Computational Morphology and Lexicography Modeling of Modern Standard Arabic Nominals

Christian Khairallah,[†] **Reham Marzouk,**[†,††] **Salam Khalifa,**[†,‡]
Mayar Nassar,[†,‡‡] **Nizar Habash**[†]
Computational Approaches to Modeling Language (CAMeL) Lab
[†]New York University Abu Dhabi, [††]Alexandria University
[‡]Stony Brook University, [‡‡]Ain Shams University
{christian.khairallah,nizar.habash}@nyu.edu, igsr.r.marzouk@alexu.edu.eg,
salam.khalifa@stonybrook.edu, mayar.nassar@art.asu.edu.eg

Abstract

Modern Standard Arabic (MSA) nominals present many morphological and lexical modeling challenges that have not been consistently addressed previously. This paper attempts to define the space of such challenges, and leverage a recently proposed morphological framework to build a comprehensive and extensible model for MSA nominals. Our model design addresses the nominals' intricate morphotactics, as well as their paradigmatic irregularities. Our implementation showcases enhanced accuracy and consistency compared to a commonly used MSA morphological analyzer and generator. We make our models publicly available.

1 Introduction

Arabic poses many challenges to computational morphology: its hybrid templatic and concatenative processes, rich collections of inflectional and cliticization features, numerous allomorphs, and highly ambiguous orthography. Over the decades, many approaches have been explored in developing Arabic morphological analyzers and generators (Beesley et al., 1989; Kiraz, 1994; Buckwalter, 2004; Graff et al., 2009; Habash et al., 2022). These tools continue to show value for Arabic natural language processing (NLP) even when paired with state-of-the-art neural models on various tasks such as morphological tagging (Zalmout and Habash, 2017; Inoue et al., 2022), sentiment analysis (Baly et al., 2017), and controlled text rewriting (Alhafni et al., 2022). Developing such tools is neither cheap nor easy; and some of them are not freely available, or incomplete, e.g., Habash et al. (2022) points out how a popular Arabic analyzer, SAMA (Graff et al., 2009), has very low coverage for phenomena such as command form or passive voice.

The effort presented in this paper is about the modeling of Modern Standard Arabic (MSA) nominals in an open-source Arabic morphology project (CAMELMORPH) introduced by Habash et al. (2022), who demonstrated their approach on verbs in MSA and Egyptian Arabic. Verbs are generally seen as the *sweethearts* of Arabic computational morphology: while they have some complexity, they are very regular and predictable. Nominals are far more complex — in addition to their numerous morphotactics, they have complicated paradigms with different degrees of completeness and many irregular forms, e.g., broken plural and irregular feminines (Alkuhlani and Habash, 2011).

Our contributions are (a) **defining** the space of challenges in modeling MSA nominals (*nouns*, *adjectives*, and *elatives/comparative adjectives*); (b) **developing** a large-scale implementation which is easily extendable within the recently introduced CAMELMORPH framework; (c) **benchmarking** our models against a popular Arabic morphology database (Graff et al., 2009; Taji et al., 2018) and demonstrating them to be more accurate and consistent; and finally (d) making our databases and code **publicly available**.[1]

Next, we present relevant terminology (§2), and related work (§3). We follow with a discussion of Arabic nominal modeling challenges (§4), and give an overview on the CAMELMORPH framework (§5) and how we utilize it (§6). Finally, we present an evaluation of our system (§7).

2 Relevant Terminology

We present the relevant terminology we use in this paper and illustrate it with examples in Table 1. The table presents four different ways to represent the morphological information. Arabic words are created by combining different types of **morphemes**: some are concatenative **affixes** (*nominals only take* ***suffixes***) and **clitics**, and others are templatic **roots** and **patterns** that interdigitate to form **stems**, which concatenate with the suffixes and

[1]All system details and guidelines are available under the official_releases/eacl2024_release/ directory of the project GitHub: http://morph.camel-lab.com.

Findings of the Association for Computational Linguistics: EACL 2024, pages 1071–1084
March 17-22, 2024 ©2024 Association for Computational Linguistics

<table>
<tr><td>Word</td><td colspan="11">(a) وَلِسَفِيرَاتِهِم walisafiyraAtihim `and for their ambassadors [f]'</td></tr>
<tr><td rowspan="3">Surface Segmentation</td><td colspan="3" rowspan="2">Proclitics</td><td colspan="7">Baseword</td><td rowspan="2">Enclitic</td></tr>
<tr><td colspan="3">Stem</td><td colspan="4">Suffixes</td></tr>
<tr><td>wa+</td><td>li+</td><td></td><td colspan="3">safiyr</td><td colspan="2">+aAt</td><td colspan="2">+i</td><td>+him</td></tr>
<tr><td rowspan="2">Morpheme & Features</td><td>prc2</td><td>prc1</td><td>prc0</td><td>lex</td><td>root</td><td>pattern</td><td>gen</td><td>num</td><td>cas</td><td>stt</td><td>enc0</td></tr>
<tr><td>wa+</td><td>li+</td><td>∅</td><td>safiyr</td><td>s.f.r</td><td>1a2iy3</td><td>f</td><td>p</td><td>g</td><td>c</td><td>+hum</td></tr>
<tr><td rowspan="2">Buckwalter Database</td><td colspan="3">DBPrefix</td><td colspan="3">DBStem</td><td colspan="5">DBSuffix</td></tr>
<tr><td colspan="3">wali+</td><td colspan="3">safiyr</td><td colspan="5">+aAtihim</td></tr>
<tr><td rowspan="2">Camel Morph Specs</td><td>[Conj]</td><td>[Prep]</td><td>[Art]</td><td colspan="2">[Stem]</td><td>[Buffer]</td><td colspan="4">[Suff]</td><td>[Pron]</td></tr>
<tr><td>wa+</td><td>li+</td><td>∅</td><td colspan="2">safiyr</td><td>∅</td><td colspan="2">+aAt</td><td colspan="2">+i</td><td>+him</td></tr>
</table>

<table>
<tr><td>Word</td><td colspan="11">(b) وَلِسُفَرَائِهِم walisufaraAŷihim `and for their ambassadors [m]'</td></tr>
<tr><td rowspan="3">Surface Segmentation</td><td colspan="3" rowspan="2">Proclitics</td><td colspan="7">Baseword</td><td rowspan="2">Enclitic</td></tr>
<tr><td colspan="5">Stem</td><td colspan="2">Suffixes</td></tr>
<tr><td>wa+</td><td>li+</td><td></td><td colspan="5">sufaraAŷ</td><td colspan="2">+i</td><td>+him</td></tr>
<tr><td rowspan="2">Morpheme & Features</td><td>prc2</td><td>prc1</td><td>prc0</td><td>lex</td><td>root</td><td>pattern</td><td>gen</td><td>num</td><td>cas</td><td>stt</td><td>enc0</td></tr>
<tr><td>wa+</td><td>li+</td><td>∅</td><td>safiyr</td><td>s.f.r</td><td>1u2a3aA'</td><td>m</td><td>p</td><td>g</td><td>c</td><td>+hum</td></tr>
<tr><td rowspan="2">Buckwalter Database</td><td colspan="3">DBPrefix</td><td colspan="5">DBStem</td><td colspan="3">DBSuffix</td></tr>
<tr><td colspan="3">wali+</td><td colspan="5">sufaraAŷ</td><td colspan="3">+ihim</td></tr>
<tr><td rowspan="2">Camel Morph Specs</td><td>[Conj]</td><td>[Prep]</td><td>[Art]</td><td colspan="2">[Stem]</td><td>[Buffer]</td><td colspan="4">[Suff]</td><td>[Pron]</td></tr>
<tr><td>wa</td><td>li</td><td>∅</td><td colspan="2">sufaraA</td><td>ŷ</td><td colspan="2">∅</td><td colspan="2">+i</td><td>+him</td></tr>
</table>

Table 1: Two examples in four different Arabic morphological representation schemes.

clitics. Nominal suffixes typically represent **gender**, **number**, **case** and **state** features. However, occasionally some of these features are realized through patterns, e.g., Table 1 (b)'s example of **templatic** (aka **broken**) **plural**. **Proclitics** (conjunctions, prepositions, and definite article) and **enclitics** (possessive pronouns) are syntactically independent but phono-orthographically dependent morphemes. We use the term **baseword** to refer to the most basic complete word form (stem+suffixes) without clitics. Some morphemes have contextually variable alternatives, called **allomorphs**, e.g., in Table 1, the enclitic +هُم +*hum*[2] has an allomorph +هِم +*him* which is used if an /i/ vowel precedes it. Systematic allomorphic changes in stem endings can be represented using stem sub-strings called **stem buffers** (Habash et al., 2022), e.g., Table 1 (b)'s **[Buffer]** in the Camel Morph Specs row has two other forms that may vary based on the vowel of the suffix that follows it: سُفَرَا(ء|ؤ|ئ) *sufaraA('|ŵ|ŷ)*.

At a higher level beyond a single word, and inspired by Stump (2001), we define the **lexeme** as the set of words varying through inflection and cliticization operations. The lexeme is headed by a representative form called the **lemma** (**lex** in Table 1). We refer to the **paradigm** as the space occupied by a lexeme over the inflectional grid, which is structured according to a set of morphosyntactic **functional features**. Different combinations of the **values** of these features define **paradigm slots**, and these slots are either occupied by one word form or more (e.g., words having two plural forms), or they may be empty. For an Arabic nominal, the obligatory features are **POS**, **case**, **state**, **gender**, and **number**, and optional ones come in the form of concatenative **clitics** (Habash, 2010). Hence, given a lemma and a set of feature values, one can generate all the word forms in a lexeme, i.e., **inflection**. Within this framework, any other (i.e., non-inflectional) morphological transformation maintaining the same templatic root of a lexeme results in a different lexeme, and this is called **derivation**.

Finally, Appendix A presents a glossary of the discussed terms, with their abbreviations,[3] Arabic equivalents, and examples.

3 Related Work

Morphological Analysis & Generation This work builds on a long history of morphological analysis and generation tools which may, or may not, have tried to extensively model Arabic nominals (Al-Sughaiyer and Al-Kharashi, 2004; Habash, 2007; Sawalha and Atwell, 2008; Habash, 2010; Altantawy et al., 2011). Altantawy et al. (2011) categorizes different approaches along a conituum based on their modeling of morphological representations of words. At one end, the representations are characterized by rich linguistic abstractions and a greater reliance on a templatic-affixational perspective of morphology (Beesley et al., 1989; Kiraz, 1994; Beesley, 1996; Habash and Rambow, 2006; Smrž, 2007a; Boudchiche et al., 2017); while at the other end, the representations tend to be more surface-form oriented and organized along precompiled derivation-inflectional solutions (Buckwalter, 2004; Graff et al., 2009; Taji et al., 2018). The former tends to rely on multi-tiered representations that map underlying forms to surface forms, generally using finite-state transducers through complex rules; and can either model at the morpheme (Beesley, 1996) or lexeme level (Smrž, 2007a). The

[2]HSB Arabic transliteration (Habash et al., 2007b).

[3]A quick reference to abbreviations: masculine, feminine, singular, and plural for functional gender-number; Masculine, Feminine, Singular, and Plural for form gender-number; accusative, nominative, and genitive for case; indefinite, definite, and construct for state; 1 and 3 for 1st and 3rd person; Noun (Rational or Irrational) or Adjective for POS.

latter tends to follow a more stem-based approach where morphotactic rules are built directly into the lexicon and inherently models at the morpheme and features level, without including roots and patterns into the rules. The most widely used of these models rely on the six-table approach used in the Buckwalter/Standard Arabic Morphological Analyzer (BAMA/SAMA) (Buckwalter, 2004; Graff et al., 2009), which entails a lexicon of morphemes and compatibility tables between them.

Aligned, to a degree, with the stem-based methodologies, Habash et al. (2022) presented a *middle ground* approach, within the open-source Arabic morphology project CAMELMORPH. They modeled morphotactic allomorphy via linguistically motivated inter-allomorphic compatibility rules, and facilitated the creation of lexicons (closed and open-class) that are comparatively easy to manipulate and modify. They demonstrated their approach building on top of, and comparing to, Buckwalter (2004)'s latest extension (Taji et al., 2018). They presented results on modeling the Arabic verbal system in MSA and Egyptian Arabic. In this paper, we leverage their approach to comprehensively model MSA nominals.

Computational Modeling of Arabic Nominals Modeling Arabic nominal morphology presents a more intricate challenge when compared to verbs, as the latter generally follow strictly regular inflectional patterns (Al-Sughaiyer and Al-Kharashi, 2004; Altantawy et al., 2010; Habash, 2010; Alkuhlani and Habash, 2011). Even when nominals are modeled, their treatment is often incomplete. For example, broken plurals are not always linked to their singular forms (or lemmas), which adds a cost to using them in downstream applications (Xu et al., 2002). Even in systems that modeled broken plurals lexically, e.g., Buckwalter (2004), there were major gaps such as not specifying their functional gender and number (Smrž, 2007b; Alkuhlani and Habash, 2011). Furthermore, Buckwalter (2004) confounded the definite and construct states for some morphemes (Smrž, 2007b).

Several attempts were undertaken to tackle these issues (Soudi et al., 2001; Smrž, 2007b; Habash et al., 2007a; Altantawy et al., 2010; Alkuhlani and Habash, 2011; Neme and Laporte, 2013; Taji et al., 2018); however, they either lacked a comprehensive approach, focused only on a subset of nominals, or proved challenging to extend straightforwardly.

		ni	**nd**	**nc**	**ai**	**ad**	**ac**	**gi**	**gd**	**gc**
MS	**ms**	◌ٌ ũ	◌ُ u		◌ًا Aã	◌َ a		◌ٍ ĩ	◌ِ i	
FS	**fs**	◌َةٌ aħũ	◌َةُ aħu		◌َةً aħã	◌َةَ aħa		◌َةٍ aħĩ	◌َةِ aħi	
MD	**md**	◌َانِ aAni		◌َا aA	◌َيْنِ ayni		◌َيْ ay	◌َيْنِ ayni		◌َيْ ay
FD	**fd**	◌َتَانِ ataAni		◌َتَا ataA	◌َتَيْنِ atayni		◌َتَيْ atay	◌َتَيْنِ atayni		◌َتَيْ atay
MP	**mp**	◌ُونَ uwna		◌ُو uw	◌ِينَ iyna		◌ِي iy	◌ِينَ iyna		◌ِي iy
FP	**fp**	◌َاتٌ aAtũ	◌َاتُ aAtu		◌َاتٍ aAtĩ	◌َاتِ aAti		◌َاتٍ aAtĩ	◌َاتِ aAti	

Table 2: The set of MSA nominal suffixes and their **default** mapping to functional values of gender-number (rows) and case-state (columns). The capitalized tags refer to the set of suffixes by form, not function. Trivially, they match here because this is a *default mapping table*. Merged cells indicate instances of syncretism in adjacent cells. Greyed cells indicate syncretism with non-adjacent cells. For example, in the last row, the feminine plural form *aAti* maps to four functional feature combinations: **fp(ad|ac|gd|gc)** – accusative/genitive and definite/construct.

4 Arabic Nominal Morphology

Default word composition assumes a straightforward one-to-one mapping from features to morphemes, with simple interdigitation and concatenation. In practice, however, there are many variations and exceptions. We outline the most important issues next, starting with word-level inflection and cliticization, and following with lexicographic and paradigmatic challenges.

4.1 Inflection and Cliticization Particularities

Default Nominal Suffixes The **default** Arabic nominal suffixes express combinations of four features: gender, number, case, and state. As Table 2 demonstrates, many of the unique 28 suffixes map to different subsets of the 54 possible feature combinations. Some of the suffixes can be decomposed into smaller compositional units, such as case and state endings with feminine and masculine singular, as well as feminine plural suffixes, but there are some inconsistencies such as the identical accusative and genitive suffixes for feminine plural. While there is a default functional meaning to these morphemes, we find many instances in which there are mismatches between their form and the functional feature values in the word, mostly in number and gender, but also in case and state. We will refer to the morpheme **forms** using a capitalization of their default **functional** feature values. For example, **FP** refers to the suffix set typically associated with the functional features **fp** without the requirement that the functional features be **fp**, e.g., امتحانات *AmtHAnAt* 'exams' where this is a functionally masculine plural (**mp**) noun which takes a feminine plural (**FP**) suffix (see last row in Table 2). Taking a **FP** suffix does not change its functional masculinity. In this case, the function of the **FP** suffix is not **fp**, its default, but another value (**mp**).[4]

Gender-Number Suffix Mismatch Some nominals have suffixes that, by default, express gender and number values that do not match those of the nominals themselves. Examples include خليفة *xaliyfaħ* 'Caliph' (**ms** noun, **FS** suffix), نار *nAr* 'fire' (**fs** noun, **MS** suffix), طلبة *Tlbħ* 'students' (**mp** noun, **FS** suffix), and نيران *nyrAn* 'fires' (**fp** noun, **MS** suffix).

Broken and Other Plurals A majority of gender-number suffix mismatches occur with **broken plurals**, nominals whose number is specified through templatic pattern change. Examples include حوامل *HwAml* 'pregnant [p]' (**fp** noun, **MS** suffix), كلاب *klAb* 'dogs' (**mp** noun, **MS** suffix), and طلبة *Tlbħ* 'students' (**mp** noun, **FS** suffix). In a minority of cases, there are sound plurals that require slight changes in the stems. An example of such **semi-sound plurals** is the noun حَفَلَات *HafalaAt* 'parties'(**fp**, **FP**), whose base stem would suggest the incorrect form حَفْلَات* *Haf.laAt*. Another case is **plurals of plurals**, nominals that use broken plural patterns with plural suffixes, e.g., رِجَالَات *rijaAlaAt* 'leading men'(**mp** broken plural stem, **FP** suffix).

Diptotes, Invariables, Indeclinables, and Defectives There are many classes of nominals with different variations in terms of how case and state features are realized (Buckley, 2004). In contrast to **triptotes** (the default nominals), **diptotes** (الممنوع من الصرف), identified typically by pattern or foreign origin, express exceptional syncretism in their case suffixes: *indefinite* diptotes use default definite suffixes, and they also use default accusative suffixes for both accusative and genitive case. When they are not indefinite, they use default suffixes normally. One example is the noun سُفَرَاءَ *sufaraA'+a* 'ambassadors' (**MSAD** suffix, but ambiguous **ai**, **gi**, **ad**, or **ac**).

Invariables use a zero suffix for all case and state features, e.g. دُنيا *dun.yA* 'world'. **Indeclinables** use the default accusative singular for all cases, e.g., فتىً *fatayã* 'young man'. And **Defectives** use the default genitive suffix for nominative in indefinite form, e.g., قَاضٍ *qaADĩ* 'judge' (**MSGI** suffix, but ambiguous **gi**, **ni**). In addition to the above, there are very special sets of nominals with unique behavior, such as the so-called *five nouns*, which exceptionally represent case in long vowels, e.g., أبو, أبا, أبي *Âbw*, *ÂbA*, *Âby* 'father *of* ...' (nominative, accusative, genitive, respectively). Finally, the **MS** suffix (اً *Aã*) is written without its *Alif* (long vowel [A]) when the stem ends with a *hamza* (glottal stop), e.g., هواءً *hwA'ã* 'air' as opposed to هواءاً* **hwA'Aã*).

Variable Stem Endings There are many nominal classes where the stem ending changes based on the presence of specific suffixes and clitics. The following are two of the most common classes. **Alif-hamza-final** nominal stems vary their *hamza* (glottal stop) form when followed by a clitic. The variation reflects orthographic harmony with the vowels that follow it, e.g., سُفَرَاءَهُ *sufaraA'**a**hu*, سُفَرَاؤُهُ *sufaraAŵ**u**hu*, سُفَرَائِهِ *sufaraAŷ**i**hi*, 'his ambassadors' in accusative, nominative and genitive, respectively. **Defective** nominal stems lose their final letter in some contexts, e.g., قَاضٍ *qaADĩ* and قَاضِياً *qaADiyAã*, 'a judge' in the nominative/genitive and accusative, respectively. For all such regular cases, we model the varying stem ending as part of the stem buffer.

Proclitics Most nominal proclitics do not vary in form when attached to basewords. One common exception is the Arabic determiner +ال *Al+*, whose

[4] Some readers may question the logic of the word امتحانات *AmtHAnAt* 'exams' being masculine since it requires a feminine number (3-10) quantifier and feminine singular adjective: خمسة امتحانات صعبة *xmsħ AmtHAnAt Sςbħ* 'five hard exams'. However, MSA agreement rules require reverse-gender agreement for number (3-10) quantifiers, and feminine singular adjective for irrational (non-human) plurals. Furthermore, the singular form امتحان *AmtHAn* 'exam' is masculine, and simply pluralizing a noun does not change its gender. For more details, see Alkuhlani and Habash (2011).

first letter elides after the prepositional proclitic لِ+ *li+* ‘for’. The presence of the determiner leads to the addition of a gemination diacritic on the first letter in the baseword if it is a coronal consonant, aka, *sun letter*, e.g., لِ+ ال+ شَمْسِ *li+Al+šam.si* realizes as لِلشَّمْسِ *lilš∼am.si* ‘for the sun’.

Enclitics Pronominal possessive enclitics tend to interact in different ways with stems and suffixes. Some examples were presented above under *Variable Stem Endings*. The following are other common cases of such interactions. The feminine singular suffix ة *ħ* changes to a ت *t* before a clitic, e.g., سَفِيرَةُ+نَا *safiyraħu+naA* realizes as سَفِيرَتُنَا *safiyratunaA* ‘our ambassador’. Similarly, the stem ending ى *ý* turns to ا *A* before a clitic, e.g., مَبْنَى+ي *mab.naý+iy* مَبْنَايَ *mab.naAya* ‘my building’. The 1st person singular pronominal clitic has three allomorphs, and each of the 3rd person pronominal clitics has two. Table 1(a) and (b) illustrate one case of the latter (*i+hum*→*i+him*).

4.2 Paradigmatic Variation

An important difference between modeling verbal and nominal morphology in Arabic is the consistent completeness of verbal paradigms (with very few exceptions), and the high degree of variability and incompleteness in nominals. While this issue does not affect the modeling of specific words, it matters for linking words in the same lexeme and for taming the lexicon. Table 3 presents examples of different nominal paradigms using a simplified four-slot format covering gender and number (columns) for different lexemes (rows). We omit the *dual* value due to its regularity, and case and state for simplicity. The slots (cells) specify the suffix morphemes using the default values discussed above.

Paradigm Completeness and Stem Count A simple standard paradigm uses one stem for all slots and default nominal suffix mapping (perfect match in form and function), e.g., Table 3 (1, 3). Some complete paradigms use multiple stems, typically to accommodate one or more broken plurals, e.g., Table 3 (2). Incomplete paradigms do not inflect for certain gender and/or number combinations, and some may use one or many stems, e.g., Table 3 (all except 1, 2, 3). Of course, some paradigms are complicated by function-form mismatches, e.g., Table 3 (6, 7, 9).

				Features			
	Lemma	**Gloss**	**Stem**	**ms**	**mp**	**fs**	**fp**
1	kAtib	writer/writing (A)	*kAtib*	+MS	+MP	+FS	+FP
2a	kAtib	writer/author (N:R)	*kAtib*	+MS		+FS	+FP
2b			*kut∼Ab*		+MS		
3	muxAbar	addressed (A)	*muxAbar*	+MS	+MP	+FS	+FP
4	muxAbaraħ	call (N:I)	*muxAbar*			+FS	+FP
5	muxAbarAt	intelligence (N:I)	*muxAbar*				+FP
6a	nAr	fire (N:I)	*nAr*			+MS	
6b			*niyrAn*				+MS
7a	xaliyfaħ	caliph (N:R)	*xaliyf*	+FS			
7b			*xulafA'*		+MS		
7c			*xalAŷif*		+MS		
8	lay.l	night (N:I)	*lay.l*	+MS			
9	nisA'	women (N:R)	*nisA'*				+MS
10a	tam.r	dates (N:I)	*tam.r*	+MS			
10b			*tumuwr*		+MS		
11a	tam.raħ	date (N:I)	*tam.r*			+FS	
11b			*tamar*				+FP

Table 3: Arabic nominal paradigm examples pairing *functional* feature values with *form* values. See footnote 3 for abbreviations.

Inter-paradigm Ambiguity Considering Table 3, some paradigm stems seem like they could neatly fit as a subset of a different paradigm, like in the case of Table 3 (3, 4, 5), (1 and 2a), and (10 and 11). However, because they share different meaning spaces and sometimes different POS, they belong to different lexemes. There is no denying the derivational relationship among these lexemes: they come from the same root and same initial pattern, but due to derivational specification, the meaning and the paradigm size are affected beyond simple semantic shift. For example, lemmas (3, 4, 5) in Table 3 go from a passive participle adjective (‘addressed/called’) to a specific common noun (‘a call’) to a more specific common noun that has no singular (‘intelligence services’). The lemma pairs (10 and 11) represent common derivational pairs of mass/collective nouns and instances of them. Given the high degree of variability and inconsistency due to derivational history, this aspect of morphology modeling is complex and demanding.

5 The CAMELMORPH Approach

The CAMELMORPH approach is based on a general framework that could, in principle, be used to build morphological analysis and generation models for any language with concatenative morphology and allomorphic variations (Habash et al., 2022). The CAMELMORPH approach requires designing **morphological specifications** describing the language's grammar and lexicon, which are then converted via an offline process powered by its **DB Maker** algorithm into a **morphological database** (DB) in the style of BAMA/SAMA DBs (Buckwalter, 2004; Graff et al., 2009; Taji et al., 2018). The created DBs can be used by any analysis and generation engine familiar with its format, such as Camel Tools (Obeid et al., 2020).

The CAMELMORPH morphological specifications can be divided into **Order** and **Morpheme** specs. The *order* specifies the positions of all *morpheme classes* in a word. The morpheme class consists of *allomorphs* organized into *morphemes*. These are divided into closed-class (suffixes and clitics), and open-class (stem lexicon) morphemes. Associated with each allomorph is a set of handcrafted **conditions**, which control allomorph selection for a specific morpheme. There are two types of conditions: **set conditions** are activated by the allomorph, and **required conditions** are needed by the allomorph. The lexicon is a large repository that contains the stems and their associated lemmas, and other features. Within this framework, the stems also *set* and *require* conditions just like the closed-class morphemes. The offline **DB Maker** process makes heavy use of these conditions to determine proper combinations and compatibility among the allomorphs in a word. Finally, the framework accommodates the use of ortho-phonological rewrite regex rules (such as sun-letter handling) as part of the analysis/generation engine.

6 Modeling Nominals in CAMELMORPH

Next, we discuss the morphological and lexicographic design decisions, which we used to solve all the challenges mentioned in Section 4, and more. The full guidelines will be publicly available (see foonote 1). The last subsection below presents statistics on the resulting database.

6.1 Morphotactic Modeling

Given the complexity of the full system, we employ a highly redacted example in Figure 1 to explain how the system behaves and cover the cases in Table 1 and a bit more.

Morph Order The top of Figure 1 shows a segment of the **Order** part of the *Morphology Specifications* for genitive suffixes. The order specifies the prepositional clitics that can occur with genitive suffixes, and the relative order of conjunctions, prepositions and determiner clitics (**DBPrefix**; see also Table 1). The stem part consists of a nominal stem and buffer, and the suffix part includes the pronoun enclitic only for the construct suffixes. The presence of a class in the order sequence does not necessarily mean a morpheme has to be present. Optional classes, such as determiner or pronoun allow a *nothing* option – see Figure 1 (P1,C1).

Lexicon and Buffers The **Lexicon** section shows a lemma with two stems, which together make up a paradigm with a broken plural. The base stem in Figure 1 (L1a) does not specify any feature values as it will acquire them from the suffixes. It lists three required conditions which correspond to the default **MS**, **FS** and **FP** (no **MP**), as defined in Section 4.1. The broken plural stem (L1b) specifies the gender and number features, which override any features from suffixes. It also indicates being an *Alif-hamza-final* (`#A'`) stem and a diptote (`#dip`) under **Set Conditions**, and requires the **MS** suffix only. The **Buffers** section provides the possible segments to complete the `#A'` stems under different required conditions.

Suffixes The suffixes provided in this redacted example are only for **MS** and **FP** (see Section 4.1). Here, we see how a diptote suffix behavior is modeled through the use of the `#dip` condition: the morpheme `Suff.MSIG` has two allomorphs, both of which set the condition `MS`, but one requires the condition `#dip`, and the other requires the negation of `#dip` `[else of #dip]`. Also, the construct suffixes that interact with pronouns set the condition `suff-i` indicating the presence of a final /i/.

Proclitics and Enclitics The determiner proclitic in this redacted example has no special constraints. However, in complete models, the determiner requires that sun letters that follow it take a *shadda* diacritic. Although this requirement is not covered in Figure 1, it is modeled in our full system with a regex rule in the analysis/generation engine. The pronoun enclitic, `Pron.3MP` shows two allomorphs that vary depending on the presence of a suffix /i/, which is set by some of the suffixes.

End-to-End Examples The right-hand side of Figure 1 demonstrates four cases of morpheme and buffer combinations that this model permits. In essence, the design of the morph class allows all class members to coexist; but only word forms where all required conditions are actually set are allowed. For example, the first case of (سُفَرَاءَ *su-*

	Morph Order		
	DBPrefix	**DBStem**	**DBSuffix**
O1	[Conj] [Prep]	[NomStem] [NomBuff]	[NomSuff.IG]
O2	[Conj] [Prep]	[NomStem] [NomBuff]	[NomSuff.CG] [Pronoun]
O3	[Conj] [Prep] [Determiner]	[NomStem] [NomBuff]	[NomSuff.DG]

Example	Order
sufaraA+'+a	O1
safiyr+aAt+i+him	O2
sufaraA+ŷ+i+him	O2
Al+safiyr+aAt+i	O3

		Class	**Lemma/ Morpheme**	**Form**	**Gloss**	**gen**	**num**	**stt**	**cas**	**Set Conds**	**Required Conds**	sufaraA+'+a	safiyr+aAt+i+him	sufaraA+ŷ+i+him	Al+safiyr+aAt+i
Lexicon	L1a	[NomStem]	*safiyr*	*safiyr*	ambassador	-	-	-	-		MS|FS|FP		✓		✓
	L1b	[NomStem]	*safiyr*	*sufaraA*	ambassador	m	p	-	-	#A' #dip	MS	✓		✓	
Prc	P1	[Determiner]													
	P2	[Determiner]	Prc.Al	Al	the										✓
Buffers	B1	[NomBuff]									else				
	B2a	[NomBuff]		'							#A'	✓			
	B2b	[NomBuff]		ŷ							#A' obj suff-i			✓	
	B2c	[NomBuff]		ŵ							#A' obj suff-u				
Suffixes	S1a	[NomSuff.IG]	Suff.MSIG	ĩ		m	s	i	g	MS	else				
	S1b	[NomSuff.IG]	Suff.MSIG	a		m	s	i	g	MS	#dip	✓			
	S2	[NomSuff.IG]	Suff.FPIG	aAt+ĩ		f	p	i	g	FP					
	S3	[NomSuff.CG]	Suff.MSCG	i		m	s	c	g	MS suff-i				✓	
	S4	[NomSuff.CG]	Suff.FPCG	aAt+i		f	p	c	g	FP suff-i			✓		
	S5	[NomSuff.DG]	Suff.MSDG	i		m	s	d	g	MS					
	S6	[NomSuff.DG]	Suff.FPDG	aAt+i		f	p	d	g	FP					✓
Enclitics	C1	[Pronoun]													
	C2a	[Pronoun]	Pron.3MP	hum	their					obj	else				
	C2c	[Pronoun]	Pron.3MP	him	their					obj	suff-i		✓	✓	

Figure 1: A sample of the CamelMorph system implementation for Arabic nominals. The character '|' represents the boolean OR, and `else` represents a negation of the disjunction of conditions below it in the same morpheme. The greyed out elements are not handled in this sample. See Appendix B for condition meanings.

faraA+'+a) uses three elements, which together set the conditions (`#A'`, `#dip`, `MS`) and require the same conditions (`#A'`, `#dip`, `MS`). An implausible form such as سُفَرَائِي* **sufaraA+ŷ+ĩ*) would not be allowed as these elements set the conditions (`#A'`, `#dip`, `MS`) but require the conditions (`MS`, `#A'`, `obj`, `suff-i`, and not `#dip`) – which cannot hold.

Finally, we note that the conditions are agnostic to functional features, and are only concerned with surface form. For example, the lemma هَوَاء *hawaA'* 'air' in its functionally masculine singular form would have the stem هَوَا *hawaA*, and set the condition `#A'`, the same condition set by the stem سُفَرَا *sufaraA* 'ambassadors', which is functionally plural.

Debugging and Quality Check The space of combinations to validate in the actual system is in the order of billions, of which only a fraction is valid. To debug this system, the generator engine was run on a subset of the nominal paradigm – chosen along the dimensions which vary the most, using lemmas chosen to represent the continuum of annotated conditions, and the outputs were manually checked by an annotator.

6.2 Lexicographic Modeling

The approach we took to model the morphology of words allows us to clearly disentangle many variables such as case-state, gender-number, and stem class variations. The next step is the lexicographic modeling to group stems belonging to the same lexemes together. To aid us in modeling the lexicon systematically, we extracted stems and their features from the publicly available CALIMA$_{Star}$ DB (Taji et al., 2018), and extended its root annotations with patterns, stem paradigms, and lexeme paradigms automatically. With the help of that information, we proceeded to manually annotate (with conditions) and carefully check all the stem clusters (lexemes) for soundness with the help of three annotators. This resulted in all clusters being categorized into one of the lemma paradigms that can be found in Appendix C. Future lemmas can therefore be added to the lexicon with ease by determining which paradigm they belong to without worrying about conditions. Conditions are only added upon determining the stem paradigm which mainly depends on the surface pattern and form. Were the lexicon conditions not purely concerned with form, it would have not been possible to do

	ms	mp	fs	fp	Example	
(a)	①+MS	①+MP	①+FS	①+FP	موظّف	employee
	①+MS	①+FP	①+FS	①+FP	بروفسور	professor
	①+MS	②+MS	①+FS	①+FP	طالب/طلاب	student
	①+MS	②+FS	①+FS	①+FP	سيد/سادة	master
	①+MS	②+MS	③+MS	③+FP	أحمر/حمر/حمراء	red
(b)	①+MS				حبّ	love
	①+MS	①+MP			أقدم/ون	elder
	①+MS	①+FP			امتحان/امتحانات	exam
	①+MS	②+MS			يوم/أيّام	day
	①+MS	②+FS			دواء/أدوية	medication
	①+MS	②+MP			ابن/بنون	son
	①+MS	②+FP			همْز/همَزات	temptation
(c)			①+FS		محبّة	affection
				①+FP	معلوماتيات	informatics
			①+FS	①+FP	مجلّة/مجلّات	magazine
			①+FS	②+MS	جريدة/جرائد	newspaper
			①+FS	②+FP	حَمْلَة/حَمَلات	campaign
(d)			①+MS		شورى	consultation
				①+MS	نساء	women
			①+MS	①+FP	دنيا	world
			①+MS	②+FP	بنت/بنات	girl
			①+MS	②+MS	نار/نيران	fire
(e)	①+FS	①+FP			خواجة/خواجات	foreigner
	①+FS	①+FS			رحّالة	explorer
	①+FS	②+MS			خليفة/خلفاء	caliph

Table 4: Examples of different lexicographic classes with different degrees of completeness and form-function matching. Greyed out cells mark cases with mismatching form-function in gender or number, or using secondary stems. See Appendix C for the full table.

that. Therefore the CAMELMORPH approach objectively renders the annotators' job simpler as the only layer they are required to interface with is the **Lexicon**. The annotators should not have to deal with conditions which are internal to the closed-class specifications, i.e., **Proclitics (Prc)**, **Buffers**, **Suffixes**, and **Enclitics** (see Figure 1).

As part of this effort, we developed guidelines for making decisions on boundaries between lexemes by (a) morpho-syntactic behavior, e.g., agreement patterns and their interaction with rationality (Alkuhlani and Habash, 2011), and (b) semantic change and relationships, e.g., lexical specification turning adjectives into nouns, or systematic derivational relationships between mass/collective nouns and their instance noun forms. Given the high degree of variability among nominal lexemes, we developed models for well-formedness checks to identify out-of-norm clusters for quality check.

Table 4 shows 25 lemma paradigms with varying paradigm completeness and gender-number form-function consistency. Circular digits indicate shared stem indices.

6.3 Statistics

In this section, we discuss the statistics of our specifications (**Our Specs**) and their associated resulting DB (**Our DB**), and we compare **Our DB** with the **Calima MSA** DB (Taji et al., 2018),[5] as a baseline, since both have the exact same format. Table 5 contains counts related to the three different entities.

We note that the number of lemmas is the same in **Our Specs** and **Our DB**, naturally, and is only slightly larger than **Calima MSA**'s. While the number of stems is almost the same in **Our DB** and **Calima MSA**, it is 13% less in **Our Specs** showing that we are able to get comparable results from a more succinct, and hence, more annotator-friendly, way using our morphological modeling. Similarly, the small number of morphological modeling elements (Table 5.b) and the large number of complex prefix/suffix sequences they produce (Table 5.c) highlight our approach's modeling power. The main reasons for the higher numbers in **Our DB** in Table 5.c are the modeling of the undefined case,[6] and the addition of the question proclitic أَ+ *Âa*+, which is only present in a few hard-coded cases in **Calima MSA**. These differences translate into **Our DB** having roughly two times more analyses than **Calima MSA**. The increase is still sensible when clitics are excluded, with an increase of ~26% in the analysis count (Table 5.d).[7]

7 Evaluation

We assess the quality of our system by (a) evaluating its coverage of the *training* portion of the Penn Arabic Treebank (PATB; latest versions of parts 1,2,3) (Maamouri et al., 2004) as defined by Diab et al. (2013), and (b) comparing the analyses it generates with those of **Calima DB** over a list of specific words.

Morphological Coverage For the coverage experiment, we drop all incomplete PATB gold analyses marked with placeholder values ($\sim$1% of all entries). Of the rest, we are able to recall 95.3% of gold analyses provided by the PATB (94.5% in

[5]Version: `calima-msa-s31_0.4.2.utf8.db`.

[6]The **Calima MSA** model produces a number of analyses with case *undefined* for some suffixes, e.g., كِتَابَات kitaAbaAt 'writings' in contrast with defined cases such as كِتَابَاتُ kitaAbaAtu (see full set in Table 2). However, this treatment is not consistent for all suffixes. In **Our DB**, we extend all suffixes with case *undefined* variants that are in common use.

[7]The statistics in Table 5.d are computed using combinatorics, not generation.

		Our Specs	Our DB	Calima MSA		
(a)	**Lemmas (Stems)**	27,023 (33,497)	27,023 (37,910)	26,990 (38,323)	**Lemmas (Stems)**	(a)
	Noun	19,858 (25,293)	19,858 (28,302)	19,970 (29,370)	*Noun*	
	Adjective	6,922 (7,921)	6,922 (9,184)	6,808 (8,703)	*Adjective*	
	Comparative Adjective	243 (283)	243 (424)	212 (250)	*Comparative Adjective*	
(b)	**DBPrefix Morphemes (Allom.)**	18 (20)	213	77	**DBPrefix Sequences**	(c)
	DBSuffix Morphs (Allom.)	99 (197)	614	391	**DBSuffix Sequences**	
	Stem Buffers	22	3,442	1,423	**Compatibility Tables**	
	Unique Condition Terms	51	83,649,166	28,359,701	**Unique Diacritized Forms**	(d)
	Morph Order Lines	42	246,880,683	126,176,265	**Unique Analyses**	
			1,300,068	1,041,949	**Unique Analyses (no Clitics)**	

Table 5: Statistics comparing our morphological specifications and DB with Calima MSA on Arabic nominals.

unique type space) based on matching on all of lemma, diacritization, and morphological analysis (BW tag). We performed a human evaluation on a sample of 100 unique words from the mismatching *noun* instances chosen randomly (but weighted by the PATB frequency of the gold analysis). We found that 86% of mismatches are due to gold inconsistencies or errors. These include – among other issues listed in Section 4.2 – spelling inconsistencies between lemma and stem, or attributing a stem to a wrong lemma because of paradigm ambiguity. Our system produces valid analyses for these cases, but it fails for the remaining 14%. A similar 100 adjective sample reveals that 95% of mismatches are due to inconsistent gold tags, and are mainly due to a wrong POS attribution and lemma-stem spelling mismatch. Our system handles these cases correctly. In the released version, we made sure to include all missing analyses.

Analysis Evaluation Finally, we choose 50 random words from the 100-sample taken for the nouns in the previous paragraph for closer inspection, and we manually compared all analyses generated by both **Our DB** and **Calima DB** for these words. Of the union of all manually inspected analyses generated by the two systems (1,406 analyses for the 50 words), 21% are generated by both, 44% are generated only by **Our DB**, and 35% are generated only by **Calima DB**. We find that about 60% of the analyses generated only by **Our DB** are due to unmodeled or incompletely modeled phenomena in **Calima DB**, e.g., the question proclitic morpheme or some instances of the *undefined* case. The remaining 40% are due to inaccurate modeling on the **Calima DB** side. For example, **Calima DB** only provides one lemma for معلومات *maς.luwmaAt*, معلوم *maς.luwm* 'known', and misses the lemma معلومة *maς.luwmah̄* 'a piece of information', while **Our DB** provides both.

One systematic mistake is allowing the +ال *Al*+ determiner to attach to construct noun stems, whereas this behavior should only be restricted to adjectives participating in a *False Idafa* construction (إضافة لفظية), e.g., الأبيض اللّون 'the-white-colored' (Hawwari et al., 2016). Other mistakes include wrong lemma gender, and spelling inconsistencies between lemma and stem. Finally, about 6% of the **Our DB** analyses in this sample are admittedly wrong, but can easily be fixed in our specifications.

8 Conclusion and Future Work

We presented a detailed review of the challenges of modeling Arabic nominals morphologically and lexically. We developed an annotator-friendly and easily extendable system for modeling nouns, adjectives and comparative adjectives building on an existing open-source framework for Arabic morphology. We evaluated our system against a popular analyzer for Arabic, showing that our resulting database is more consistent and provides a more accurate linguistic representation. We make our models, system details, and guidelines publicly available (see footnote 1).

In the future, we plan to extend our work to other MSA POS tags and to Arabic dialects. We also plan to make our model more robust to spelling variations and integrate it in downstream applications, e.g., morphological disambiguation, tokenization and diacritization (Obeid et al., 2022), readability visualization (Hazim et al., 2022), gender rewriting (Alhafni et al., 2022), error typing (Belkebir and Habash, 2021), and grammatical error correction (Alhafni et al., 2023).

9 Limitations

The current system faces several limitations: it lacks robustness in handling input orthographic errors, restricting its usability in spontaneous orthography contexts. Additionally, it does not comprehensively model valid spelling variants commonly used. The high coverage generates numerous options, including some less likely but theoretically correct ones, potentially overwhelming downstream processes without optimized filtering and ranking models. There is also a lack of explicit linking across lemmas sharing derivational history. Furthermore, the model is currently limited to nouns, adjectives, and comparative adjectives, representing the open-class nominals at this stage.

10 Ethics Statement

All annotators received fair wages for their contributions to the development, quality checking of lexical resources, and debugging the overall system. While we recognize the possibility of unforeseen errors in our lexical resources, we anticipate that the associated risks to downstream applications are minimal. Additionally, we acknowledge that, like many other tools in natural language processing, our tool could be misused in the wrong hands for manipulating texts for harmful purposes.

References

Imad A. Al-Sughaiyer and Ibrahim A. Al-Kharashi. 2004. Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology*, 55(3):189–213.

Bashar Alhafni, Nizar Habash, and Houda Bouamor. 2022. User-centric gender rewriting. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 618–631, Seattle, United States. Association for Computational Linguistics.

Bashar Alhafni, Go Inoue, Christian Khairallah, and Nizar Habash. 2023. Advancements in Arabic grammatical error detection and correction: An empirical investigation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6430–6448, Singapore. Association for Computational Linguistics.

Sarah Alkuhlani and Nizar Habash. 2011. A Corpus for Modeling Morpho-Syntactic Agreement in Arabic: Gender, Number and Rationality. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, Portland, Oregon, USA.

Mohamed Altantawy, Nizar Habash, and Owen Rambow. 2011. Fast Yet Rich Morphological Analysis. In *Proceedings of the International Workshop on Finite-State Methods and Natural Language Processing (FSMNLP)*, Blois, France.

Mohamed Altantawy, Nizar Habash, Owen Rambow, and Ibrahim Saleh. 2010. Morphological Analysis and Generation of Arabic Nouns: A Morphemic Functional Approach. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Valletta, Malta.

Ramy Baly, Hazem Hajj, Nizar Habash, Khaled Bashir Shaban, and Wassim El-Hajj. 2017. A sentiment treebank and morphologically enriched recursive deep models for effective sentiment analysis in arabic. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(4):23.

Kenneth Beesley. 1996. Arabic Finite-State Morphological Analysis and Generation. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 89–94, Copenhagen, Denmark.

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*.

Riadh Belkebir and Nizar Habash. 2021. Automatic error type annotation for Arabic. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 596–606, Online. Association for Computational Linguistics.

Mohamed Boudchiche, Azzeddine Mazroui, Mohamed Ould Abdallahi Ould Bebah, Abdelhak Lakhouaja, and Abderrahim Boudlal. 2017. AlKhalil Morpho Sys 2: A robust Arabic morpho-syntactic analyzer. *Journal of King Saud University - Computer and Information Sciences*, 29(2):141–146.

Ron Buckley. 2004. *Modern Literary Arabic: A Reference Grammar*. Librairie du Liban.

Tim Buckwalter. 2004. Buckwalter Arabic Morphological Analyzer Version 2.0. LDC catalog number LDC2004L02, ISBN 1-58563-324-0.

Mona Diab, Nizar Habash, Owen Rambow, and Ryan Roth. 2013. LDC Arabic treebanks and associated corpora: Data divisions manual. *arXiv preprint arXiv:1309.5652*.

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. 2007. *Arabic Morphological Representations for Machine Translation*, pages 263–285. Springer Netherlands, Dordrecht.

Nizar Habash, Ryan Gabbard, Owen Rambow, Seth Kulick, and Mitch Marcus. 2007a. Determining case in Arabic: Learning complex linguistic behavior requires complex linguistic features. In *Proceedings of* *the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic.

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, Abdelhadi Soudi, and Tim Buckwalter. 2007b. 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.

Abdelati Hawwari, Mohammed Attia, Mahmoud Ghoneim, and Mona Diab. 2016. Explicit fine grained syntactic and semantic annotation of the idafa construction in Arabic. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3569–3577, Portorož, Slovenia. European Language Resources Association (ELRA).

Reem Hazim, Hind Saddiki, Bashar Alhafni, Muhamed Al Khalil, and Nizar Habash. 2022. Arabic word-level readability visualization for assisted text simplification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 242–249, Abu Dhabi, UAE. Association for Computational Linguistics.

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.

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.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *Proceedings of the International Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

Alexis Amid Neme and Éric Laporte. 2013. Pattern-and-root inflectional morphology: the arabic broken plural. *Language Sciences*, 40:221–250.

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 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Majdi Sawalha and Eric Atwell. 2008. Comparative evaluation of Arabic language morphological analysers and stemmers. In *Proceedings of COLING 2008 22nd International Conference on Comptational Linguistics (Poster Volume)*, pages 107–110, Manchester.

Otakar Smrž. 2007a. 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.

Otakar Smrž. 2007b. *Functional Arabic Morphology. Formal System and Implementation*. Ph.D. thesis, Charles University in Prague, Prague, Czech Republic.

Abdelhadi Soudi, Violetta Cavalli-Sforza, and Abderrahim Jamari. 2001. A Computational Lexeme-Based Treatment of Arabic Morphology. In *Proceedings of the Workshop for Arabic Natural Language Processing (WANLP)*, pages 50–57, Toulouse, France.

Gregory T. Stump. 2001. *Inflectional Morphology. A Theory of Paradigm Structure*. Cambridge Studies in Linguistics. Cambridge University Press.

Dima Taji, Salam Khalifa, Ossama Obeid, Fadhl Eryani, and Nizar Habash. 2018. An Arabic Morphological Analyzer and Generator with Copious Features. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology (SIGMORPHON)*, pages 140–150.

Jinxi Xu, Alexander Fraser, and Ralph Weischedel. 2002. Empirical Studies in Strategies for Arabic Retrieval. In *Proceedings of the International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 269–274, Tampere, Finland. ACM.

Nasser Zalmout and Nizar Habash. 2017. Don't throw those morphological analyzers away just yet: Neural morphological disambiguation for Arabic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 704–713, Copenhagen, Denmark.

A Glossary of Terms and Abbreviations

Term	Abbreviation	Arabic Equivalent	Example
Adjective	A	صفة	أحمر
Affixes (Prefixes/Suffixes)		لواصق (سابقة/لاحقة)	كاتب+ين، كاتب+ٌ
Broken Plural		جمع تكسير	كُتّاب
Case - Accusative	a (cas)	منصوب	كاتباً
Case - Genitive	g (cas)	مجرور	كاتبٍ
Case - Nominative	n (cas)	مرفوع	كاتبٌ
Clitics (Proclitics/Enclitics)		زوائد (سابقة/لاحقة)	وَ+كاتب+هُ
Defective Nominal		اسم معتلّ الآخر	فتى، قاضي
Derivation		اشتقاق	كتب ← مكتب
Diptote		ممنوع من الصرف	أحمر
Form Features		سمات لفظية	مذكر/مؤنث لفظاً
Form Gender - Feminine	F	مؤنث لفظيّ	كاتبة، خليفة، نابغة
Form Gender - Masculine	M	مذكر لفظيّ	كاتب، سماء، دنيا
Functional Features		سمات معنوية	مذكر/مؤنث، نكرة/معرفة
Gender - Feminine	f (gen)	مؤنث	كاتبة
Gender - Masculine	m (gen)	مذكر	كاتب
Inflection		تصريف	كاتب ← كاتبين، كُتُب،...
Lemma		مدخل معجمي	كاتب
Nominal / Noun	N	اسم	كاتب
Number - Dual	d (num)	مثنى	كاتبين
Number - Plural	p (num)	جمع	كُتُب
Number - Singular	s (num)	مفرد	كاتب
Pattern		وزن	فاعِل (1A2i3)
Person - First	1	متكلم	كاتب+ي
Person - Second	2	مخاطب	كاتب+ك
Person - Third	3	غائب	كاتب+ه
POS (part-of-speech)		فئة/نوع/قسم الكلمة	اسم
Radical		أصل	ك
Rationality - Irrational	I	غير عاقل	كاتب
Rationality - Rational	R	عاقل	إنسان
Root		جذر	ك.ت.ب
Sound Plural		جمع سالم	كاتبون، كاتبات
State - Construct	c (stt)	مضاف	كاتب القواعد
State - Definite	d (stt)	معرفة	الكاتب
State - Indefinite	i (stt)	نكرة	كاتب
Stem		جذع	وَ+بِ+كُتُبِ+هِم

Table 6: Table featuring the Arabic equivalents of the terms used in this paper, including their abbreviations.

B Conditions Index

Condition	Meaning	Classes that *set* it	Examples
#A'	Generally *set* by stems ending in اء *A'*. Stem in the lexicon is written without the *hamza* as it acquires it from the buffer to which it connects. This allows for the multiple stem endings depending on the morphological context.	[NomStem]	Aib.tid**A'** ابتداء 'start' [ms] Saw.f**A'** صوفاء 'woolen' [fs] buŵas**A'** بؤساء 'miserable' [mp] nis**A'** نساء 'women' [fp]
#dip	Generally *set* by stems of diptotes, resulting in partial syncretism in the indefinite state.		Âað.laq أذلق 'fluent' [ms] kub.raý كبرى 'larger, largest' [fs] šarAŷiT شرائط 'tapes' [mp] tarAniym ترانيم 'hymns' [fp]
suff-u	Generally *set* by suffixes and *required* by buffers. Denotes that a suffix starts with a *Damma (u),* making sure that it attaches to the correct buffer variant.	[NomSuff.XXCG]	mab.daŵ**u**hu مبدؤه 'his principle' [ms]
suff-i	Generally *set* by suffixes and *required* by buffers. Denotes that a suffix starts with a *kasra (i),* making sure that it attaches to the correct buffer variant.		mab.daŷ**i**hi مبدئه 'his principle' [ms]
obj	*Set* by clitics to denote the presence of an attached clitic object pronoun which affects certain variations in suffixes and buffers.	[Pronoun]	nis.watu**hu** نسوته 'his women' [fp] mabnaý+**iy** = mabnaA**ya** مبناي 'my building' [ms]
MS	*Set* by suffixes and *required* by the stem complex (stem + buffer). The suffixes that set it are all gender-number neutral. If a lexeme does not require FS in any of it stems, then at least one must require MS as it is the default suffix.	[NomSuff.XXIG] [NomSuff.XXCG] [NomSuff.XXDG]	ςitAb عتاب 'reprimand' [ms] faHošA' 'atrocity' [fs] Sun~Aς صنّاع 'manufacturers' [mp] maqArib مقارب 'shortcuts' [fp]
FP	*Set* by suffixes and *required* by the stem complex (stem + buffer). It represents the ات *At* morpheme and its allomorphs.		musowad~**At** مسودّات 'drafts' [fp] DuγuwT**At** ضغوطات 'stresses' [mp]

Table 7: Index of pre-defined conditions used in Figure 1 and their meanings, with examples.

C Nominal Lemmas Paradigm Index

Stems	ms	md	mp	fs	fd	fp	Example	Noun	Adj.	Comp. Adj.	Total
①	①+MS	①+MD	①+MP	①+FS	①+FD	①+FP	موظّف	2,867	6,724	1	**9,592**
①-②	①+MS	①+MD	②+MS	①+FS	①+FD	①+FP	طالب/طلاب	142	13	0	**155**
①-②-③	①+MS	①+MD	②+MS	③+MS	③+MD	③+FP	أحمر/حمر/حمراء	2	153	8	**163**
①-②	①+MS	①+MD	②+FS	①+FS	①+FD	①+FP	سيّد/سادة	17	2	0	**19**
①-②	①+MS	①+MD	②+MS	①+FS	①+FD	②+MS	إنسان/أناس	8	4	0	**12**
①	①+MS	①+MD	①+FP	①+FS	①+FD	①+FP	بروفسور	8	0	0	**8**
①	①+MS	①+MD					حبّ	4,644	0	205	**4,849**
①	①+MS	①+MD	①+FP				امتحان/امتحانات	3,094	0	0	**3,094**
①-②	①+MS	①+MD	②+MS				يوم/أيّام	2,422	14	2	**2,438**
①-②	①+MS	①+MD	②+FS				دواء/أدوية	143	0	0	**143**
①			①+MS				أوزاع	49	0	0	**49**
①	①+MS						تحت، فوق، جنب	32	2	0	**34**
①	①+MS	①+MD	①+MP				أقدم/ون	6	0	24	**30**
①			①+MP				عشرون	8	0	0	**8**
①-②	①+MS	①+MD	②+FP				قطر/قطورات، همْز/همَزات	7	0	0	**7**
①				①+FS	①+FD	①+FP	مجلّة/مجلّات	4,725	0	0	**4,725**
①-②				①+FS	①+FD	②+MS	جريدة/جرائد	891	0	0	**891**
①						①+FP	معلوماتيات	185	0	0	**185**
①-②				①+FS	①+FD	②+FP	حَمْلَة/حَمَلات	168	0	0	**168**
①				①+MS	①+FD	①+FP	بيتزا	154	0	0	**154**
①				①+FS	①+FD		محبّة	35	0	0	**35**
①				①+MS	①+MD		شورى	95	0	0	**95**
①				①+MS	①+MD	①+FP	دنيا	69	6	1	**76**
①-②				①+MS	①+MD	②+MS	عين/عيون، نار/نيران	45	0	0	**45**
①						①+MS	نساء	20	0	0	**20**
①-②				①+MS	①+MD	②+FS	خِوان/أخْوِنة	1	0	0	**1**
①-②				①+MS	①+MD	②+FP	بنت/بنات	4	0	0	**4**
①-②	①+FS	①+FD	②+MS				خليفة/خلفاء	2	0	0	**2**
①	①+FS	①+FD	①+FP				خواجة/خواجات	1	0	0	**1**
①	①+FS	①+FD	①+FS				رحّالة	1	0	0	**1**
...	...	...	...	...	...	...	...	13	4	2	**19**
							Total	**19,858**	**6,922**	**243**	**27,023**

Table 8: Index of basic lemma paradigms identified. See Appendix A for abbreviations and Section 4.2 for an explanation of the form feature suffix sets. Statistics included pertain to the number of lemmas per paradigm for each POS.