venezuelans	فِنِزْوِيلِيّ	*finiz.wiyliy∼*	3MP → ϕ
الجيبوتيون	*Aljybwtywn*	Djiboutians	جِيبُوتِيّ	*jiybuwtiy∼*	DET+3MP → ϕ

Table 2: Examples of lemmatization transformations from Arabic input (inflected) words to canonical lemmas, with English glosses and corresponding changes.

uses letters with attached Hamza diacritics, e.g., أ *Â*, إ *Ǎ*, آ *Ā*, ؤ *ŵ*, and ئ *ŷ*. The omission of Hamzas is treated as a spelling error and corrected during diacritization.

See Table 1 for examples, and Darwish et al. (2017) and Elgamal et al. (2024) for more details on Arabic diacritics.

2.2 Arabic Lemmatization

In Arabic morphology, the lemma is the canonical form (also known as citation form) of a word that abstracts over its inflected variants, including gender, number, person, and case, as well as attached clitics (Roth et al., 2008; Habash, 2010). Table 2 shows examples of input forms and their corresponding lemmas. In our context, lemmatization is simpler than in free-form text: we focus only on proper nouns, an English gloss is available to guide vowelization, and clitics are rare. The main challenges are distinguishing between base-word and determiner uses of ال *Al* (DET) initial substring (see Table 2 rows 1-2), and handling plural endings (3MP) ون *uw* in demonyms (Table 2 rows 5-6).

2.3 Arabic Transliterations

Transliteration from Roman script to Arabic script presents several challenges, primarily due to the misalignment between the phonology of the original language and its Roman script orthography, as well as differences between the phonology of the original languages and Arabic. Arabic, for example, has fewer vowels (6 in Arabic vs. 15 in English), and some missing (no /p/ or /v/) and additional consonants (e.g., emphatic /d/ and /q/). Arabic dialects vary in phonology, including sound quality, letter mapping, and syllabification, leading to multiple valid transliterations. For instance, the borrowed word 'plastic' can have different pronunciations and spellings, reflecting variations in vowels and syllabification (see Table 3). During annotation, we followed Wikipedia spelling and aligned with the English gloss. The team included Egyptian, Sudanese, and Levantine speakers, with an Egyptian speaker as the primary annotator.

	Pronunciation	Arabic	Transliteration
(a)	/bla:stik/	بْلَاسْتِك	*b.laAs.tik*
(b)	/bila:stik/	بِلَاسْتِك	*bilaAstik*
(c)	/bla:stik/	بْلَاسْتِيك	*b.laAs.tiyk*
(d)	/bila:stik/	بِلَاسْتِيك	*bilaAstiyk*
(e)	/bla:sti:k/	بْلَاسْتِيك	*b.laAs.tiyk*
(f)	/bila:sti:k/	بِلَاسْتِيك	*bilaAstiyk*
(g)	/bala:sti:k/	بَلَاسْتِيك	*balaAstiyk*
(h)	/ibla:stik/	إِبْلَاسْتِيك	*Aib.laAs.tiyk*

Table 3: Variants of the pronunciation and transliteration of the Arabic word for 'plastic'. Three basic spellings: (a-b) بلاستك *blAstk*, (c-g) بلاستيك *blAstyk*, and (h) ابلاستك *AblAstk*, with various diacritizations.

3 Related Work

3.1 Diacritization in Arabic NLP

Arabic diacritization has been extensively studied using both statistical and neural methods. Some approaches treat it as a standalone task (Zitouni et al., 2006; Mubarak et al., 2019), while others integrate it into multitask learning frameworks alongside linguistically related tasks such as part-of-speech tagging (Habash and Rambow, 2005; Alqahtani et al., 2020).

A commonly adopted strategy involves the use of morphological analyzers. For instance, Camelira (Camel Tools) implements an analyze-and-disambiguate pipeline: a morphological analyzer generates candidate analyses, which are then ranked by a classifier (Obeid et al., 2020, 2022). Similarly, Farasa uses morphological patterns to diacritize words (Darwish et al., 2017).

Systems such as Farasa and Camel Tools have demonstrated strong performance on sentence-level diacritization tasks. However, these systems are not directly applicable to our task, which centers on isolated proper nouns, adheres to a task-specific diacritization schema, and incorporates lemma mapping. Unlike sentence-based systems that leverage surrounding context for disambiguation, our task involves context-free diacritization, which poses distinct challenges (see Section 5.1).

3.2 Lemmatization in Arabic NLP

Lemmatization is another core task in Arabic NLP, and several tools offer robust performance across a variety of syntactic categories (Obeid et al., 2020, 2022; Jarrar et al., 2024). However, our lemmatization task has a narrower scope: it is limited to proper nouns that have a limited inflectional space (see Section 5.1 for further details on our lemmatization space).

3.3 Transliteration in Arabic NLP

Earlier research on Arabic–English transliteration relied on statistical approaches (Abduljaleel and Larkey, 2004), followed by more targeted work on proper nouns using models such as phonemic memory networks (Tian et al., 2022). A persistent challenge in this area is the lack of standardization in transliterating foreign names into Arabic, a problem exacerbated by the omission of diacritics (Aziz, 1983; Odisho, 1992).

To address the lack of standardization and limited resources, we introduce a new dataset and annotation guidelines specifically designed for the task of utilizing proper noun transliteration as a signal for Arabic diacritization.

Prior efforts investigated the intersection of transliteration and diacritization, such as Mubarak et al. (2009) and Darwish et al. (2017). Mubarak et al. (2009) used diacritization as a preprocessing step to transliteration. Although, the approach presented in Darwish et al. (2017) for automatically diacritizing transliterated words included leveraging English transliterations to generate Arabic diacritized proper nouns, both their training and test sets were limited in size (500 and 200 instances, respectively). Our resource, in contrast, is publicly available, much larger (3,000 diacritized lemmas), and benchmarked for robust evaluation and development.

3.4 Arabic Proper Noun Resources

Although various Arabic proper noun datasets exist, they often suffer from limited accessibility, lack of diacritics, or domain constraints. For example, Matthews (2007) compiled a list of 10,001 Arabic names, but the dataset is not publicly available. Eryani and Habash (2021) provide automatically Romanized Arabic bibliographic entries without diacritics, and both the Dan database (Halpern et al., 2009) and SAMA Graff et al. (2009) include diacritized proper nouns, but they were mainly collected form news sources.

Khairallah et al. (2024) released a large set of proper nouns as part of their CamelMorph Arabic morphological analyzer (henceforth CAMELPROP, CP for short). The dataset consists of two distinct portions: (a) CP-SAMA, which extends the SAMA (Graff et al., 2009) proper-noun list and updates their diacritizations; and (b) CP-WIKI which comprises 63K entries extracted from a Wikidata dump (14-Mar-2023).[3] The CP-WIKI was filtered by Khairallah et al. (2024) to include only single word entities in Arabic and English, and covering only personal and family names, locations and organizations. Unfortunately, Khairallah et al. (2024) did not provide diacritizations for the CP-WIKI portion. Our interest in this topic started by this problem in their open-source resource, which was not usable for our purposes. We discuss these datasets further in Section 4.

In this work, we present the first publicly available dataset of maximally manually diacritized and lemmatized Arabic proper nouns on a portion of the CP-WIKI dataset sourced from Wikimedia and manually annotated using English equivalents in a consistent and standardized annotation scheme. To support future work, we also release detailed annotation guidelines and provide the first benchmark of GPT-4o's performance on this task, offering a new resource for evaluating Arabic proper noun diacritization and transliteration.

[3] https://dumps.wikimedia.org/wikidatawiki/entities/

	CP-SAMA	CP-WIKI	CP-WIKI-D3K
Unique Arabic	6,022	63,417	3,000
Arabic-English Entries	7,202	71,251	3,362
English glosses per entry	1.20	1.12	1.12
Average Freq	205,077	97,438	61,544
Median Freq	11,732	87	75
Average Freeman Score	0.92	0.91	0.91
Diacritizations	Yes	No	Yes

Table 4: Comparison of dataset statistics across CP-SAMA, CP-WIKI, and the annotated subset CP-WIKI-D3K.

Class	CP-WIKI	CP-WIKI-D3K
Location	77.1%	85.2%
Name	25.5%	35.0%
Organization	2.0%	2.0%

Table 5: Distribution of different named entity classes across CP-WIKI and CP-WIKI-D3K

4 Datasets

We work with the CAMELPROP dataset, released as part of CamelMorph, an Arabic morphological analyzer, by Khairallah et al. (2024). As noted in Section 3, it consists of two parts: CP-SAMA and CP-WIKI. We randomly selected 3,000 unique Arabic-script proper nouns from CP-WIKI for manual annotation, forming our dataset CP-WIKI-D3K.

Table 4 compares the three datasets in terms of unique Arabic entries and full Arabic–English gloss pairs, average and median frequency and Arabic-English phonological similarity. For frequency we used the Arabic Frequency list from Khalifa et al. (2021). For phonological similarity, we used the Freeman similarity score (Freeman et al., 2006). The original data included multiple glosses per Arabic word (12–20% extra on average). We normalized this by splitting them into separate one-to-one pairs. For example, آنا *ĀnA*, glossed as 'A'ana; Ana; Anna', became three distinct entries: (آنا *ĀnA*, 'A'ana'), (آنا *ĀnA*, 'Ana'), and (آنا *ĀnA*, 'Anna'). Thus, our 3,000 Arabic words expanded to 3,362 Arabic–gloss pairs. While phonological similarity is only slightly lower in CP-WIKI-D3K and CP-WIKI, the overall frequency in CP-WIKI and CP-WIKI-D3K is significantly lower than CP-SAMA, highlighting the importance of modeling the diacritization of low-frequency proper nouns in Wikipedia and NLP.

In addition to frequency and phonological similarity, we examined the distribution of named entity categories, namely, personal and family names, locations, and organizations, across both the original CP-WIKI dataset and the manually annotated subset, CP-WIKI-D3K. The distributions were broadly similar, with location entities being the majority in both (CP-WIKI: 77.1%, CP-WIKI-D3K: 85.2%), followed by names and organizations. This consistency supports the representativeness of CP-WIKI-D3K for studying diacritization across entity types. Table 5 reports the detailed percentage breakdown of entity classes in both datasets.

5 Data Annotation

In this section, we discuss the diacritization guidelines we used, as well as the setup for initial automatic processing followed by manual correction.

5.1 Diacritization Guidelines

We follow the Arabic maximal diacritization guidelines as presented in Elgamal et al. (2024) with a small number of modifications to fit the purpose of our task. We list the most important decisions that are different from standard Arabic diacritization.

The Lemmatization Requirement This effort focuses exclusively on the diacritization of proper nouns and mapping them to their lemmas. As such, we require the removal of clitics such as the definite article and the removal of plural suffixes (see Section 2.2).

Input Spelling Integrity Aside from the minimal changes connected to lemmatization, and cor-

Invalid Lemma		Gloss	Issue	Corrected Lemma	
سانشِيز	*sAnšiyz*	Sanchez	Long vowels require preceding diacritics	سَانْشِيز	*saAn.šiyz*
كَرَمُ	*karamu*	Karam	Final letter cannot have a diacritic	كَرَم	*karam*
عَضُّوم	*ςaDu∼wm*	Addoum	Short diacritic cannot precede Shadda	عَضُّوم	*ςaD∼uwm*

Table 6: Examples of malformed words and their corrected lemmas with transliterations.

rections of the obligatory Hamza diacritic in Alif Hamza forms (see Section 2.1), we do not add, remove, or modify any letters in the provided input.

Consonant Clusters in Foreign Names While standard Arabic generally avoids consonant clusters, our dataset includes many foreign proper nouns where such clusters are phonetically natural. To more faithfully capture their pronunciation, we allowed forms with consecutive consonants, either multiple letters marked with Sukuns, or a Sukun followed by a letter with Shadda (geminated), even though this departs from Standard Arabic diacritization norms. For example, إلكتريك *Ǎlktryk* 'Electric' should be diacritized as إِلِكْتْرِيك *Ǎilik.t.riyk* (with the consonant cluster /tr/), and زدينيك *zdynyk* 'Zdeněk' should be diacrtized as زْدِنِيك *z.diniyk* with initial /zd/ cluster.

Final Letter Ya The final letter ي *y* has multiple diacritizations that overlap with changes in dialectal Arabic, i.e. the softening of final y-gemination into /i/. As such, we had to dedicate part of the guidelines to outline the rules for diacritizing it as a geminated /yy/, a long vowel /i:/ or a glide /ay/.

The geminated version is the most specific in requirements with three possible cases:

- The gemination comes from the root or pattern of the word such as the final Ya in رَخِيّ *raxiy∼* 'Ar-Rakhi'.
- The lemma can be interpreted as having the derivational attribution suffix Ya-Nisba, e.g., إِشْبِيلِيّ *Ǎiš.biyliy∼* 'Sevillian' (of or related to إِشْبِيلِيَّة *Ǎiš.biyliy∼aħ* 'Seville').
- Gemination is necessary to reflect the pronunciation of certain foreign names, such as أَرْكْوِيّ *Âar.k.wiy∼* 'Arcueil'.

For other cases, if the final vowel sounds like a short /i/ or a long /i:/ and has a corresponding ي *y*, it is diacritized resembling a long vowel, e.g., أغاسي *ÂgAsy* Agassi, should be diacritized as أَغَاسِي *ÂagaAsiy*. The glide version is straightforward as it has a distinct phonological signal. One example is the word نَيْ *nay* 'Ney'.

Checking Well-formedness To ensure consistency with our annotation guidelines, we implemented automated checks to validate the well-formedness of diacritized lemmas. While these checks do not guarantee correctness, they are effective at identifying common errors and inconsistencies. We use these checks on both human and automatic annotations. See Table 6 for examples.

5.2 Initial Automatic Diacritization

To speed up the annotation process, we gave our annotator an automatically diacirtized version of the data. We used GPT-4o with Arabic Input and English Gloss (comparable to the best setting in Section 6). At the time of generating the initial automatic diacritization, we considered this a reasonable starting point.

GPT-4o postprocessing The output of GPT-4o was not always usable as is. When applying well-formedness checks to the diacritized outputs generated by GPT-4o, we observed several recurring patterns of errors that compromised the validity of the diacritized forms. In response, we developed an automated pipeline specifically aimed at correcting these systematic errors.[1] The automatic correction procedures included the following operations:

- Insertion of Fatha before Alif (ا *A*).
- Insertion of Kasra after Alif-Hamza-Below (إ *Ǎ*).
- Normalization of Shadda-Vowel clusters such that the vowel diacritic follows the Shadda diacritic.
- Removal of final diacritics as lemmas do not have them.

Type of Disagreement	Freq	Gloss	First Annotator	Second Annotator
Kasra ↔ Sukun	13	Tibet	تِبِت tibit	تِبْت tib.t
Kasra ↔ Fatha	9	Shechem	شِكِيم šikiym	شَكِيم šakiym
Consonant ↔ Long vowel	9	Jane	جَيْن jay.n	جِين jiyn
Sukun ↔ Damma	5	Acquaviva	أَكُوَافِيفَا ÂakuwaAfiyfaA	أَكْوَافِيفَا Âak.waAfiyfaA
Sukun ↔ Fatha	1	Aminadav	عَمِينْدَاف ςamiyn.daAf	عَمِينَدَاف ςamiynadaAf
Shadda ↔ ϕ	1	Oss	أُوسّ Âuws∼	أُوس Âuws

Table 7: Types of disagreements in Inter-Annotator Evaluation

- Insertion of missing Sukuns to indicate vowel absence at the end of syllable or in a consoant cluster.
- Removal of Fatha after Alif Madda (آ $\bar{A}$).
- Mapping Non-Arabic Arabic-script letters, such as those used in Urdu or Persian, to their closest Arabic language form.

5.3 Manual Diacritization

The manual diacritization and quality checks were carried out by a native speaker of Arabic from Egypt who is a trained linguist and a highly experienced annotator. The annotation process initially was done in tandem with the finalization of the guidelines with a team of the authors working jointly to optimize the quality of the annotation. The annotator was provided an Arabic word, along with its English gloss, and a proposed diacritization from GPT-4o after being refined by the automatic post-process described above. The annotations were carried on Google Sheets in a very simple setup. The annotator reviewed the proposed diacritization making changes where needed in accordance to the guidelines. The annotator made changes to 909 proposed lemmas out of 3,362 ($\sim$27%). In 213 instances (6.3% of all entries), there was a change connected with lemmatization: 74% relative involved the Al determiner, 22.5% a change in Alif-Hamza spelling, and 3.3% involving the demonym plural ending.

5.4 Inter-annotator Agreement

To assess the quality of our annotation and the consistency of our guidelines, we conducted an inter-annotator agreement study. A second annotator, a native Arabic speaker from Egypt, independently re-annotated a subset of 500 randomly selected samples from the dataset, utilizing the same annotation process and adhering to the same guidelines as the first annotator. Out of the 500 samples, the annotators fully agreed on 462 instances and disagreed on 38, resulting in an inter-annotator agreement rate of 92.4%. Table 7 presents the various types of inter-annotator disagreements along with their corresponding frequencies. Each row in the table represents a type of disagreement where the annotators selected different diacritics for the same word. For example, the first row shows instances where either one of the annotators chose a Kasra while the other selected a Sukun.

6 Evaluation

6.1 Experimental Setup

We perform computational experiments to perform the task of diacritization of proper nouns. For this, we prompt GPT-4o on all of the annotated dataset described in Section 5. We prompt the model with different input formats to assess its capabilities while giving it different levels of information: the inputs and the number of examples shown to the model (shots). We used default settings for optional parameters (e.g., temperature, top_p) from the gpt-4o-2024-11-20 snapshot.[4]

Inputs The model is given a detailed description of the task to be performed. Our main experiments reflect all the information given to our annotator, where we provide the model with both the Arabic Input and the English Gloss (**Arabic + Gloss**). Additionally, we also experiment with a more constrained setup where the model is provided solely with the the Arabic Input (**Arabic Only**).

[4] https://platform.openai.com/docs/api-reference/chat/create

Input Format	Shots	Accuracy	Distance
Arabic + Gloss	Zero	46.5%	1.02
Arabic + Gloss	One	61.9%	0.64
Arabic + Gloss	Few	**73.0%**	0.41
Arabic Only	Zero	36.7%	1.29
Arabic Only	One	49.7%	0.86
Arabic Only	Few	55.9%	0.71

Table 8: GPT-4o model results on CP-WIKI-D3K in terms of exact match accuracy and Levenshtein edit distance.

Shots In addition to the different inputs, we also consider further experiments where we supply the model with varying number of examples to learn from[1] Hence, in addition to just providing the input to diacritize (**Zero-Shot**), we also supply the model with a single example (**One-Shot**), and 80 examples (**Few-Shot**). The examples are randomly sampled from the CP-SAMA data. The one-shot and few-shot examples were selected once and reused across all model prompts. However, since CP-SAMA has fully lemmatized Arabic Inputs, we manually manipulated some of the examples to have a representation of clitic removal and Hamza normalization. Refer to Appendix A for a more detailed description of the prompts used.

Post-processing As a post-processing step, the outputs were ran through the same processing pipeline mentioned in Section 5.2. To evaluate the performance of the different experiments, we computed two metrics: accuracy by measuring the exact match between the post-processed output and the gold-standard diacritization and Levenshtein edit distance (Levenshtein, 1966) between the output and gold-standard diacritization.

6.2 Results

The results demonstrate that while diacritizing proper nouns remains a challenging task, incorporating the English gloss offers a valuable signal for the model. Notably, the best performance is achieved with few-shot, showing the effectiveness of providing a diverse and representative sample. Table 8 shows the results with different prompts.

6.3 Interplay of Frequency, Similarity, and Accuracy

We investigated how lexical frequency and phonological similarity (Freeman et al., 2006) affect model performance under our best configuration: few-shot prompting with Arabic + Gloss.

The Freeman similarity score averaged a high 91% across the dataset, consistent with the transliteration focus of the task. We binned the data into 10 intervals based on Freeman score. The lowest-similarity bins (up to 50%), comprising only 3% of the data, contained mostly high-frequency named entities and translations, e.g., مصر *mSr* for ‘Egypt’ and عمالاق *ςmlAq* for ‘jötnar’. Despite their low similarity, this group achieved 13.9% higher accuracy and had, on average, 10 times the frequency compared to the rest of the data. The bins up to 90% similarity comprised 35% of the data; their average frequency is only 5% higher than the last bin, but their average accuracy is lower by 3.6% absolute.

We found strong negative correlation between accuracy and edit distance (-0.95), confirming that higher accuracy aligns with fewer character edits. Frequency and Freeman score showed a moderate negative correlation (-0.69), likely due to high-frequency translated names. Freeman similarity and accuracy were also moderately negatively correlated (-0.70), indicating that frequent but phonetically dissimilar words are still predicted accurately.

We analyzed performance across frequency quartiles (Q1 to Q4). Accuracy rose steadily from 65% in Q1 to 80% in Q4. The correlation between average frequency and accuracy across quartiles was 0.68, confirming the positive impact of frequency on model performance. Full analysis tables are presented in Appendix B.

6.4 Error Analysis

We analyzed errors from a randomly selected sample of 1,010 output entries from the best performing setup from Section 6.2, and classified errors into several categories based on observed patterns. There were 740 (73.3%) exact matches (correct generations).

Of the 270 (26.7%) errors, there were 175 cases where the error was only diacritization differences. See examples in Table 9. Upon further analysis of this class of errors, we found that the model overpredicts Fathas (+25%) and Shaddas (+96%), while underpredicting Kasras (-18%) and Sukuns (-23%), indicating imbalanced vowel modeling and overuse of gemination.

The next largest class of errors, 60 cases, were those with spelling changes limited to the set of

Input	Gloss	Reference	Prediction	Error Type
العمود Alςmwd	Al-Amud	عَمُود ςamuwd	عَمُود ςamuwd	Exact Match
أفراموفو ÂfrAmwfw	Avramovo	أَفْرَامُوفُو Âaf.raAmuwfuw	أَفْرَامُوفُو Âaf.raAmuwfuw	Exact Match
هاغن hAγn	Hagen	هَاغِن haAγin	هَاغِن haAγin	Exact Match
إشتهارد ĂšthArd	Eshtehard	إِشْتِهَارْد Ăiš.tihaAr.d	إِشْتَهَارْد Ăiš.tahaAr.d	Diac
بلاجيفيتش blAjyfytš	Blažević	بْلَاجِيفِيتْش b.laAjiyfiyt.š	بِلَاجِيفِيتْش bilaAjiyfiyt.š	Diac
دسوق dswq	Desouk	دِسُوق disuwq	دُسُوق dusuwq	Diac
ريبلاي ryblAy	Ripley	رِيبْلَاي riyb.laAy	رِيبْلِي riyb.liy	*AWY*
ريكسينغين ryksynγyn	Rexingen	رِيكْسِينْغِين riyk.siyn.γiyn	رِيكْسِنْغِن riyk.sin.γin	*AWY*
جوندريزيك jwndryzyk	Gondrezick	جُونْدْرِيزِيك juwn.d.riyziyk	جُنْدْرِيزِيك jun.d.riyziyk	*AWY*
ميشيغان myšyγAn	Michigan	مِيشِيغَان miyšiyγaAn	مِيشِيجَان miyšiyjaAn	$j \leftrightarrow \gamma$
تسيخانوف tsyxAnwf	Ciechanów	تِسِيخَانُوف tisiyxaAnuwf	تِسِيهَانُوف tisiyhaAnuwf	$h \leftrightarrow x$
أردينة Ârdynħ	Ardineh	أَرْدِينَة Âar.diynaħ	أَرْدِينَه Âar.diynah	$\hbar \leftrightarrow h$
إيغيل Ăyγyl	Eagle	إِيغِيل Ăiyγiyl	إِيجِل Ăiyjil	Multiple
كرامة krAmħ	Gourrama	كُرَامَة kuraAmaħ	كُورَامَا kuwraAmaA	Multiple
ايكوميديا AykwmydyA	Eco-Médias	إِيكُومِيدْيَا Ăiykuwmiyd.yaA	إِيكُومِيدْيَآس Ăiykuwmiyd.yaĀs	Multiple
بارافرانكا bArAfrAnkA	Barrafranca	بَارَّافْرَانْكَا baAr aAf.raAn.kaA	بَارَافْرَانْكَة baAraAf.raAn.kaħ	Multiple

Table 9: Examples of evaluated instances along with their, reference and predicted diacritized forms, and corresponding error types. The error categories are diacritic mismatches (Diac), *AWY* spelling changes (*AWY*), several consonant and ta-marbuta substitutions ($j \leftrightarrow \gamma$, $h \leftrightarrow x$, and $\hbar \leftrightarrow h$), and those with multiple changes (Multiple).

long vowel (and glides) letters ا *A*, و *w*, and ي *y* (*AWY*). As we see in the examples in Table 9, the model has the tendency of dropping such letters rather than adding them. Another class of errors, 10 cases, are those with specific letter replacements such as ج $j \leftrightarrow$ غ γ, خ $x \leftrightarrow$ ه h, and ة $\hbar \leftrightarrow$ ه h. The final class of errors, 25 cases, are those with multiple changes happening at once.

While these cases don't match the gold reference, they are plausible and acceptable alternatives in most cases, especially in the context of linguistic variation discussed in Section 2. For example, the generated diacritization for بلاجيفيتش *blAjyfytš* 'Blažević' as seen in Table 9 (row 5), follows the common phenomena of breaking word initial complex onsets in many spoken dialects of Arabic and in MSA. Another example is the entry ايكوميديا *AykwmydyA* 'Eco-Médias', where the input follows a pronunciation-based transliteration while the generated form adhered to the orthography of the gloss.

These variations highlight the need for modeling techniques and evaluation metrics that account for this aspect of Arabic proper noun diacritization, which in turn requires additional annotated data.

7 Conclusion and Future Work

We presented a new 3,362 entry dataset of Arabic Wikipedia proper nouns annotated with gold-standard lemma diacritizations, paired with their English equivalents. This resource enables the joint study of diacritization and transliteration in a realistic setting characterized by ambiguity and spelling variation. We benchmarked GPT-4o on this task, providing insights into its capabilities and limitations. While the model performs reasonably well, especially on frequent names, it struggles with rarer entries and variant mappings.

Looking ahead, we plan to expand the dataset with more diverse names, integrate it into a morphological analyzer, and explore fine-tuned models for diacritizing proper nouns in broader contexts. We also plan to fine-tune dedicated models for this task and develop more robust approaches to name ambiguity, especially with multiple valid diacritizations. We hope this resource advances Arabic NLP and name normalization in multilingual settings like Wikipedia.

Limitations

A primary limitation of this work lies in the inherent subjectivity of diacritization, particularly for proper nouns where multiple correct variants may exist depending on regional, historical, or phonetic conventions. Despite rigorous annotation guidelines and quality checks, variability is an inevitable aspect of any human-annotated linguistic resource. Our current benchmark relies solely on GPT-4o, and we acknowledge the importance of evaluating performance across a broader range of large language models. While initial results are promising, the overall performance remains limited and, in our assessment, not yet suitable for reliable downstream use.

Ethics Statement

All data used in this project were sourced from publicly available Arabic Wikipedia entries and their corresponding English titles, in accordance with Wikimedia's terms of use. The annotation process was conducted transparently and ethically, with fair compensation provided to the annotators. We make both the corpus and the annotation guidelines publicly accessible under an open license, supporting reproducibility and community collaboration. Our goal is to contribute a valuable resource for Arabic language processing and to aid the broader Wikimedia effort by enhancing the quality of Arabic Wikipedia entries. Finally, we acknowledge that all NLP tools and resources can be used with malicious intent; this is not our intention, and we categorically discourage it.

Benefits

This work directly supports the Wikimedia community by enhancing the quality and accessibility of Arabic Wikipedia content. By providing more accurate diacritization for proper nouns from all over the world on Arabic Wikipedia, we aim to improve readability, pronunciation, and downstream tasks such as named entity recognition and machine translation. The dataset, code, and annotation guidelines are all released under the Creative Commons Attribution-ShareAlike (CC BY-SA) license to ensure community reuse and adaptation. Filtering was applied to select single-word proper nouns related to people, locations, and organizations, drawn from Arabic Wikipedia entries that have clear English counterparts, thereby supporting multilingual alignment and cross-lingual research.

Risks

Our project poses no known risks to Wikimedia editors or contributors. We do not name, identify, or reference any individual editor (by username or otherwise), nor do we expose any metadata that could be used to infer editor identities. The work focuses solely on content-level linguistic annotation and transformation. There are no known ways in which this research could be used to derive sensitive or personal information about contributors, and we strongly discourage any attempts to repurpose the resource for such purposes.

Acknowledgments

We would like to express our sincere gratitude to Hamdy Mubarak for his valuable insights and his generous willingness to answer our questions throughout the course of this work. We thank Djellel Difallah for advice on initial data collection. We would like to also thank Bashar Alhafni and Mostafa Saeed for their insightful discussions and helpful conversations.

References

Gheith A. Abandah, Alex Graves, Balkees Al-Shagoor, Alaa Arabiyat, Fuad Jamour, and Majid Al-Taee. 2015. Automatic diacritization of Arabic text using recurrent neural networks. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(2):183–197.

Nasreen Abduljaleel and Leah Larkey. 2004. English to Arabic transliteration for information retrieval: A statistical approach.

Sawsan Alqahtani, Ajay Mishra, and Mona Diab. 2019. Efficient convolutional neural networks for diacritic restoration. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1442–1448, Hong Kong, China. Association for Computational Linguistics.

Sawsan Alqahtani, Ajay Mishra, and Mona Diab. 2020. A multitask learning approach for diacritic restoration. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8238–8247, Online. Association for Computational Linguistics.

Yowell Aziz. 1983. Transliteration of English proper nouns into Arabic. *Meta*, 28(1):70–84.

Kenneth R. Beesley. 1997. Romanization, Transcription and Transliteration. Http://www.xrce.xerox.com/Research-Development/Historical-projects/Linguistic-Demos/Arabic-Morphological-Analysis-and-

Generation/Romanization-Transcription-and-Transliteration.

Fernando Benites, Gilbert François Duivesteijn, Pius von Däniken, and Mark Cieliebak. 2020. TRANSLIT: A large-scale name transliteration resource. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3265–3271, Marseille, France. European Language Resources Association.

Nancy Chen, Rafael E. Banchs, Min Zhang, Xiangyu Duan, and Haizhou Li. 2018. Report of NEWS 2018 named entity transliteration shared task. In *Proceedings of the Seventh Named Entities Workshop*, pages 55–73, Melbourne, Australia. Association for Computational Linguistics.

Peter T Daniels. 2013. The Arabic writing system. *The Oxford handbook of Arabic linguistics*, pages 422–431.

Kareem Darwish, Hamdy Mubarak, and Ahmed Abdelali. 2017. Arabic diacritization: Stats, rules, and hacks. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 9–17, Valencia, Spain. Association for Computational Linguistics.

Salman Elgamal, Ossama Obeid, Mhd Kabbani, Go Inoue, and Nizar Habash. 2024. Arabic diacritics in the wild: Exploiting opportunities for improved diacritization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14815–14829, Bangkok, Thailand. Association for Computational Linguistics.

Fadhl Eryani and Nizar Habash. 2021. Automatic Romanization of Arabic bibliographic records. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 213–218, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Andrew Freeman, Sherri Condon, and Christopher Ackerman. 2006. Cross linguistic name matching in English and Arabic. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 471–478, New York City, NY.

David Graff, Mohamed Maamouri, Basma Bouziri, Sondos Krouna, Seth Kulick, and Tim Buckwalter. 2009. Standard Arabic Morphological Analyzer (SAMA) Version 3.1. Linguistic Data Consortium LDC2009E73.

Nizar Habash and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 573–580, Ann Arbor, Michigan. Association for Computational Linguistics.

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.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*, volume 3. Morgan & Claypool Publishers.

Jack Halpern et al. 2009. Lexicon-driven approach to the recognition of Arabic named entities. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, pages 193–198. Citeseer.

Mustafa Jarrar, Diyam Akra, and Tymaa Hammouda. 2024. Alma: Fast lemmatizer and pos tagger for Arabic. *Procedia Computer Science*, 244:378–387. 6th International Conference on AI in 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 .

V. I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.

David Matthews. 2007. Transliteration using statistical machine translation. Master's thesis, University of Edinburgh, Edinburgh, United Kingdom. An automatic transliteration system built using Moses, modeled at the surface level with phrase-based SMT techniques.

Hamdy Mubarak, Ahmed Abdelali, Hassan Sajjad, Younes Samih, and Kareem Darwish. 2019. Highly effective Arabic diacritization using sequence to sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2390–2395, Minneapolis, Minnesota. Association for Computational Linguistics.

Hamdy Mubarak, Mohamed Al Sharqawy, and Esraa Al Masry. 2009. Diacritization and transliteration of proper nouns from Arabic to English. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt. The MEDAR Consortium.

Ossama Obeid, Go Inoue, and Nizar Habash. 2022. Camelira: An Arabic multi-dialect morphological disambiguator. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 319–326, Abu Dhabi, UAE. 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.

Edward Y. Odisho. 1992. Transliterating English in Arabic. *Zeitschrift für Arabische Linguistik*, 1(24):21–34.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, et al. 2024. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. 2008. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, Columbus, Ohio.

Yuanhe Tian, Renze Lou, Xiangyu Pang, Lianxi Wang, Shengyi Jiang, and Yan Song. 2022. Improving English-Arabic transliteration with phonemic memories. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3262–3272, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Imed Zitouni, Jeffrey S. Sorensen, and Ruhi Sarikaya. 2006. Maximum entropy based restoration of Arabic diacritics. In *Proceedings of the International Conference on Computational Linguistics and the Conference of the Association for Computational Linguistics (COLING-ACL)*, pages 577–584, Sydney, Australia.

A GPT-4o Prompts

In the system role, we provide the task description, and optionally, the few-shot demonstrations, when they are used. For the user role, we always provide the single instance to be diacritized. Table 10 lists all of the prompts used for the different settings. Table 11 shows a sample of the few-shot examples. These are formatted as a markdown table in the prompts.

Shots	Prompt
	Arabic Word+Gloss Input
Zero	You are an expert in Arabic. You are given the undiacritized proper noun in Arabic and its English gloss. Your task is to generate the corresponding diacritized proper noun lemma in Arabic. Arabic lemmas are dictionary entries that have no attached definite article (ال). Diacritization is adding the correct diacritic markings to undiacritized words. Remove the Arabic definite article (ال) when present. Do not add, remove, or substitute any other letters in the input. Determine the most accurate diacritization that matches the English gloss pronunciation. The user will provide a Markdown table with 1 rows. Each row contains an undiacritized proper noun in Arabic in the "Input" column and its English gloss in the "Gloss" column. Return exactly 1 diacritized lemmas, one per line. Do not include extra text, explanations, or formatting.
Few/One	You are an expert in Arabic. You are given the undiacritized proper noun in Arabic and its English gloss. Your task is to generate the corresponding diacritized proper noun lemma in Arabic. Arabic lemmas are dictionary entries that have no attached definite article (ال). Diacritization is adding the correct diacritic markings to undiacritized words. Remove the Arabic definite article (ال) when present. Do not add, remove, or substitute any other letters in the input. Determine the most accurate diacritization that matches the English gloss pronunciation. The user will provide a Markdown table with 1 rows. Each row contains an undiacritized proper noun in Arabic in the "Input" column and its English gloss in the "Gloss" column. Return exactly 1 diacritized lemmas, one per line. Do not include extra text, explanations, or formatting. Here are some examples of triplets of an undiacritized proper noun in Arabic ("Input"), its respective English gloss ("Gloss"), and its diacritized lemma ("Output") for reference `<Few-Shots-table>`
	Arabic Word Only Input
Zero	You are an expert in Arabic. You are given the undiacritized proper noun in Arabic. Your task is to generate the corresponding diacritized proper noun lemma in Arabic. Arabic lemmas are dictionary entries that have no attached definite article (ال). Diacritization is adding the correct diacritic markings to undiacritized words. Remove the Arabic definite article (ال) when present. Do not add, remove, or substitute any other letters in the input. The user will provide a Markdown table with 1 rows. Each row contains an undiacritized proper noun in Arabic in the "Input" column. Return exactly 1 diacritized lemmas, one per line. Do not include extra text, explanations, or formatting.
Few/One	You are an expert in Arabic. You are given the undiacritized proper noun in Arabic. Your task is to generate the corresponding diacritized proper noun lemma in Arabic. Arabic lemmas are dictionary entries that have no attached definite article (ال). Diacritization is adding the correct diacritic markings to undiacritized words. Remove the Arabic definite article (ال) when present. Do not add, remove, or substitute any other letters in the input. The user will provide a Markdown table with 1 rows. Each row contains an undiacritized proper noun in Arabic in the "Input" column. Return exactly 1 diacritized lemmas, one per line. Do not include extra text, explanations, or formatting. Here are some examples of pairs of an undiacritized proper noun in Arabic ("Input"), and its diacritized lemma ("Output") for reference `<Few-Shots-table>`

Table 10: System prompts used in the experiments. `<Few-Shots-table>` is a placeholder for few-shot examples. In either setting, the user prompts consist solely of a single instance to be diacritized.

Arabic Word		Gloss	Diacritized Reference	
ايدكس	*Aydks*	IDEX	إِيدِكس	*Ǎiydiks*
الغارديان	*AlgArdyAn*	Guardian	غَارْدِيَان	*gaAr.diyaAn*
رودريغيز	*rwdrygyz*	Rodriguez	رُودْرِيغِيز	*ruwd.riygiyz*
اوروغواي	*AwrwgwAy*	Uruguay	أُورُوغْوَاي	*Âuwruwg.waAy*
بوتيه	*bwtyh*	Boutier	بُوتِيِه	*buwtiyih*
وايزمن	*wAyzmn*	Weizman	وَايزْمَن	*waAyz.man*

Table 11: A sample of few-shot examples used for prompting GPT-4o

B Supplementary Interplay of Frequency, Similarity, and Accuracy

Freeman Bin	Instances	Instance %	Frequency	Matches	Accuracy	Distance
10%	6	0.2%	2,280,059	5	83.3%	0.17
20%	7	0.2%	454,346	6	85.7%	0.29
30%	23	0.7%	303,728	20	87.0%	0.22
40%	27	0.8%	690,729	23	85.2%	0.26
50%	26	0.8%	64,814	23	88.5%	0.12
60%	71	2.1%	30,274	45	63.4%	0.69
70%	164	4.9%	57,361	124	75.6%	0.37
80%	271	8.1%	22,803	185	68.3%	0.46
90%	587	17.5%	22,909	404	68.8%	0.52
100%	2,180	64.8%	60,343	1,619	74.3%	0.38
10–90%	1,182	35.2%	63,761	835	70.6%	0.48
10–50%	89	2.6%	496,420	77	86.5%	0.20
60–100%	3,273	97.4%	49,719	2,377	72.6%	0.42
All	3,362	100.0%	61,544	2,454	73.0%	0.41

Table 12: Accuracy, average frequency, and edit distance across Freeman similarity score bins.

Frequency Range	Instances	Average Freq.	Matches	Accuracy	Avg. Freeman
Q1 (lowest 25%)	787	2	510	64.8%	91.1%
Q2 (25–50%)	893	25	627	70.2%	90.4%
Q3 (50–75%)	840	567	646	76.9%	91.2%
Q4 (highest 25%)	842	245,145	671	79.7%	89.7%
All	3,362	61,544	2,454	72.99%	90.6%

Table 13: Accuracy, Average Frequency, and average Freeman similarity scores across word frequency quartiles.