Arabic Morphological Analysis Techniques

A Survey and Classification

Ameerah Alothman[1], AbdulMalik Alsalman[2]
Department of Computer Science, College of Computer and Information Sciences
King Saud University, Riyadh, Saudi Arabia

***Abstract*—Recently, activity surrounding Arabic natural language processing has increased significantly. Morphological analysis is the basis of most tasks related to Arabic natural language processing. There are many scientific studies on Arabic morphological analysis, yet most of them lack an accurate classification of Arabic morphology and fail to cover both recent and traditional techniques. This paper aims to survey Arabic morphological analysis techniques from 2005 to 2019 and to organize them into a reasonable and expandable classification system. To facilitate and support new research, this paper compares the currently available Arabic morphological analyzers, reaches certain conclusions, and proposes some promising directions for future research in Arabic morphological analysis.**

***Keywords*—Arabic analyzer; Arabic lexicon; classification morphology; morphological analysis; natural language processing**

I. Introduction

Since the advent of the computing era, researchers have been trying to develop systems which can interact with humans; these systems play an essential role in facilitating human life by saving time and improving the quality of work. Morphological analyzers are one such system and constitute an important component of many applications dealing with natural language processing (NLP), machine translation, information search and retrieval, and more.

Morphology is a challenge in Arabic natural language processing (ANLP), and a somewhat complex task. This is because the most important characteristic of Semitic languages is their nonconcatenative nature. Arabic words are composed of roots, derived from certain patterns extracted from stems and their affixes. One root and a small number of patterns with several affixes can form many stems (word formations).

Accordingly, it is necessary to study and classify the techniques of Arabic morphological analyses, because doing so may contribute to greater understanding and improved construction of morphological methodologies, and will pave the way for future researchers in the field of ANLP.

The main purpose of this article is to survey Arabic morphological analysis techniques and bridge the gap in scientific survey studies from 2005 to 2019. This paper is organized as follows: In the second section, we provide basic definitions for this article's most frequently used terms. In the third section, we propose a classification of Arabic morphological analysis techniques and describe some of the shortcomings of earlier classifications. In the fourth section, we present a survey of Arabic morphological analysis techniques. The fifth section presents a discussion of the comparative study undertaken. Finally, we conclude and summarize some important future directions for Arabic morphological analysis techniques. We adopt Buckwalter [1] for the transliteration of Arabic characters, providing transliterations in brackets where relevant.

II. Basic Definitions

There are many terms related to Arabic morphological analyses, and many papers have made great efforts towards the Arabization and standardization of these terms. The book Introduction to Arabic Language Processing, [2] as well as its translation into the Arabic language [3], is one of the most important references in this field of study. Table I presents the meanings and translations of the most frequently used terms in this research.

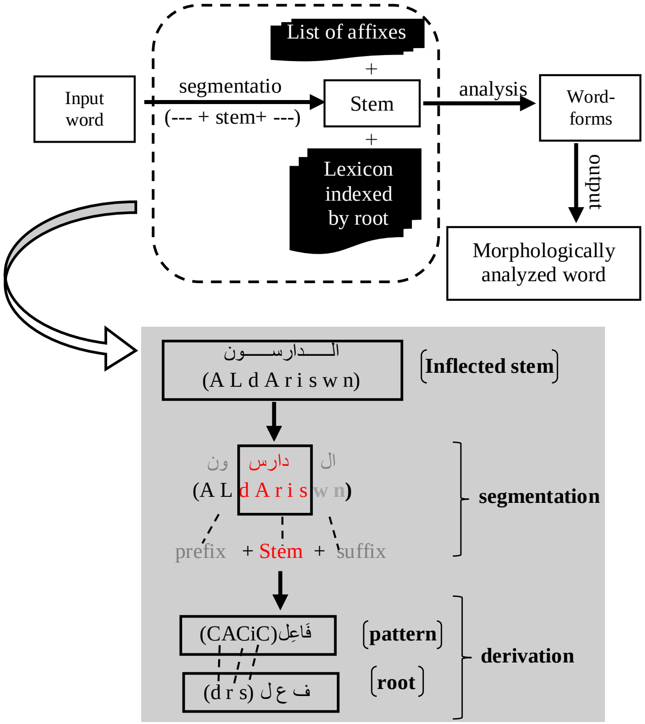

Fig. 1. Root-Pattern Morphology Process.

TABLE. I. BASIC TERMS USED FREQUENTLY IN ARABIC MORPHOLOGICAL ANALYSIS TECHNIQUES (ARRANGED ALPHABETICALLY)

Term	**Translation**	**Meaning**	**Example**
Affix	*اللواصق*	Three types that attach to the root: prefixes, suffixes, and infixes.	ال - فـ - ة
Basic Arabic letter used in patterns	*الحروف العربية الأساسية في الأوزان*	The three basic letters used to construct a pattern in Arabic, which are: [l *ل*- E *ع*- f *فـ*].	*فَعِلَ*(faEila) *يَفْعَلُ*(yafEalu)
Inflected stem	*الجذع الإعرابي*	A stem that may have a prefix and/or suffix to provide meaningful context, also known as a surface word.	فسيكتبونها (fsyktbwnhA)
Lexeme	*المُعَيْجِمَة*	The smallest part of the lexicon that has meaning.	بيت - حقل
Long vowel	*حروف المد*	Also called the "weak letters set" (*أحرف العلة*); consists of three letters of the alphabet ("a"*الألف*, "w"*الواو*, and "y"*الياء*).	The long vowel in قال (qAl) is ا (A)
Morpheme	الوحدة الصرفية	The smallest unit of the language that has meaning.	ال, إلى أكل ,
Morphological analysis techniques	*تقنيات التحليل الصرفي*	The process used to determine all possible morphological analyses of a word.	See shaded part of Fig. 1
Pattern	*الصيغة أو الوزن الصرفي*	Abstract CV-template (C: Consonant, V: Vowel) representation of how to order the root and short vowels (and some affixes) to generate the stem. It conveys a grammatical meaning, such as part of speech (POS) and tense.	(fEl) فعل (fAEl)فاعل
Root	*الجذر*	A sequence of three (most commonly), four (less commonly), or five (rarely) consonants. It can be derived based on various patterns. It identifies the general meaning of a word.	(zrE)زرع
Short vowel or Diacritic	*الحركات*	Includes diacritics, which are marks usually written above or below a letter. Diacritics include: 1) three short vowels ("a",*الفتحة* "u" *الضمة* and "i" *الكسرة*), and the absence of any vowel (*"ـْ" السكون*); 2) three nunations (*التنوين*) occurring in the final positions of a word in nominals only; and 3) Shadda ("ـّ" *الشدة*).	ـَ (a) ـِ (i) ـُ (u)
Stem	*الجذع*	The core of concatenative morphology, it is a surface word generated by inserting the radicals of roots and short vowels into the pattern template slots (e.g., the interdigitating of roots with the patterns).	Stem in (fsyktbwnhA) (فسيكتبونها) is (yktbwn) (يكتبون)

III. CLASSIFICATION OF ARABIC MORPHOLOGICAL ANALYSIS TECHNIQUES

Many scientific papers have tackled the classification of Arabic morphology, and several reviews exist of the most cited Arabic morphological analyzers [4-9]. These studies have many shortcomings, including the following: 1) They are very general in their classification process, and most existing analyzers are classified under one category, "linguistic". 2) They are somewhat outdated (especially in terms of classification methods) and do not take new techniques into consideration. 3) The authors of these review papers do not provide a standard or basis for the construction of their morphological analyzers or define the approaches that were used to analyze words.

Our aim is to bridge the gaps in the previous studies. Therefore, we have classified Arabic morphology in a more detailed and precise manner than previous studies, in terms of the units used in the analysis. This is based on the approach adopted (linguistic or data-driven lexicon) to adequately clarify the variation in work (see Fig. 2). We also limit ourselves to morphology work carried out after 2004, so that our research will complement the comprehensive survey conducted in this field by Al-Sughaiyer and Al-Kharashi [4] in 2004.

According to [4], the classification of Arabic morphological analysis techniques falls into four main approaches, namely, pattern-based, combinatorial, table lookup, and linguistic. This classification neglects the core unit of how to build lookup tables or linguistic rules (i.e. What should they be based on – root, stem, or lexeme?). As we know, Semitic languages are rich in morphology, and therefore the unit of Arabic used in the analysis must first be specified. Moreover, this classification ignores the machine learning approaches that have received more attention in the latest research. In addition, it does not differentiate between different levels of linguistics and does not take Arabic syntax into consideration. Lastly, it includes a pattern-based approach, which can be more accurately described as part of an approach rather than a separate approach in itself. In the next section, we present in greater detail the proposed classification, which is legitimate and covers all recent and traditional techniques.

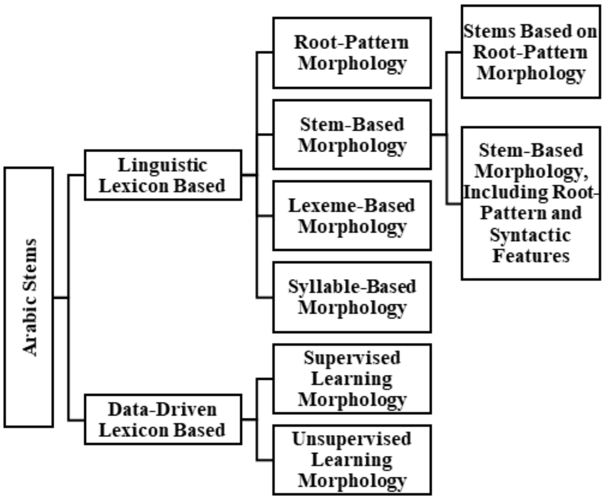
Arabic Stems
Linguistic Lexicon Based
Data-Driven Lexicon Based
Root-Pattern Morphology
Stem-Based Morphology
Lexeme-Based Morphology
Syllable-Based Morphology
Supervised Learning Morphology
Unsupervised Learning Morphology
Stems Based on Root-Pattern Morphology
Stem-Based Morphology, Including Root-Pattern and Syntactic Features

Fig. 2. Suggested Classification of Arabic Morphological Analysis Techniques.

IV. Survey of Arabic Morphological Techniques

This section reviews the main approaches to building Arabic morphological analyzers found in the existing literature. Additionally, it lists the morphological systems that have adopted these approaches. Table II provides a summary of the approaches surveyed.

TABLE. II. SUMMARY OF SURVEYED APPROACHES

Approach	Morphology	Author and reference	Date	Known as	Test Data	Result (%)	Language coverage
Data-driven	Supervised	Elghamry [10]	2005	A constraint-based algorithm	2,700 unique words	Percentage of correct root = 92%	Information not available (N/A)
		Daya et al. [11]	2008	Identifying Semitic roots	N/A	Precision: 87.92%; Recall: 92.19%	MSA
		Boudlal et al. [12]	2011	A Markovian approach	38,022 words	Percentage of correct root: Training set: 98%; Testing set: 93.81%	Non-vowelized
	Unsupervised	Rodrigues and Ćavar [13]	2007	Learning Arabic morphology using statistical constraints	10,000 words from BAMA1 dataset	Root predicted with 75% precision	Non-vowelized
		Snyder and Barzilay [14]	2008	Unsupervised multilingual learning	Snyder & Barzilay (S&B) dataset	Performance of automatic segmentation: Precision = 67.75% Recall = 77.29%	N/A
		Poon et al. [15]	2009	Unsupervised with log-linear models	– S&B dataset – Arabic Treebank (ATB)	– S&B : F1 = 90 – ATB: F1 = 80.2	N/A
		Botha and Blunsom [16]	2013	Adaptor grammar for learning	– BW corpus (without diacritics) – BW’ with diacritics – Quranic Arabic (QA)	– Triliteral root identification accuracy: BW = 67.1% BW' = 0.7% – Segmentation: BW = 73.66% BW' = 74.54% – QA has a low performance (excluded from comparison)	Vowelized and non-vowelized
		Fullwood and O’Donnell [17]	2013	Learning nonconcatenative morphology	N/A	Accuracy = 92.3%	Vowelized
		Khaliq and Carroll [18, 19]	2013	Unsupervised induction of Arabic root and pattern lexicons	Quranic Arabic Corpus (QAC)	Root extraction accuracy = 87.2%	Non-vowelized
Linguistic	Root-pattern	Gridach and Chenfour [20]	2014	Developing a new system for Arabic morphological analysis and generation	ALECSO Corpus	Accuracy = 95.08%	MSA
	Lexeme	Habash and Rambow [21]	2006	MAGEAD	Penn Arabic Treebank (PATB), Levantine Arabic Treebank (LATB)	**MSA:** Context type recall (CTyR) = 52.9% Context token recall (CToR)= 60.4% **LEV:** CTyR = 95.4% CToR= 94.2%	MSA & Levantine
		Smrz [22]	2007	ElixirFM	N/A	N/A	N/A
		Habash [23]	2007	ALMOR	1m Arabic words from the United Nations Arabic-English corpus	Precision = 99.61% Recall = 87.78%	MSA
		Attia et al. [24]	2011	AraComLex	– 400,000 words from general news – 400,000 semi-literary words	– 87.13% coverage rate on words from the general news – 85.73% coverage rate on semi-literary words	MSA

		Habash et al. [25]	2012	CALIMA$_{EGY}$	Manually annotated EGY corpus	1 – Correct Answer = 84.1% 2 – Correct Answer = 92.1%	MSA, Dialectal Arabic (DA)
		Khalifa et al. [26]	2017	CALIMA$_{GLF}$	4,000 words from Emirati novels	Conventional Orthography for Dialectal Arabic (CODA) = 89.7%	MSA, DA
		Taji et al. [27]	2018	CALIMA$_{star}$	1m words from the Arabic Gigaword corpus	Coverage of 1.3 % out-of-vocabulary (OOV) rate	MSA, DA
	stems based on root-pattern morphology	Buckwalter [1, 28]	2004	BAMA2	N/A	N/A	MSA
		Maamouri et al. [29]	2010	SAMA 3	N/A	N/A	MSA
	stem-based morphology, including root-pattern and syntactic features	Sawalha et al. [7]	2013	SALMA	– 1000 words from Chapter 29 of the Qur’an, representing Classical Arabic (CA) – Corpus of Contemporary Arabic (CCA) representing MSA	Prediction accuracy of all features = 53.50% for the Qur’an 71.21% for the CCA	CA and MSA
		Boudchiche et al. [30]	2017	AlKhalil	– Tashkeela corpus – Nemlar corpus	99.31% coverage rate	Non-vowelized, partially or totally vowelized text

A. *Linguistic Lexicon-Based Approach*

In the linguistic lexicon-based approach, solid linguistic rules represented in the heavy lexicon are the core data upon which analysis depends. The lexicon contains two main sections: the first comprises word roots and/or patterns and/or stems, grouped in morphological ways, and the second contains any information related to these contents that the system shows in the results. This approach follows the steps in Fig. 3, with some variations depending on the lexicon and its analyses. The following shows the four basic linguistic lexicon-based approaches:

1) Root-pattern morphology: In brief, morphology is the study of the relationship between meaning and form. It is one of the most challenging tasks in Semitic languages like Arabic, Maltese, and Hebrew. For the most part, Arabic morphology is not concatenative (also called discontiguous or nonlinear). Arabic words are generated from their base roots [5]. In linguistics, there are several nonconcatenative methodologies for deriving the stems of words, because they provide linguistic information [6]. Root-pattern is one of these methodologies.

It is useful to briefly review one of the most important theories of nonconcatenative morphology. In 1979, McCarthy [31, 32] proposed a theorem accepted by linguists (especially computational) to form a stem through a derivational integration of roots and patterns. This mechanism is important for representing the structure of a word in Semitic language morphology.

McCarthy’s [32] work depends on autosegmentalizing the vowels and placing them in a separate tier from the pattern. It has three tiers, as seen in Fig. 4, where C stands for Consonant, V for Vowel:

1) Root tier: refers to consonantal segments, including the meaning of a lexeme, such as (k t b ك ت ب), which means “write”.

2) Pattern tier: refers to a prosodic template associated with a particular meaning or grammatical function such as ((katab) كَتَبَ = CVCVC =CaCaC), which means, “he wrote”.

3) Vocalization tier: represents pronounced letters and involves grammatical information such as tense, number, and derivational functions.

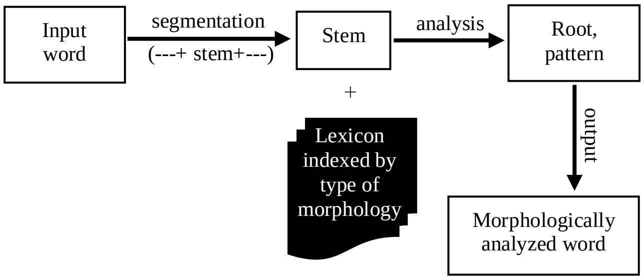

Fig. 3. The basic Steps of the Linguistic Lexicon-based Approach.

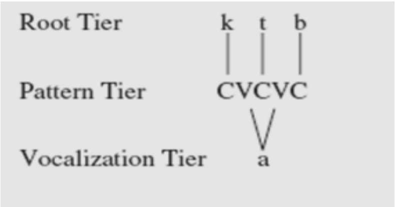

Fig. 4. An Example of McCarthy’s [32] Work.

To form an abstract stem, association rules are matched between consonants from the root tier and the pattern tier, and between vowels from the vocalization tier and from the pattern tier. There have been many systems attempting to model Arabic morphology based on McCarthy's theorem. Most of these systems adopted finite-state language modelling tools [33].

Root-pattern morphology depends on the root and pattern of the word entered for analysis (see Table III). The method involves building lexicons of roots and patterns (or lists of Arabic roots and affixes to cover all prefixes, suffixes, and infixes). Continuous research is being done to extract words that belong to one of the entries in these lists. This process is meant to output analysis of stem forms. Fig. 1 illustrates the main steps followed in this morphology.

One of the earliest published works to adopt this morphology was a system proposed by Hlal [34] and Hegazi and El-Sharkawi [35, 36]. It was also adopted by the Xerox lexicon [37], whose entries depend on root and pattern morphemes. Gridach and Chenfour [20] adopted this morphology with some variations in building their lexicon, depending on XML-based morphological definition language (XMODEL) for its construction.

2) Stem-based morphology: Dichy and Farghaly [6] and Farghaly and Senellart [38] support the claim that building a stem-based lexicon is more intuitive, efficient, and easy to develop and extend compared to a lexicon based on roots.

On the other hand, earlier Arabic morphologies were only responsible for the analysis and/or generation of the correct formations of Arabic words. Many Arabic NLP systems, such as machine translations and automatic summarizations, need linguistic information related to each lexical entry to ascribe elaborate knowledge to each word, in order to become more efficient. This information involves the tense of the verb, number, gender, and part of speech (POS), as well as syntactic features such as the type of subject or object, the count of nouns, and so on. In this context, one adds semantic information, such as the categorization of the noun as human, time, place, and so on. This linguistic information is associated with the stems, which are neither roots nor patterns nor a combination of them [6].

According to the above, Arabic stem-based morphology can achieve a more effective morphological strategy by reducing the complexity of word formations and granting linguistic and semantic information to each entry, thus eliminating the greater lexical gaps.

TABLE. III. EXAMPLES OF ROOT-PATTERN MORPHOLOGY

Root	Pattern	In Arabic	Meaning
د ر س (d r s)	(CaCaCa)فَعَلَ	(darasa) دَرَسَ	study
	(CACiC)فَاعِل	(dAris) دَارِس	student
	(CaC~aCa)فَعَّلَ	(dar~asa) دَرَّسَ	he teaches
	(CACiC)فَاعِل	(dAriswn) دَارِسُون	group of students

Two approaches have been built based on this morphology: 1) stems based on root-pattern morphology; and 2) stem-based morphology, including root patterns and syntactic features.

a) Stems based on root-pattern morphology: Briefly, this morphology can be described as follows: each existing lexical entry is checked against candidate entries integrating root and pattern (to generate a stem), in addition to prefix or suffix combinations. Therefore, if the lexicon in this morphology contains, for example, X root and Y pattern, then the XY root-pattern virtual links represent all possible stems, which must be severely restricted to give a reasonable number of meaningful words [33].

The major difference between root-pattern morphology and stems based on root-pattern morphology lies in their analysis mechanisms (see Fig. 1 and 5). The former uses the root and pattern morphemes themselves, while the latter uses stems based on root and pattern morphemes [39].

In this regard, the most famous Arabic analyzer to adopt this morphology is the Buckwalter Arabic Morphological Analyzer (BAMA) [1, 28]. BAMA is based on Buckwalter's lexicon, which is integrated with the Xerox lexicon [38].

Currently, there are three main versions of BAMA. BAMA 1.0 is available for public use, while BAMA 2.0 and Standard Arabic Morphological Analyzer (SAMA) 3.0 [29] are available through the Linguistic Data Consortium (LDC).

b) Stem-based morphology, including root patterns and syntactic features: Dichy and Farghaly [33] present the significance of syntactic features in Arabic computational morphology in detail. Systems based on this method produce a higher level of morphological analyzers, called morpho-syntactic analyzers. As we know, there are six linguistic levels: phonetics, phonology, morphology, syntax, semantics, and pragmatics (see Fig. 6). This approach takes advantage of the features of the syntax and morpheme levels.

This morphology differs from previous approaches because it applies the additional grammatical features step to results such as prepositions "ب"<b> and "ك"<k>, which only appear in the genitive case with nouns. These features play an important role in ensuring proper insertion of lexical entries, especially the main ones, such as nouns and verbs.

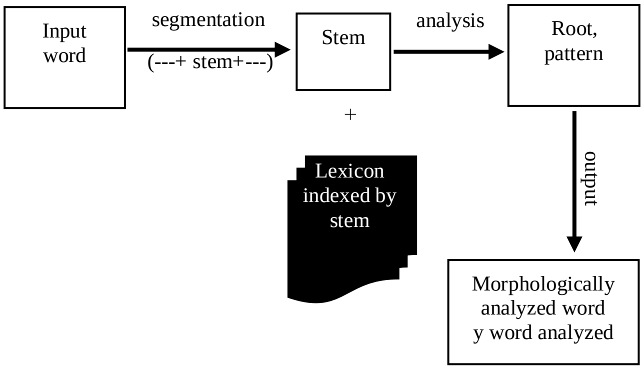

Fig. 5. Stems based on the Root-Pattern Morphology Process.

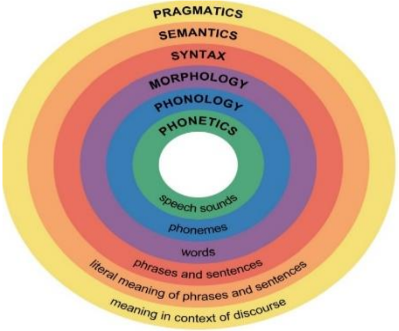

Fig. 6. Linguistic Levels [40].

Standard Arabic Language Morphological Analysis (SALMA) tools [7, 41] fall under this morphological approach. They include SALMA–Tagger, SALMA–ABCLexicon, and SALMA–Tag Set. AlKhalil morphological analyzer [30, 42] also depends on this morphology.

3) Lexeme-based morphology: Typically, lexemes differ only in inflection and cliticization (*الملحقات مثل: أل التعريف وحروف الجر المتصلة كالباء والكاف*). To put it simply, more than one word can be formed from one lexeme. For example, the lexeme *(*bayt*)* *بَيْت* includes *(*bayt*)* *بَيْت* , *(*lilbayt*)* *لِلبَيْت*, , and (buyuwt*)* *بُيُوت*. Therefore, the lexeme is not equivalent to a word in any language. It is considered an important abstraction used in linguistic morphology, and is the smallest part of the lexicon that has meaning (or semantic content). Additionally, a lexeme has a morphological form and syntactic category [2].

The claim that the stem is a morphological part with greater relevance to the lexeme is the premise underpinning lexeme-based morphology. This methodology depends on the crucial information of the stem, which must be extracted from the word in the right way. Soudi et al. [43] develop a lexeme-based morphology and present an Arabic version of a morphology rule compiled in the MORPHE tool (MORPHE is a general computational engine that works based on transformational rules and a discrimination hierarchy which must be constructed for each language).

In the lexeme-based methodology, the primary representation is made for the stem (including all operations on the stem, such as transformational rules applied to a stem to handle stem variation issues in several contexts of prefixes and/or suffixes). In other words, this methodology adopts a computational implementation of a non-sub-fragmented lexicon. Thus, this methodology differs from the root-pattern methodology, which gives equal consideration and separate lexicons to each constituent of a word (i.e., sub-lexicons for the root, for the pattern, and for vocalization) [5].

Many works on Arabic morphological analyzers adopt this methodology. Among these works are the following: a) a prototype lacking broad coverage, such as the MORPHE tool [43, 44]; and b) large-scale systems such as:

- ElixirFM [22], which reused the Buckwalter lexicon [1, 28].
- MAGEAD [21, 45] CALIMA, both of which handle Arabic dialects (MAGEAD entirely manually designed while CALIMA manually verified the annotated data lexicon using several computational techniques). There are three versions of CALIMA: CALIMAEGY [25], CALIMAGLF [26], and CALIMAstar [27]. Respectively, these cover Egyptian Arabic, Gulf Arabic, and all variants of MSA and Arabic dialects.
- AL-MORGEANA (abbreviated to ALMOR) [23], which extends the BAMA morphological databases with the lexeme and feature keys that are used in the analysis. For example, ALMOR uses the BAMA lexicon but changes the mode of analysis to produce a lexeme-and-feature format as output, rather than the stem-and-affix format, which is the Buckwalter output. It is important to mention here that ALMOR is the analyzer used in the MADA [46] tool. In addition, the new version of MADA is called MADAMIRA [47]. It is a Java NLP tool combining MADA with a shallow syntactic parser called AMIRA [48].
- AraComLex [24], which is based on the MSA lexical database[1], was specifically constructed for this purpose using a corpus of more than one million words.

4) Syllable-based morphology: Most syllable-based morphology work has been performed on European languages such as German, English, and Italian. Cahill [49] asserts the possibility of analyzing the Semitic languages using syllable-based morphology in a way that is not significantly different from that applied to European languages.

However, to our knowledge, there have been no attempts to build an Arabic morphological analyzer adopting this morphology to substantiate or reject this claim.

B. Data-Driven Lexicon-Based Approach

Machine learning techniques underpin these morphologies. These techniques are fast and do not require extensive linguistic knowledge because they depend on the annotated or unannotated corpus used in the training stage. Dinh et al. [50] claim that doubts could be raised around purely data-driven systems (which do not possess any linguistic base), but they are based on a hybrid. The new techniques prove this claim to be untrue. Recently, many supervised and unsupervised learning techniques have proved valuable in this area, as we will demonstrate in the two following subsections. Thus, we predict a promising future for these morphologies.

1) Supervised learning morphology: This approach attempts to infer parameter values from labeled resources without linguistic expertise about data. Supervised learning resources involve lexica of affixes and pairs of inflected words with their roots [51].

Supervised approaches are not famous in the domain of nonconcatenative morphology acquisition. These approaches require a massive lexicon in the training stage to achieve high precision. Some researchers take pride in their ability to avoid these massive lexica, but the disadvantages can be seen in their results, which have many limitations and are therefore not

[1] http://arabiconly.com/aracomlex/form_nominals.php

highly precise in general. However, this is the reality of any new technique. This method will become more promising as more annotated data becomes available.

The existing literature on Arabic morphology that uses this approach to identify Arabic roots is limited. There are two types which adopt some supervised learning: a) learning that is based on pre-existing dictionaries using Hidden Markov Models (HMM) [12] or neural network (NN) models [52], and b) learning that only uses rule constraints [10] or multi-class classifier models [11].

2) Unsupervised learning morphology: Unsupervised learning morphology, in essence, is the process of acquiring intra-word structures and the rules by which they merge to generate word forms [16]. In other words, morphology is induced without prior knowledge, based on training that uses large volumes of unannotated data, without supplying an example of the expected output. This research field began in the mid-1990s and continues today. Researchers consider unsupervised approaches attractive because of the large quantities of unlabelled data available on the Internet [15]. In recent years, unsupervised learning of concatenative language morphology (e.g., stem+affix morphology) has received more attention than nonconcatenative language morphology (e.g., root and pattern morphology) [53].

There are few studies in this field, but they vary according to the objectives of their algorithms. Some aim to learn segmentation [13-15], which means transforming a given word into its stem and affix(es), whereas others aim to learn lexica and patterns [16, 17], which means providing a list of the patterns and assigning each pattern the lexicon information related to all stems belonging to it.

In a significant contribution to this field of research, Khaliq and Carroll [18, 19] have built a morphological analyzer based on roots and patterns induced from the lexicon, based on learning from an unannotated corpus rather than linguistic rules, as noted in the section of this paper dealing with root-pattern morphology. This analyzer achieved good accuracy with root extraction, achieving 94% after many iterative reinforcement stages.

V. Discussion

As shown in the previous survey section, there are multiple morphological analyzers, with varying accuracy and features. No analyzer provides perfect performance, and none has been adopted as standard. Therefore, choosing one of these existing analyzers is difficult and represents a challenge in NLP tasks.

In this section, we compare the analyzers available for public use. Most relevant morphological analyzers achieved acceptable results (according to their developers) but were not available for reuse or evaluation.

To the best of our knowledge, the most recent and efficient morphological analyzers to achieve good accuracies for Arabic morphology are AlKhalil, AraComLex, and ALMOR. ALMOR is no longer available for download. It was distributed as part of MADA Distribution from Columbia University. A new version of MADA, called MADAMIRA, is now available. MADAMIRA is a morphological analyzer and a POS tagger (i.e., MADAMIRA operates *within* a word context while AlKhalil and AraComLex operate *outside* of a word context). Table IV compares these analyzers according to various attributes.

TABLE. IV. Comparison of Available Arabic Morphological Analyzers

Attribute name	ALMOR	AraComLex	AlKhalil
Different configurations (pre-/post-processing)	Yes	No	Yes
Performance metric	Precision & recall	Coverage rate	Coverage rate
Running through	Application Programming Interface (API)	API	Graphical User Interface (GUI)
Directionality	Analysis and generation	Analysis only	Analysis only
Expected input	Text only (works on diacritized text, but no consideration of these diacritics)	Non-vowelized word (does not work on a diacritized word)	Fully or partially diacritized text
Accuracy (sample of 50 words) [8]	88%	56%	90%
Engine	Code-based (Java and Perl languages)	Finite-state machinery	Code-based (Java and Perl languages)
Input format	Word or text	Just one word per query	Word or text
Output format	Text of (feature: value) pairs	In one line, separate between features by (+)	Table (like CSV file) of features
Tag set	About 36 basic tag sets	About 14 tag sets	About 118 tag sets
Transliteration schemes for results	Buckwalter	UTF-8	UTF-8
Last version	As a part of MADAMIRA 2014	AraComLex 2.1 (2018)	AlKhalil 2 (2016)

VI. Conclusion

Many scientific studies discuss Arabic morphological analysis techniques, reviews, and analyzer tools, but they lack a specific and accurate classification of traditional and recent methods. In fact, the linguistic lexicon-based and data-driven lexicon-based approaches are the two main approaches for morphological analysis techniques. All techniques found in the existing literature align with these approaches. This classification can guide us towards standard Arabic morphological analysis techniques.

A linguistic lexicon-based approach depends on solid linguistic rules derived from the lexicon. It covers four types of morphology based on analysis process terms: root-pattern, stem, lexeme, and syllable. The data-driven lexicon-based approach depends on an annotated or unannotated corpus to undergo a training process on data, in order to collect rules which are then used to output word forms.

Most of the systems mentioned in this survey are not available for public use. We highlighted the most recent available systems, and compared them on various aspects.

It is important that future research in Arabic morphological analysis investigate the following issues:

- Developing a gold standard Arabic corpus that can be used to compare morphological analysis systems.
- Developing a large annotated Arabic corpus to be used in the promising data-driven approach morphologies.
- Developing a hybrid approach using linguistic and data-driven morphologies to merge the advantages and strengths of these two approaches.
- Using a unified standard of performance metrics in evaluation systems to compare approaches.
- Building a multicomponent toolkit for Arabic morphological analyzers to integrate these analyzers' results and choose the one with the best performance.
- Building a multicomponent toolkit for Arabic morphological analyzers in order to facilitate a selection process for the one that best fits the researcher's/user's needs.

References

[1] T. Buckwalter, Buckwalter Arabic Morphological Analyzer Version 1.0. Philadelphia: Linguistic Data Consortium, 2002.

[2] N. Y. Habash, Introduction to Arabic Natural Language Processing. San Rafael: Morgan & Claypool, 2010.

[3] H. Alkalifah, مقدمة في المعالجة الطبيعية للغة العربية. Saudi Arabia: King Saud University Press, 2014.

[4] I. A. Al‑Sughaiyer and I. A. Al‑Kharashi, "Arabic morphological analysis techniques: A comprehensive survey," J. Am. Soc. Inf. Sci. Technol., vol. 55, pp. 189‑213, February 2004.

[5] A. Soudi, G. Neumann, and A. van den Bosch, "Arabic computational morphology: Knowledge-based and empirical methods," in Arabic Computational Morphology, A. Soudi, A. Bosch, and G. Neumann, Eds. Dordrecht: Springer, 2007, pp. 3–14.

[6] J. Dichy and A. Farghaly, "Roots & patterns vs. stems plus grammar-lexis specifications: On what basis should a multilingual lexical database centred on Arabic be built," in The MT-Summit IX Workshop on Machine Translation for Semitic Languages. New Orleans, 2003.

[7] M. Sawalha, E. Atwell, and M. A. M. Abushariah, "SALMA: Standard Arabic language morphological analysis," in 2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA), 2013, pp. 1–6.

[8] A. Alosaimy and E. Atwell, "Tagging classical arabic text using available morphological analysers and part of speech taggers," J. Lang. Technol. Comput. Linguist., vol. 32, pp. 1–26, December 2017.

[9] I. Guellil, H. Saâdane, F. Azouaou, B. Gueni, and D. Nouvel, "Arabic natural language processing: An overview," J. King Saud Univ. Comput. Inf. Sci., doi:10.1016/j.jksuci.2019.02.006, February 2019.

[10] K. Elghamry, "A constraint-based algorithm for the identification of Arabic roots," in Proceedings of the Midwest Computational Linguistics Colloquium. Bloomington: University of Indiana, 2005.

[11] E. Daya, D. Roth, and S. Wintner, "Identifying semitic roots: Machine learning with linguistic constraints," Comput. Linguist., vol. 34, pp. 429–448, September 2008.

[12] A. Boudlal, R. Belahbib, A. Lakhouaja, A. Mazroui, A. Meziane, and M. Bebah, "A markovian approach for Arabic root extraction," Int. Arab J. Inf. Technol., vol. 8, pp. 91–98, January 2011.

[13] P. Rodrigues and D. Cavar, "Learning Arabic morphology using statistical constraint-satisfaction models," Amst. Stud. Theory Hist. Linguist. Sci. 4, vol. 289, pp. 63–75, January 2007.

[14] B. Snyder and R. Barzilay, "Unsupervised multilingual learning for morphological segmentation," in Proceedings of ACL-08: HLT. Ohio: Association for Computational Linguistics, 2008, pp. 737–745.

[15] H. Poon, C. Cherry, and K. Toutanova, "Unsupervised morphological segmentation with log-linear models," in Annual Conference of the North American Chapter of the ACL. Boulder, Colorado: Association for Computational Linguistics, 2009, pp. 209–217.

[16] J. A. Botha and P. Blunsom, Adaptor Grammars for Learning Non-Concatenative Morphology. Stroudsburg: Association for Computational Linguistics, 2013.

[17] M. Fullwood and T. O'Donnell, "Learning non-concatenative morphology," in Proceedings of the 4th Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL). Sofia: Association for Computational Linguistics, 2013, pp. 21–27.

[18] B. Khaliq and J. Carroll, "Unsupervised induction of arabic root and pattern lexicons using machine learning," in Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013. Shumen: INCOMA Ltd., 2013, pp. 350–356.

[19] B. Khaliq and J. Carroll, "Induction of root and pattern lexicon for unsupervised morphological analysis of Arabic," in Proceedings of the 6th International Joint Conference on Natural Language Processing. Japan: Asian Federation of Natural Language Processing, 2013, pp. 1012–1016.

[20] M. Gridach and N. Chenfour, "Developing a new system for Arabic morphological analysis and generation," in Proceedings 2nd Workshop on South Southeast Asian Natural Language Processing (WSSANLP). Thailand: IJCNLP, 2011, pp. 52–57.

[21] N. Habash and O. Rambow, "MAGEAD: A morphological analyzer and generator for the Arabic dialects," in Proceedings of 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics. Sydney: Association for Computational Linguistics, 2006, pp. 681–688.

[22] O. Smrz, "ElixirFM – implementation of functional arabic morphology," in Proceedings of 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources. Prague: Association for Computational Linguistics, 2007, pp. 1–8.

[23] N. Habash, "Arabic morphological representations for machine translation," in Arabic Computational Morphology: Knowledge-Based and Empirical Methods, A. Soudi, A. van den Bosch, and G. Neumann, Eds. Dordrecht: Springer Netherlands, 2007, pp. 263–285.

[24] M. Attia, P. Pecina, A. Toral, L. Tounsi, and J. van Genabith, "An open-source finite state morphological transducer for modern standard Arabic," in Proceedings of 9th International Workshop on Finite State Methods and Natural Language Processing. France: Association for Computational Linguistics, 2011, pp. 125–133.

[25] N. Habash, R. Eskander, and A. Hawwari, "A morphological analyzer for Egyptian Arabic," in Proceedimgs of 12th Meeting of the Special Interest Group on Computational Morphology and Phonology. Canada: Association for Computational Linguistics, 2012, pp. 1–9.

[26] S. Khalifa, S. Hassan, and N. Habash, "A morphological analyzer for Gulf Arabic verbs," in Proceedings of Third Arabic Natural Language Processing Workshop. Stroudsburg: Association for Computational Linguistics, 2017, pp. 35–45.

[27] D. Taji, S. Khalifa, O. Obeid, F. Eryani, and N. Habash, "An arabic morphological analyzer and generator with copious features," in Proceedings of 15th Workshop on Computational Research in Phonetics, Phonology, and Morphology. Belgium: Association for Computational Linguistics, 2018, pp. 140–150.

[28] T. Buckwalter, Buckwalter Arabic Morphological Analyzer: Version 2.0. Philadelphia: Linguistic Data Consortium, 2004.

[29] M. Maamouri, D. Graff, B. Bouziri, S. Krouna, A. Bies, and S. Kulick, LDC Standard Arabic Morphological Analyzer (SAMA), Version 3.1. Philadelphia: Linguistic Data Consortium, 2010.

[30] M. Boudchiche, A. Mazroui, M. O. A. O. Bebah, A. Lakhouaja, and A. Boudlal, "AlKhalil morpho Sys 2: A robust Arabic morpho-syntactic analyzer," J. King Saud Univ. Comput. Inf. Sci., vol. 29, pp. 141–146, April 2017.

[31] J. J. McCarthy, Formal Problems in Semitic Phonology and Morphology, Ph.D. Thesis. Cambridge: Massachusetts Institute of Technology, 1979.

[32] J. J. McCarthy, "A prosodic theory of nonconcatenative morphology," Linguist. Inq., vol. 12, pp. 373–418, January 1981.

[33] J. Dichy and A. Farghaly, "Grammar-lexis relations in the computational morphology of Arabic," in Arabic Computational Morphology, A. Soudi, A. Bosch, and G. Neumann, Eds. Dordrecht: Springer, 2007, pp. 115–140.

[34] Y. Hlal, "Morphology and syntax of the Arabic language," in Computers and the Arabic language, A.M. Pierre, Ed. Bristol: Taylor & Francis/Hemisphere, 1990, pp. 201–207.

[35] N. H. Hegazi and A. A. El-Sharkawi, "An approach to a computerized lexical analyzer for natural Arabic text," in Proceedings of the Arabic Language Conference. Kuwait, 1985.

[36] N. H. Hegazi and A. A. El-Sharkawi, "Natural Arabic language processing," in Proceedings of the National Computer Conference. Riyadh, 1986, pp. 10–15–11–10–15–17.

[37] K. R. Beesley, "Finite-state morphological analysis and generation of Arabic at Xerox Research: Status and plans in 2001," in ACL Workshop on Arabic Language Processing: Status and Perspective, 2001, pp. 1–8.

[38] A. Farghaly and J. Senellart, "Intuitive coding of the Arabic lexicon," in SYSTRAN, MT, Summit IX Workshop, Machine Translation for Semitic Languages: Issues and Approaches, Tuesday September. New Orleans: Citeseer, 2003.

[39] T. Buckwalter, "Issues in Arabic morphological analysis," in Arabic Computational Morphology, N. Ide, J. Veronis, A. Soudi, A. van den Bosch, and G. Neumann, Eds. Netherlands: Springer, 2007, pp. 23–41.

[40] Wikimedia Commons, Major levels of linguistic structure. 2019. Available: https://commons.wikimedia.org/wiki/File:Major_levels_of_linguistic _structure.svg.

[41] M. Sawalha and E. S. Atwell, "ب ناء ف ي وال صرف ال نحو ق واعد ت وظ يف ال عرب ية ل لغة صرف ي محلل (Adapting language grammar rules for building a morphological analyzer for Arabic text)," in Proceedings of ALECSO Arab League Educational Cultural and Scientific Organization workshop on Arabic morphological analysis. Damascus, 2009.

[42] A. Boudlal, A. Lakhouaja, A. Mazroui, A. Meziane, M. Bebah, and M. Shoul, "Alkhalil morpho sys1: A morphosyntactic analysis system for arabic texts," in International Arab Conference on Information Technology. Benghazi, 2010, pp. 1–6.

[43] A. Soudi, V. Cavalli-Sforza, and A. Jamari, "A computational lexeme-based treatment of Arabic morphology," in Proceedings of the Arabic Natural Language Processing Workshop, Conference of the Association for Computational Linguistics (ACL 2001), 2001, pp. 50–57.

[44] V. Cavalli-Sforza, A. Soudi, and T. Mitamura, "Arabic morphology generation using a concatenative strategy," in Proceedings of the 1st North American chapter of the Association for Computational Linguistics Conference. Stroudsburg: Association for Computational Linguistics, 2000, pp. 86–93.

[45] N. Habash, O. Rambow, and G. Kiraz, "Morphological analysis and generation for Arabic dialects," in Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages. Ann Arbor: Association for Computational Linguistics, 2005, pp. 17–24.

[46] N. Habash, O. Rambow, and R. Roth, "MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization," in Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR). Egypt, 2009, pp. 102–109.

[47] A. Pasha, M. Al-Badrashiny, M. Diab, A. E. Kholy, R. Eskander, N. Habash, M. Pooleery, O. Rambow, and R. Roth, "MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic," in Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC-2014). Reykjavik: European Language Resources Association (ELRA), 2014, pp. 1094–1101.

[48] M. Diab, "Second generation AMIRA tools for Arabic processing: Fast and robust tokenization, POS tagging, and base phrase chunking," in 2nd International Conference on Arabic Language Resources and Tools, 2009.

[49] L. Cahill, "A syllable based account of Arabic morphology," in Arabic Computational Morphology, A. Soudi, G. Neumann, and A. van den Bosch, Eds. Netherlands: Springer, 2007, pp. 45–66.

[50] D. Dinh, H. Kiem, and E. Hovy, "BTL: A hybrid model for English-vietnamese machine translation," in Proceedings of the IXth MT Summit. New Orleans, 2003, pp. 87–94.

[51] J. A. Botha, Probabilistic modelling of morphologically rich languages. 2015. Available: https://ui.adsabs.harvard.edu/\#abs/2015arXiv150804271B.

[52] H. Al-Serhan and A. Ayesh, "A triliteral word roots extraction using neural network for Arabic," in 2006 International Conference on Computer Engineering and Systems. Cairo, Egypt: IEEE, 2006, pp. 436–440.

[53] H. Hammarström and L. Borin, "Unsupervised learning of morphology," Comput. Linguist., vol. 37, pp. 309–350, June 2011.