An Integrated, Bidirectional Pronunciation, Morphology, and Diacritics Finite-State System

Maha Alkhairy
University of Massachusetts Amherst,
Amherst, MA, USA

Afshan Jafri
King Saud University,
Riyadh, Saudi Arabia

Adam Cooper
Northeastern University,
Boston, MA, USA

***Abstract*---A bidirectional phonetizer, morphologizer, and diacritizer pipeline (FSPMD) for modern standard Arabic (MSA) that integrated pronunciation, concatenative and templatic morphology, and diacritization were developed. Grammar and segmental phonology rules were applied in the forward direction to ensure the order of the proper rules, which were supplemented with special backward direction rules. The FSPMD comprises bidirectional finite-state transducers (FSTs) consisting of an ordered composition of FSTs, unordered parallel FSTs, unioned FSTs, and for validity, finite-state acceptors. The FSPMD has unique, innovative features and can be used as an integrated pipeline or standalone phonetizer (FSAP), morphologizer (FSAM), or diacritizer (FSAD). As the system is bidirectional, it can be used in forward (generation, synthesis) and backward (analysis, decomposition) directions and can be integrated into systems such as automatic speech recognition (ASR) and language learning tools. The FSPMD is rule-based and avoids stem listings for morphology or pronunciation dictionaries, which makes it scalable and generalizable to similar languages. The FSPMD models authentic rules, including fine granularity and nuances, such as rewrite and morphophonemic rules, subcategory identification and utilization, such as irregular verbs. FSAP performance regarding text from the Tashkeela corpus and Wikipedia demonstrated that the pronunciation system can accurately pronounce all text and words, with the only errors related to foreign words and misspellings, which were out of the system's scope. FSAM and FSAD coverage and accuracy were evaluated using the Tashkeela corpus and a gold standard derived from its intersection with the UD_PADT treebank. The coverage of extraction of root and properties from words is 82%. Accuracy results are roots computed from a word (92%), words generated from a root (100%), non-root properties (97%), and diacritization (84%). FSAM non-root results matched and/or surpassed those from MADAMIRA; however, root result comparisons were not conducted because of the concatenative nature of publicly available morphologizers.**

Keywords---Computational linguistics; phonology; morphology; modern standard Arabic; diacritization; text-to-speech; language learning tools

I. Introduction

Natural language processing technologies, such as automatic speech recognition (ASR) systems, rely on pronunciation dictionaries that provide word listings and corresponding phone pronunciations for both training and recognition. In the ASR training phase, an orthographic text passage is transformed into its phonetic transcription (pronunciation), which comprises a sequence of phonemes or phones. In the recognition phase, the phonetic transcription is transformed into its associated word sequence orthographic text (1).

Because effective ASR requires a large dictionary that lists the words and their pronunciation, the system perplexity increases, and the accuracy declines. One possibility to reduce the size for languages that have deep orthography and highly irregular mapping, such as English, French, and Danish, is to list the affix and stem pronunciations rather than the words; however, this requires a concatenative morphologizer (prefix, stem, and suffix). At the other end of the spectrum, languages that have shallow orthography (phonetic languages) and a one-to-one correspondence between the letters and phonemes, such as Finnish and Turkish, only require a very simple dictionary of letters and the associated phonemes. Because middle-spectrum languages, such as Russian, German, and Spanish, have complex correspondences between the letters and pronunciation, they are not amenable to rules [1] (2).

However, other languages in the middle spectrum, such as Arabic and Hebrew, do have rules-based pronunciation, which means that rule-based transducers between the words and their phonemic transcriptions could resolve the need for large pronunciation dictionaries. If a transducer is bidirectional, it could be used for both the ASR training and recognition phases. In the backward direction, in which a phonetic sequence is mapped to an orthographic text, a word acceptor based on morphemes is needed to avoid invalid words, the use of which could avoid large word lists that would require an integrated phonetizer and morphologizer. While building a more efficient, accurate ASR could be a valid motivation for designing a bidirectional rule-based pronunciation transducer (phonetizer), such automata would be useful in its own right as it could be applied to other domains, such as text-to-speech synthesis, that require the identification of both suprasegmental features and segmental phonology. In addition to designing and constructing a bidirectional rule-based phonologizer that maps the relationships between orthographic text and its phonetic transcription, this study also developed a bidirectional concatenative (prefixes, stems, suffixes) and templatic (roots, patterns) morphologizer that generates words from morphemes and decomposes words into morphemes. Templatic morphology is another major feature not present in languages such as English.

Besides being important in its own right in multiple technologies, there are two main reasons a morphologizer is required in phonetizers: as an acceptor to filter out invalid words without the need for a large word list and as a phonological morphology-based rules regulator to determine whether a word is a noun or a verb to reduce pronunciation ambiguity.

In languages such as Arabic and Hebrew, the written script

[1] https://en.wikipedia.org/wiki/Orthography

is either undiacritized or diacritized. As phonetizers and morphologizers generally require diacritized text, a diacritizer is also needed to complete the pipeline. In the system developed in this study, the diacritizer is independent of the phonetizer and morphologizer; however, it uses the same constructs as the morphologizer.

The links between morphology, phonology, morphology, and diacritics result in an integrated system. Therefore, this study proposes a method and structure that can exploit the innate grammar of a language. While Arabic is used as the demonstration language in this study, the proposed method can be equally applied to other such languages. Semitic languages, such as Arabic and Hebrew, have form-based morphology and rules-based pronunciation and are usually undiacritically written (3).

An integrated bidirectional finite-state (FSPMD) system was designed and constructed that incorporates a phonetizer (FSAP), a morphologizer (FSAM) that can work with both diacritized and undiacritized text, and a diacritizer (FSAD). The various linguistic and segmental phonological rules were fully incorporated and finite-state transducers (FST) were solely employed to build the system; therefore, it was not necessary to include the additional features found in other systems, such as two-level finite states and flags.

In the forward direction, the FSAP transforms a diacritized passage into its corresponding pronunciation; FSAM generates words from the patterns, roots, prefixes, and suffixes; and FSAD inserts diacritics into undiacritized words. In the backward direction, the FSAP produces a text passage for a sequence of phones; FSAM decomposes the words into their prefixes, patterns, roots, suffixes, and linguistic features, such as gender and part of speech; and FSAD strips the diacritics from diacritized words. The FSAM can also work as an acceptor for morphologically valid words

The FSPMD, therefore, connects phonetic transcriptions and diacritization to morphology to create a tight system that among other constraints, limits words corresponding to a pronunciation when there is an absence of listings. The FSPMD's bidirectional pipeline synthesizes words from affixes, patterns, and roots, computes the pronunciation from texts based on segmental phonological units, and diacritizes words, and in the opposite direction, analyzes a word into its morphemes, transforms pronunciation into text, and undiacritizes words.

This study excluded end-of-word diacritics as these are governed by syntactic rules that are unable to be formulated as regular expressions that can be realized as finite-state transducers (FSTs), that is, they require higher-order formal language, such as context-sensitive grammar. Particular attention was paid to authentic grammar rules and many details and nuances were incorporated, such as the effects of text marks in phonology and the inclusion of rewritten rules for the morphological orthography.

The remainder of this paper is organized as follows. Section II details the problem and the integrated architecture, Section III presents the phonetizer, Section IV presents the morphologizer, and Section V presents the diacritizer. Appendix A presents the phonetizer and morphologizer literature reviews. Appendix B presents additional phonetizer and morphologizer evaluation results. Transliterations of Arabic to Roman letters were not used to reduce confusion. Supplementary material 1 is a compendium for Arabic orthography, phonetics, and morphology and provides details not necessary to understand the main paper. Supplementary material 2 gives the finite-state automata and their earlier usages in linguistics and phonology and a related literature survey.

II. Problem Formulation and Integrated Architecture

An orthographic text is a sequence of characters that make up words and marks such as tabs, text beginnings, and commas. An Arabic word comprises a sequence of graphemes that include alphabetic and non-alphabetic letters and diacritics. In addition to syntax, which governs the end-of-word diacritics, diacritized text has all the diacritics mandated by the associated spelling and morphological rules; however, undiacritized texts only have Shaddah and sometimes Tanween and Sukoon as diacritics. Because syntactic processing may not be realized by FSTs, in this study, the undiacritized Arabic texts also included end-of-word diacritics. A phonetic transcription (pronunciation) is a sequence of phones consisting of phonemes and fermatas that represent pauses of various durations and continuation. These graphemes, marks, phonemes, and fermatas are described in Supplementary material I along with the mappings between the marks and fermatas. If there is more than one realization based on the context, the phonetic transcriptions may also contain an allophone variation of a phoneme.

Phonetically transcribing an orthographic text produces phonemes and fermatas that depend on both graphemes and marks, that is, the transformation of phonetic transcription to orthographic text depends on both phonemes and fermatas, which is why the fermata plays a more important transformation role in Arabic than in other languages.

The proposed system has various FSTs to transform the input sequences into output sequences. The transducer also acts as an acceptor, which produces a FALSE notification if the input is not valid according to the transducer rules. As the forward direction FST was designed to utilize linguistic grammar rules, depending on the particular FST, the forward direction could be either generation/synthesis or analysis/decomposition. Bidirectional FSTs were employed to enable generation as well as analysis using analysis rules. This was made possible by the method used to construct the FSTs, which incorporated some unidirectional limiting rules exceptions and some additional rules for the opposite direction only. The rules were written as regular expressions, which were then transformed into non-probabilistic FSTs using the open-source Foma compiler (4).

The FSAP mapped between the diacritized texts and the phonetic transcriptions, with the forward direction producing the pronunciation from the diacritized orthographic texts, which was represented by International Pronunciation Association (IPA) and fermata symbols. Because the phonetizer can realize segmental rules, it did not embody phonological suprasegmentals, such as syllables, stress, or intonation. The FSAD mapped between the diacritized words and the corresponding undiacritized words, with the forward direction generating diacritized words from an undiacritized word. The FSAM mapped between the words and associated morphemes (pattern, root, prefix, suffix), with the backward direction generating words from the morphemes. While these FSTs were constructed from multiple FST components, as described in the various sections, each can also be used as a standalone system.

The three FSTs were integrated into a pipeline structure to compute the phonetic transcriptions for the undiacritized (or diacritized) texts or to decompose an undiacritized word into its morphemes. The integrated pipeline system's (Fig. 1) forward and backward direction functions are detailed in the following.

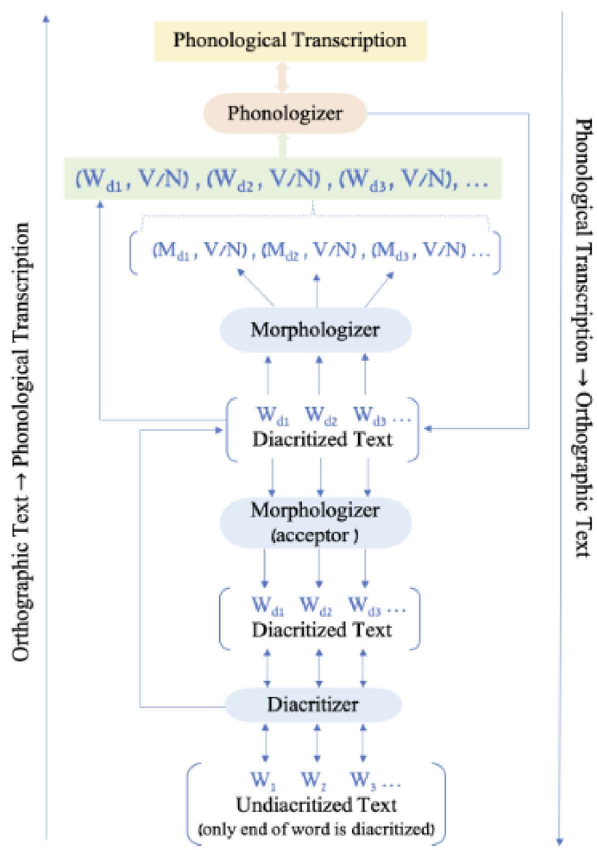

Fig. 1. Bidirectional integrated system mapping between the orthographic undiacritized text and the phonetic transcription (pronunciation). M refers to morphemes (prefix, root, suffix, and pattern), N refers to nouns, and V refers to verbs, as these affect the pronunciation.

In the forward direction, the system takes four steps to produce a phonetic transcription of an undiacritized text: (1) the diacritizer maps a given sequence of undiacritized words into diacritized words; (2) the morphologizer extracts the morphemes (prefix, root, suffix, pattern) and morphological properties (noun or verb) of each word in the sequence; (3) the system then concatenates the morphologizer and diacritizer results to produce a sequence of diacritized words and the associated morphological properties; and (4) finally, the concatenation sequence is input into the phonetizer, which produces the phone sequences.

In the backward direction, the pipeline system takes three steps to produce the undiacritized and diacritized orthographic texts to be input into the phonetic transcription: (1) the phonetizer used in the reverse direction produces multiple diacritized word sequences from the phone sequence; (2) the morphologizer, which is used as an acceptor, filters out the morphologically incorrect diacritized word sequences and produces a set of valid diacritized texts; and (3) finally, the diacritizer used in the reverse direction undiacritizes the diacritized word sequences to produce an undiacritized text.

III. Phonetic Transducer (FSAP)

The phonetic transducer was constructed using a combination of an ordered composition of FSTs and unordered parallel FSTs, with the finite-state automata (FSAs) being the acceptors to define the grapheme, marker, phoneme, and fermata subsets. The unordered FSTs embodied non-contextual rules and the ordered FSTs realized the contextual grammar, which required a thorough and precise ordering of the rules to ensure precise results.

In contrast to the current approaches outlined in Appendix A, FSTs rather than procedural methods were utilized, which avoided the need for a two-level FST that makes multiple transformations in favor of a single level; syllabic structures to compensate for shortcomings in the realization of the rule; the incorporation of context-dependent rules in a specified order, which is generally ignored; and the realization of all MSA rules, including those related to text markers, which can significantly effect phonetization.

Geminated consonants and long vowels were also considered phonemes in their own right. Previous studies(5) have tended to consider gemination by doubling a singleton consonant or mapping it into its singleton version, which has been shown to be phonetically inaccurate, as demonstrated by geminated plosives that have a single voice onset time and release. Similarly, long vowels have previously been dealt with by doubling the short version; however, the spectral characteristics of long vowels are noticeably different from their short vowel counterparts(5).

Rather than applying simplifications to these rules, special attention was paid to the precise complex rules regarding Wasl (ب ف ك و) and Illah (ا و ي) characters, including those in the Sukoon (ـْ) context. Rules regarding the noun versus verb factors in the pronunciation of diacritized words were also considered as diacritized words still have some pronunciation ambiguity in Alif Wasl (ٱ).

The following subsections present the contextual and non-contextual orthography: phoneme mappings and contextual phonemes; allophone rules and backward phonemes; and orthography results. The section ends with the phonetic transducer evaluations.

A. Orthography: Phoneme Mappings

The orthographic and phonetic representation mappings were divided into non-contextual and contextual rules. The first two subsections present the non-contextual diacritics; vowel and character and phoneme mappings; and the latter three sections present the contextual letters; the phoneme mappings and pronunciation rules governed by a word's part of speech, such as noun or verb. The markings affecting the way the letter/diacritic is pronounced are also detailed in the following rules.

1) Diacritics: vowels mappings: A Harakat (ـَـُـِ) grapheme can be mapped to its corresponding short vowel (/a/, /u/, and /i/) in all contexts; however, it is not pronounced (/•/) when it precedes its corresponding 'vowel characters' (ي), (و), (ا ى). A Tanween (ـًـٌـٍ) grapheme only diacritizes the end of words and generally maps to its corresponding vowel when followed by /n/ (/a/ / n/, /u/ /n/, /i/ /n/); however, Tanween is not pronounced (/•/) if it diacritizes a word that ends the sentence.

2) Non-contextual characters: phoneme mapping: The diacritic ّ (shaddah) causes a gemination in the sound of the preceding letter it diacritizes, and ـْ maps to a zero duration pause /Φ/. The Hamza set (ؤ ئ إ أ ء) is pronounced as /ʔ/, and the other mappings are as follows: (ٰ : /aː/), (آ: /ʔaː/), (ب: /b/), (ت: /t/), (ث: /θ/), (ج: /d͡ʒ/), (ح: / ħ/), (خ: /χ/), (د: /d/), (ذ: /ð/), (ر: /r/), (ز: /z/), (س: /s/), (ش: /ʃ/), (ص: /sˤ/), (ض: /dˤ//), (ط: / tˤ /), (ظ: /ðˤ/), (ع: /ʕ/), (غ: /ʁ/), (ف: /f/), (ق: /q/), (ك: /k/), (م: /m/), (ن: /n/), (ه: /h/).

3) Contextual letters: phoneme mapping: The context determines the pronunciation for (ة, ى, و, ي, ا, ل). The rules are presented from the simplest to the most complex for the Wasl letters (ك ل ت ب و), the Alef Wasl (ا), the definite article (ال), and the other rules. The following are the contextual mappings for the letters and letter combinations.

The letter ة is always at the end of a word followed by a Haraka and maps to /t/ or /h/. ة : /t/ if it is in a word in the middle/start of a sentence and not followed by ـْ; ة : /h/ if followed by ـْ or in a word that ends a sentence. The letter ى always ends a word and maps to /aː/ or /a/; ى maps to /a/ if it ends in a word that precedes a word that starts with ا, and ى, and /aː/ if it is in a word that ends a sentence or precedes a word that doesn't start with ا.

The letter و maps to /w/, /wː/ or /uː/. و : /w/ if followed by a diacritic other than ـْ and preceded by ـْ; و : /wː/ if followed by ـّ and preceded by ـْ ;, and و: /uː/ if preceded by ـُ. The letter ي maps to /j/, /jː/, or /iː/. ي: /j/ if preceded by ـَ or ـْ. ي and followed by a diacritic other than ـّ ; ي : /jː/ if preceded by ـَ or ـْ and followed by a diacritic other than ـّ; and ي: /iː/ if preceded by ـِ.

The letter ا maps to /ʔ/ or is not pronounced /•/. ا: /ʔ/ if it starts a word and is followed by a Harakah; ا:, and /•/ if it is preceded by a Wasl letter (ك ل ت ب و). The combination ـًا maps to /a/ if it ends a word, and maps to /aː/ if it is between letters.

The letter ل maps to /l/ or /ϵ/ (not pronounced) when it is part of the definite article (ال); otherwise, it is pronounced /l/. The letter combination ال (the definite article) maps to /ʔal/, /ʔa•/, /•l/, or /•/ (not pronounced), ال: /ʔal/ if it is followed by a Lunar letter; ال: /ʔa•/ if it is followed by a Solar letter, ال: /•l/ if it is preceded by a Wasl letter, and ال: /•/ if it is preceded by a Wasl letter and followed by a Solar letter. The pronunciation of ا and ال at the beginning of a word also depends on whether the word is a verb or a noun, as detailed in Subsection III-A4.

4) Noun and verb rules pertaining to Alif Wasl: When it occurs at the start of the word without diacritization and as part of the spelling, the pronunciation of Alif Wasl (ا) is ambiguous and requires knowledge of the word's part of speech, particularly whether it is a noun or verb. Specifically, the situations are as follows: (1) ا: /ʔa/ if ا is part of ال at the beginning of a verb and not a noun, that is, it is not treated like ال the definite article; for example, الْعَبْ: /ʔalʕab/ and الْجُمْ: /ʔald͡ʒum/; (2) : /ʔu/ if ا is at the beginning of a verb in which the third letter is diacritized with ـُ ; for example, اكْتُبْ: /ʔuktub/; (3) ا: /ʔi/ if ا is at the beginning of a noun and not part of the definite article (ال); for example, امْرُؤ: /ʔimruʔ/, اسم: /ʔism/, and ابْن: /ʔibn/.

B. Contextual Phoneme: Allophone Mappings

Multiple phoneme to allophone mappings exist and have several variations, two of the most prominent of which are described here. The first is pharyngealization, which produces a pharyngeal counterpart (if it exists) of a phoneme followed by a pharyngeal phoneme. The second is homorganic nasal place assimilation, which changes a nasal phoneme. For example, the alveolar nasal /n/ is pronounced as the bilabial nasal /m/ if it is followed by the voiced bilabial /b/ or the bilabial nasal /m/. Table I gives some examples of these rule occurrences.

C. Phonetic Transcription to Orthographic Text

The previous subsections presented the bidirectional mappings formulated in the forward direction. As mentioned, the FST's bidirectional nature allows the system to map from phonetic transcription to orthographic text. Some mappings, however, need to be explicitly expressed in the backward direction only. Table II provides a few examples generated by the phonetizer.

The rules that lead to the deletion of characters can interfere with the rules and cause an infinite loop, that is, no results. This can be resolved by including a special symbol to indicate deletion and by not applying the deletion in the reverse direction. Because some of the outputs in this direction were not valid words due to the deletions that occur in pronunciation and the lack of word lists, the produced words were input into the morphologizer in the analysis direction to treat the lack of analysis as a rejection.

D. FSAP Evaluation

In the absence of a pronunciation corpus with a sufficient number of examples to gain a numeric accuracy and recall evaluation, fully diacritized pronunciation examples, which were independently verified by a linguist, were produced to test the FSAP's scope and accuracy, with the performances being assessed based on: 1) individual words and small sentences with the associated pronunciation to test the specific context and check the inclusion and accuracy of all rules; 2) passages from Tashkeela(6) to test the ability to deal with multiple contexts at a time and to handle unknown words; and 3) examples from Wikipedia [2] [3] with the associated transcription to assess the validity of the system. The evaluation of the examples demonstrated that the pronunciation system was able to accurately pronounce all text and words, with the only errors being foreign words and misspellings, such as a missing Mad character, which was out of the system scope.

[2]https://en.wikipedia.org/wiki/Varieties_of_Arabic
[3]https://en.wikipedia.org/wiki/Arabic_phonology

Table I. Sample Allophonic Changes in Modern Standard Arabic

Word	Gloss	Phonemic	Allophonic	Change
استصلح	consider useful	/ʔis.tasˤ.la. ħa/	[ʔisˤ.tasˤ.la.ħa]	pharyngealization
فرعون	Pharaoh	/fir.ʕawn/	[firˤ.ʕawn]	pharyngealization
انبعث	regain one's strength and vividness	/ʔin.ba.ʕa.θa/	[ʔim.ba.ʕa.θa]	Homorganic nasal place assimilation
من بعد	after	/min baʕ.di/	[mim⌣baʕ.di]	Homorganic nasal place assimilation

Table II. Phonetic to Orthographic Transformation: Φ: Zero Duration Pause, μ: Short Duration Pause (such as the Breath Taken between each Word), ω: Medium Duration Pause, α: Long Duration Pause, ⌣: Continuation, •: Not Pronounced

Phonetic	Orthographic	Phonetic	Orthographic	Phonetic	Orthographic
/ma⌣ • smik/	مَا اسمِك	/ʔatːamarΦ/	التَّمَرْ	/bim • aː/	بِمَا
/ʔinbaʕaθa/	اِنبَعَث انبَعَث	/bariːʔ/	بَريء بَرِيء	/masʔuːlin/	مَسؤُولٍ
/min μ baʕdi/	مِن بَعدِ	/bima/	بِمَ	/ʔumːi/	أُمِّ
/bimaː/	بِما	/wa • stasˤlaħa/	وَاستَصلَحَ وَاِستَصلَحَ	/katabaː/	كَتَبا

1) Evaluation of the words and phrases: A rich listing of valid diacritized words and phrases was produced to test the edge cases. The following words were a test bed for Harakat: vowels and non-contextual graphemes; phonemes and contextual graphemes; phonemes; and words may have multiple pronunciations. Words and short sentences were then chosen that contained characters that had varied context pronunciation, specifically, the definite article (ال), Harakat (ـَ ـِ ـُ), Tanween (ـً ـٍ ـٌ), ta' marbutah (ة), alif (ا), lam (ل), waw (و), and ya (ي). Table XII gives the comprehensive evaluation of the phonotizer to ensure that all edge cases were tested. A comparison of the system outputs with the IPA transcription by a language specialist revealed the transducer's accuracy and coverage. Notice that phonotizer output symbols, such as zero duration pauses and deletions, were not present in the transcription. Table III presents the system output of various inputs and is a subset of Table XII in Appendix B.

2) Tashkeela corpus evaluation: Table XIII in the evaluation appendix details the system results for a random sample of sentences from Tashkeela corpus that are fully diacritized MSA texts taken from various internet sources, such as Al Jazeera and al-kalema.org. Numbers, foreign words, misspellings, partially diacritized text, and colloquial dialects were out of the system scope.

The phonetizer output was compared with the output from a native Arabic speaker trained in reading IPA and MSA, who provided an IPA transcription of the texts as this was not provided in the Tashkeela corpus. Table XIII indicates that the proposed system performed accurately on a large variety of texts.

Differences between the proposed system's output and the expected transcription were due to missing diacritization, the lack of Mad character in a word, and loan words that had a lack of diacritization and sometimes irregular pronunciation. Detailed explanations for these specific differences are shown in Table XIII in Appendix B.

3) Wikipedia sentence evaluation: Table XIV in Appendix B compares the phonetizer output and the IPA transcription for the selected Wikipedia examples [4] [5]. These examples were used because there were no corpora available that contained both orthographic texts with phonetic transcriptions. As can be seen, no deviations were found between the two.

IV. Morphological Transducer (FSAM)

A morphological FST was designed that generates/synthesizes words in the forward direction from morphemes and in the backward direction, decomposes a word into its morphemes. The morphologizer concatenates morphemes that are prefixes, stems, or suffixes and also works on a templatic level when morphemes are interdigitized patterns and roots that make a stem and are meaning-bearing units. Interdigitation refers to the insertion of root components into the corresponding placeholders in the pattern.

In contrast to the existing approaches outlined in Appendix A, the proposed morphologizer is both concatenative and templatic, with the FST used instead of tabulation for the con-

[4] https://en.wikipedia.org/wiki/Varieties_of_Arabic#Examples_of_major_regional_differences
[5] https://en.wikipedia.org/wiki/Arabic_phonology

Table III. Context-Dependent Pronunciation Examples for Various Rule Categories and the Expected (IPA) vs Output

Test phonetic transcription: characters for which the pronunciation is affected by context are colored red.
Φ: zero duration pause, μ: short duration pause, ω: medium duration pause, α: long duration pause
⌣ : continuation, •: not pronounced

Phrase	IPA	Output(s)	Phrase	IPA	Output(s)
Definite Article (ال)			**Alif mad (آ)**		
إِلَى الاِعْتِرَافِ	ʔila⌣lʔiʕtiraːfi	ʔil • a⌣lʔiʕΦtiraːfi	آبَارْ	ʔaːbaːr	ʔaːbaːrΦ
Madd (ٰ)			**Alif (ا)**		
هٰذَا	haːðaː	haːð • aː	وَاقْرَأ	waqraʔ	wa • qΦraʔ μ
Alif maqsurah (ى)			**Harakat (َ ِ ُ)**		
هُدَى القُلُوبْ	huda⌣lquluːb	hud • a⌣lquluːbΦ	حينَ	ħiːna	ħiːna
Hamza and her sisters (ء, أ, إ, ؤ, ئ)			**Waw (و) and Ya (ي)**		
فِـئَـة	fiʔah	fiʔah	مَـوز	mawz	mawz
Ta' Marbutah (ة)			**Tanween (ً ٍ ٌ)**		
الْمَدْرَسَـةْ.	ʔalmadrasah ‖	ʔalmadΦrasahΦ α	أَيٍّ	ʔajːin	ʔajːin

Fig. 2. Arabic word morpheme breakdown. A word is a concatenation of a prefix, stem, and suffix (concatenative). A stem is a meaning-bearing unit that can be further decomposed into its root and pattern (templatic). The root gives the core meaning and the pattern provides the part of speech (POS, category) and other linguistic properties, such as number, tense, and gender. This image uses the Buckwalter transliteration scheme (www.qamus.org/transliteration.htm).

catenative rules. The proposed FST has a single level rather than two levels, into which patterns and roots are input. The morphologizer has a distinct rewrite rules layer to interdigitate the patterns and roots and concatenate the affixes and stems.

Fig. 3 illustrates the proposed FST's inputs and outputs using the word example fasamiEahaA (so he heard her), which was decomposed to the prefix fa (so), the stem samiEa (he heard), and the suffix haA (her), and the stem was further analyzed to the root s m E (to hear) and the pattern faEila (he did). The forward direction FST analysis produced the morphemes and the linguistic features, such as gender, and in the backward direction, generated a word.

The FST works as an acceptor, a synthesizer, and an analyzer and uses the same architecture for both diacritized and undiacritized words. The diacritized version has diacritized morphemes and the undiacritized version has undiacritized morphemes. The morphemes and allowable combinations were derived from multiple linguistic sources (7; 8; 9).

State-of-the-art concatenative morphological formalism comprises three components: lexical automaton, morphotactic rules, and rewrite rules. FSAs are constructed to represent prefixes, stems, and suffixes, which are concatenated with markers based on morphotactic rules that specify valid combinations to separate them into lexical forms, that is, the morphotactic (governing the morphemic combinations, which are meaning-bearing units) and orthographic (spelling) rules are programmed into the FST. The orthographic changes that need to be made to the lexical form to yield the surface form (word) that incorporates contextual mapping are coded using rewrite rules in the FST.

The automaton utilizes morphotactic MSA grammatical rules that govern the allowable affixes and stem concatenations, and the Arabic grammar licit templatic morphological pattern and root combinations, which ensures that there are no invalid words. The proposed architecture incorporates roots and a wide variety of patterns, thereby generating a rich set of valid forms and an average of around 28 analyses per undiacritic word, which compares favorably to the table-based unidirectional universal machines in leading morphologizers that only provide a single analysis and do not have any root-based generation capabilities.

FSAM can be used as a forward direction generator and as a backward direction analyzer for both diacritic and undiacritic words. Therefore, finite-state machines (FSM's) benefits are its unified architecture, its bidirectionality, and its ability to hard wire patterns, which allows for the synthesis, analysis, and diacritization of words without the need for a lookup table.

The generator input is a root that can be either a pattern, an affix, or "print lower-words," and the output is all licit root, pattern, and affix combinations. A word that cannot be decomposed into a pattern and root is a fixed word, such as Washington, which is represented by the root being recognized as a fixed word without affixes and with the pattern being the identity.

The analyzer input is a word and the output is valid alternative morpheme decompositions (prefix, root, and suffix), patterns, parts-of-speech (category), and morphosyntactic features such as number and gender.

FSAM is a composite of three main automata layers, as

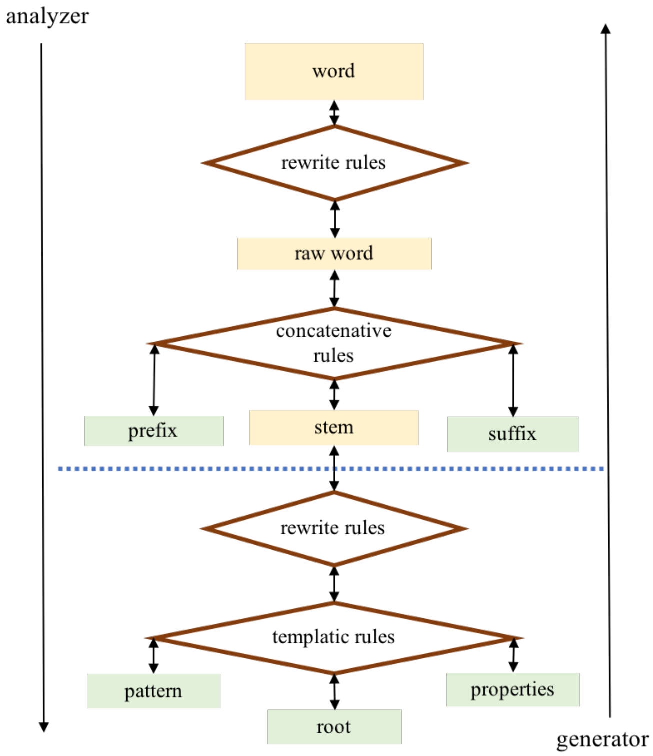

Fig. 3. Architecture for the bidirectional Finite-State Machine based morphological system. The top portion is the rule-based concatenative morphologizer and the bottom portion is the rule-based templatic morphologizer that produces the root, morphological pattern, and properties, such as the category (PoS) and the morphophonemic features. All of these are optional inputs in the generation (synthesis) direction.

shown in Fig. 3: (1) a templatic rule-based automaton that generates the pattern and root combinations into a word; (2) a concatenative rule-based automaton that generates prefix-stem-suffix combinations into a word; and (3) a rewrite rule transducer that applies orthographic and morphophonemic rules to the raw words.

A. Stem Vocabulary Coverage

Stem vocabulary is synthesized in the transducer using a "print lower-words" command, from which a full list of stems is produced, all of which are valid words. The focus is on the stems (base words) rather than the words because of the large vocabulary that arises from additional prefix and suffix combinations. Table VI of Appendix B shows the related statistics.

When the undiacritized stem vocabulary was compared to the undiacritized words in the Tashkeela corpus, an overlap of 88,784 stems was found between the generated FSAM stems and the Tashkeela words, which contrasted favorably (six times more) with the 14,951 stem overlaps between MADAMIRA and the Tashkeela corpus.

B. Expanding Coverage

Based on the compiled morpheme, the morphological automata strictly enforce the allowable prefixes, suffixes, and roots that can combine with a morphological pattern. As morpheme combinations occur, they need to be added to the system sets. To allow for this expansion, a morphological automata version is constructed in which the restrictions on the roots, prefixes, and suffixes that combine with a pattern are removed but the precisely known hardwired patterns are retained.

An example of a pattern is فَعَلَ ('has done'), which could have the restricted sets ء ك ل, د ر س, ف ه م as roots وَ, فَ as prefixes, and هَا as suffixes. Therefore, if the word كَتَبَ is input into the proposed system, which does not have the root ك ت ب in the sets related to the فَعَلَ pattern, it can be analyzed using the open system and then added to the closed system, which only allows valid words to be analyzed, to improve the coverage.

If a trilateral pattern is allowed to correspond to any three-letter root, there is an unrestricted subsystem that allows valid words to be analyzed and the list of roots in the restricted system to be expanded. However, as this subsystem also admits many invalid words, it can only be used by a language specialist to expand the morpheme list.

C. Evaluation

Different data sets and sources were employed to evaluate the various system parts. As detailed in Appendix B, to ensure there were no invalid or dialectal words that ignored the OOV and punctuation, a gold standard treebank was developed from the intersection of the PADT UD treebank[6] and the Tashkeela[7] corpus to test the morphologizer generation and analysis (synthesis) tasks for both the undiacritic and diacritic words.

The FSAM and FSAD results were compared to the leading Arabic morphologizer, MADAMIRA, which is a concatenative morphological analyzer that uses a Penn Arabic treebank as part of its training set and overlaps with the UD PADT. MADAMIRA(10) is a combination of the MADA (Morphological Analysis and Disambiguation of Arabic), which was built based on the SAMA (Standard Arabic Morphological Analyzer) and AMIRA (a morphological system for colloquial Egyptian Arabic). Different from MADAMIRA, FSAM and FSAD's rule-based system conducts an MSA templatic morphological analysis that yields a root and pattern, generation, and diacritization.

1) Synthesizer evaluation: FSAM synthesizes words in two ways: 1) it inputs the prefix-root-suffix to the system and outputs all words resulting from the many pattern and root combinations; and, 2) it issues a "print lower-words" command to the transducer to synthesize all stems that are valid pattern and root combinations or all words that are valid pattern, root, prefix, and suffix combinations. FSAM synthesizes the word vocabulary corresponding to the gold standard by inputting the root, prefix, and suffix combinations, which are decompositions of the gold standard words in the treebank. Consequently, the vocabulary is larger than the gold standard because of the additional patterns applicable to the prefix-root-suffix combinations. Table IX in the appendix illustrates the tremendous effect that the patterns have.

To evaluate the root generation ability, the root provided by the gold standard and the prefix and suffix provided by the gold standard word segmentation were used to generate possible

[6]https://github.com/UniversalDependencies/UD_Arabic-PADT
[7]https://sourceforge.net/projects/tashkeela/

TABLE IV. FSAM-GENERATED WORDS FROM THE GOLD STANDARD ROOTS. THE 'GENERATED' COLUMN SHOWS THE PERCENTAGE OF ROOTS THE MODEL GENERATED FROM THE WORDS; FOR EXAMPLE, ROOT ك IS NOT CONSIDERED A ROOT IN ARABIC, AND THEREFORE, NO WORDS WERE YIELDED. THE 'CORRECT' COLUMN IS THE PERCENTAGE OF FSAM-GENERATED WORDS THAT MATCHED THE GOLD STANDARD.

Generated/Synthesized Words from Roots

UNDIAC	generated	correct
verb	94.96	100
tool word	90.71	100
noun	91.71	100
proper name	91.28	100
noun+verb	92.48	100
all	91.89	100

words from the prefix, root, pattern, and suffix combinations. Table IV shows that 100% accuracy and 92% coverage were achieved when generating words from the root and its prefix and suffix.

2) Analyzer evaluation using the gold standard: The analyzer input is a word and the FSAM output is the root, pattern, category, or other linguistic information, such as number, gender, case, definiteness, and aspect. As the MADAMIRA output does not include the root or pattern, MADAMIRA was run in analysis-only mode.

A full analyzer evaluation should only be conducted against a gold standard reference. The Tashkeela corpus, however, is only a collection of morphologically valid Arabic words, whereas the gold standard treebank has root, category, and other linguistic information. For the undiacritized evaluation, all treebank words were input into the analyzer and matched against the analysis. The gold standard no OOV treebank was then used to evaluate the systems. Both systems had around 99% accuracy when computing gender, definiteness, person, case, aspect, and voice; however, the FSAM performed well for mood (99.7% vs 93%) and number (97.8% vs 90.5%) and was able to determine the root correctly about 92% of the time. Appendix B provides more details on the FSAM evaluation.

The advantage of the proposed system is that it can extract the word roots and patterns, that is, it can provide a shallow analysis of a word based on the pattern without needing to refer to a table of stems and their properties. Both systems' properties could produce the category, case, gender, mood, definiteness, number, person, voice, and aspect.

3) Analyzer coverage evaluation: The model coverage was evaluated by computing the percentage of analyzed words using a large corpus (Tashkeela). The FSAM analyzed 81.83% of the undiacritized words in the Tashkeela corpus and analyzed 82.24% of the undiacritized words in MADAMIRA (in analyses-only mode and no backoff). The backoff mode in MADAMIRA was not used because it admits invalid words.

The reduced coverage was largely because of the invalid words in the corpus. Invalid words are words that are misspelled, not words in the Arabic language, or a concatenation of words. Examples of words that could not be analyzed by both systems and were deemed invalid were شرنبلالي, السنبوسك , فوشيكوس , and words that were not separated by whitespace and were considered to be one word (the dash (-) indicates where the words should be separated), such as : عباس-الفواحش, بالليل-والإباحة, المصلين-والوجه, العدو-عليكم, الشعثاء-في, السدي-وخرج, الأواه-الذي, المسلمين-حال, وغيرها-وإني, مالك-والشافعي, المساجد-إلا, فأخبرني-محمد, يعني-البينات, عصير-والوجه, قوله-وهذا, سفيان-أن, أول-احتباسها

V. DIACRITIC TRANSDUCER (FSAD)

As illustrated in Fig. 5, the system's diacritizer was developed using diacritized fixed words, the prefix and suffix listing for the simple diacritizer, and the diacritized MSA patterns for the pattern-based diacritizer. The simple diacritizer was used for the fixed words and affixes because they did not follow any pattern.

The diacritizer was designed in the forward direction, in which diacritics were inserted. The FST for the fixed words and affixes is a table that maps between the diacritized and undiacritized versions. The model used for the pattern-based words was an insertion FST that inserted diacritics into an undiacritic pattern to create the diacritic counter parts; for example, [8] $\Omega\Gamma\Lambda \rightleftharpoons \Omega a\Gamma a\Lambda a, \Omega a\Gamma{\sim}a\Lambda a, \Omega a\Gamma i\Lambda a, \Omega u\Gamma i\Lambda a$, where Ω, Γ, and Λ were placeholders for the root.

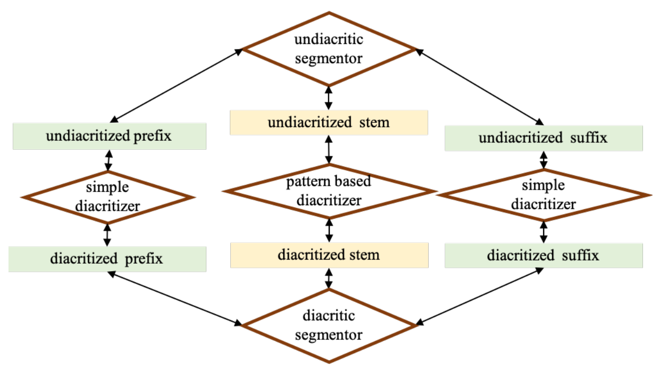

Fig. 4. Architecture for the finite-state machine-based diacritizer. The downward (forward) direction outputs diacritized words from an input undiacritized word. The segmenter decomposes the word into its prefix-stem-suffix. The pattern-based diacritizer inserts the diacritics into an undiacritized pattern to produce corresponding diacritized patterns; for example, فعل ⇌ فَعَلَ, فَعِلَ, فِعِلَ. To diacritize the stem using the pattern diacritizer, the stem is matched with the corresponding undiacritized pattern to produce the diacritic stem. The simple diacritizer inserts the diacritics directly into the undiacritized affix.

The segmenter decomposes a word into its prefix-stem-suffix components for which the stem could be a pattern-based word or a fixed word. After the word components are diacritized, they are then concatenated to form the diacritic word. The system diacritizer is illustrated in Fig. 4. A sample input and output(s) to this system is ودرسها ⇌ وَدَرَسَهَا, وَدَرَّسَهَا.

The diacritizer was evaluated by selecting all undiacritized words in the gold standard treebank, passing them into the diacritizer, and checking the output against the diacritized word contained in Vform. The diacritizer output was evaluated according to standard Arabic spelling rules. Note that the gold

[8]Please note we are using Buckwalter transliteration when not using Arabic script: http://www.qamus.org/transliteration.htm

standard meets these standards with some exceptions that are only apparent upon visual inspection.

Table V. Diacritization Accuracy for the Treebank. The Pattern-based Model had Significantly Higher Accuracy

Word	**FSAD**	**MADAMIRA**
verb	85.91	80.99
proper name	82.46	50.78
noun	83.67	53.49
noun+verb	84.01	58.76
toolword	83.34	53.43
all	**83.65**	58.59

The evaluation in Table V indicates that the FSAD performed better than the MADAMIRA for the full diacritization (84% vs 59%) because the FSAD does not learn the diacritization from the corpus but deduces it based on the patterns that exist in the Arabic language, whereas MADAMIRA trains its model on corpora and, therefore, has a more partial diacritization.

Because the gold standard has spelling inconsistencies between the diacritized and undiacritized words, the performance was reduced, as shown in Table V. The following examples had the following (inconsistencies), which could have had a significant effect on the evaluation.

- Using ى instead of ي (e.g., فِي ⇌ فى , مَدَنِيٌّ ⇌ مدنى, حَوَالَي ⇌ حوالى, أُلغِيَ ⇌ ألغى)
- Using ا instead of آ (e.g., آبَ ⇌ اب, اَلآسِتَانَةِ ⇌ الاستانة, آخَرُ ⇌ اخر)
- Using ا instead of إ (e.g., إِلَى ⇌ الى, إِطلَاقِ ⇌ اطلاق, إِشَارَةٍ ⇌ اشارة)

VI. Conclusion

This study designed and constructed a bidirectional integrated phonetizer, morphologizer, and diacritizer system (FSPMD), the coverage of which could be increased by adding foreign words and special morpheme roots with the associated rules in the appropriate order. The FSPMD structure could be mimicked to build morpho-phonological systems for rule-based languages, such as Hebrew and Aramaic. The system could also be used in many language technologies, such as speech recognition, information retrieval, and spelling and grammar checkers, without the need to incorporate large tabulations that increase system complexity, out-of-vocabulary words, and perplexity. The system could also be used to construct a semantic analyzer and word translator and as part of a suprasegmental phonologizer that applies syllables, stress, and intonation rules, which would make it useful for text-to-speech technologies. On the text side, the syntactic parser has greater scope than most FSMs, which means it can deal with long-distance rules beyond formal languages, such as context-sensitive grammar or tree adjoining grammar.

References

[1] D. Jurafsky and J. H. Martin, Eds., *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2020.

[2] F. Seifart, ``Orthography development,'' *Essentials of language documentation*, pp. 275--299, 2006.

[3] R. Hetzron, Ed., *The Semitic Languages*. Routledge, 1997.

[4] M. Hulden, ``Foma: a finite-state compiler and library,'' in *Proceedings of the Demonstrations Session at EACL 2009*, 2009, pp. 29--32.

[5] N. Halabi, ``Modern standard arabic phonetics for speech synthesis,'' Ph.D. dissertation, University of Southampton, 2016.

[6] T. Zerrouki and A. Balla, ``Tashkeela: Novel corpus of arabic vocalized texts, data for auto-diacritization systems,'' *Data in Brief*, vol. 11, pp. 147 -- 151, 2017.

[7] A. Dahdah and G. M. Abdulmassih, *A dictionary of Arabic grammar in charts and tables*. Librairie du Liban, 1981.

[8] M. b. A. B. Al-Razi, ``Mukhtar al-sihah,'' *Beirut: Dar al-Namudzajiyah*, 1999.

[9] A. El-Dahdah, E. Matar, and G. M. Abdul-Massih, ``majam qawaaid al-arabiiat al-aalamiiat (a dictionary of universal arabic grammar)/معجم قواعد العربية العالمية,'' 1990.

[10] A. Pasha, M. Al-Badrashiny, M. T. Diab, A. El 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 *Lrec*, vol. 14, no. 2014, 2014, pp. 1094--1101.

[11] 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*, vol. 1. Citeseer, 2001, pp. 1--8.

[12] N. Y. Habash and O. C. Rambow, ``Magead: A morphological analyzer and generator for the arabic dialects,'' 2006.

[13] K. Darwish, M. Diab, and N. Habash, ``Proceedings of the acl workshop on computational approaches to semitic languages,'' in *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, 2005.

[14] 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 the 9th International Workshop on Finite State Methods and Natural Language Processing*, 2011, pp. 125--133.

[15] K. Darwish, ``Building a shallow arabic morphological analyser in one day,'' in *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*, 2002.

[16] T. Buckwalter, ``Buckwalter arabic morphological analyzer version 1.0,'' *Linguistic Data Consortium, University of Pennsylvania*, 2002.

[17] D. Graff, M. Maamouri, B. Bouziri, S. Krouna, S. Kulick, and T. Buckwalter, ``Standard arabic morphological analyzer (sama),'' *Linguistic Data Consortium LDC2009E73*, 2010.

[18] O. Smrz, ``Elixirfm--implementation of functional arabic morphology,'' in *Proceedings of the 2007 workshop on computational approaches to Semitic languages: common issues and resources*, 2007, pp. 1--8.

[19] R. Roth, O. Rambow, N. Y. Habash, M. Diab, and C. Rudin, ``Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking,'' 2008.

[20] E. L. Antworth, ``Pc-kimmo: a two-level processor for morphological analysis,'' *Summer Institute of Linguistics*,

1990.
[21] L. Karttunen, *Finite-state lexicon compiler*. Xerox Corporation, Palo Alto Research Center, 1993.
[22] G. Kiraz, ``Multi-tape two-level morphology: a case study in semitic non-linear morphology,'' *arXiv preprint cmp-lg/9407023*, 1994.
[23] M. Mohri, F. Pereira, and M. Riley, ``Weighted finite-state transducers in speech recognition,'' *Computer Speech Language*, vol. 16, no. 1, pp. 69--88, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230801901846
[24] M. T. Diab, ``Second generation amira tools for arabic processing : Fast and robust tokenization , pos tagging , and base phrase chunking,'' 2009.
[25] A. A. Al-Nassir, ``Sibawayh the phonologist: A critical study of the phonetic and phonological theory of sibawayh as presented in his treatise? al kitab?'' Ph.D. dissertation, University of York, 1985.
[26] Y. A. El-Imam, ``Phonetization of arabic: rules and algorithms,'' *Computer Speech & Language*, vol. 18, no. 4, pp. 339--373, 2004.
[27] F. Biadsy, N. Habash, and J. Hirschberg, ``Improving the arabic pronunciation dictionary for phone and word recognition with linguistically-based pronunciation rules,'' in *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, 2009, pp. 397--405.
[28] A. Ramsay, I. Alsharhan, and H. Ahmed, ``Generation of a phonetic transcription for modern standard arabic: A knowledge-based model,'' *Computer Speech & Language*, vol. 28, no. 4, pp. 959--978, 2014.
[29] M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki, ``The penn arabic treebank: Building a large-scale annotated arabic corpus,'' in *NEMLAR conference on Arabic language resources and tools*, vol. 27. Cairo, 2004, pp. 466--467.
[30] J. Åkesson, ``Arabic morphology and phonology: Based on the marāḥ al-arwāḥ by aḥmad b.ʿaī b. mas ʿūd,'' in *Arabic Morphology and Phonology*. Brill, 2017.
[31] A. A. S. Farghaly, ``Arabic computational linguistics,'' *(No Title)*, 2010.
[32] M. Sipser, *Introduction to the Theory of Computation*. Cengage Learning, 2012.
[33] R. M. Kaplan and M. Kay, ``Regular models of phonological rule systems,'' *Computational linguistics*, vol. 20, no. 3, pp. 331--378, 1994.
[34] K. Koskenniemi, ``Two-level morphology,'' Ph.D. dissertation, Ph. D. thesis, University of Helsinki, 1983.
[35] L. Karttunen *et al.*, ``Kimmo: a general morphological processor,'' in *Texas Linguistic Forum*, vol. 22. Texas, USA, 1983, pp. 163--186.
[36] J. Bear, ``A morphological recognizer with syntactic and phonological rules,'' in *COLING*, vol. 86, no. 10.3115, 1986, pp. 991 365--991 445.
[37] ------, ``Morphology with two-level rules and negative rule features,'' in *Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics*, 1988.
[38] A. W. Black, G. Ritchie, S. Pulman, and G. Russell, ``Formalisms for morphographemic description,'' in *Third Conference of the European Chapter of the Association for Computational Linguistics*, 1987.
[39] K. Beesley, T. Buckwalter, and S. Newton, ``Two-level finite-state analysis of arabic morphology,'' in *Proceedings of the Seminar on Bilingual Computing in Arabic and English*, 1989, pp. 6--7.
[40] H. Trost, ``The application of two-level morphology to non-concatenative german morphology,'' 1990.
[41] G. D. Ritchie, *Computational morphology: practical mechanisms for the English lexicon*. MIT press, 1992.
[42] E. L. Antworth, ``Morphological parsing with a unification-based word grammar,'' in *Proceedings of the North Texas Natural Language Processing Workshop*. Citeseer, 1994, pp. 24--32.
[43] H. Ruessink, *Two-level formalisms*. Katholieke Universiteit, 1989.
[44] D. Carter, ``Rapid development of morphological descriptions for full language processing systems,'' *arXiv preprint cmp-lg/9502006*, 1995.
[45] E. Grimley-Evans, G. A. Kiraz, and S. G. Pulman, ``Compiling a partition-based two-level formalism,'' *arXiv preprint cmp-lg/9605001*, 1996.
[46] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.
[47] C. D. Johnson, *Formal aspects of phonological description*. Walter de Gruyter GmbH & Co KG, 2019, vol. 3.
[48] K. R. Beesley and L. Karttunen, ``Finite-state morphology: Xerox tools and techniques,'' *CSLI, Stanford*, pp. 359--375, 2003.
[49] S. Bird and T. M. Ellison, ``One-level phonology: Autosegmental representations and rules as finite automata,'' *Computational Linguistics*, vol. 20, no. 1, pp. 55--90, 1994.
[50] K. R. Beesley and L. Karttunen, ``Finite-state non-concatenative morphotactics,'' *arXiv preprint cs/0006044*, 2000.
[51] J. J. McCarthy, ``A prosodic theory of nonconcatenative morphology,'' *Linguistic inquiry*, vol. 12, no. 3, pp. 373--418, 1981.
[52] M. Kay, ``Nonconcatenative finite-state morphology,'' in *Third Conference of the European Chapter of the Association for Computational Linguistics*, 1987.
[53] J. A. Goldsmith, *Autosegmental and metrical phonology*. Basil Blackwell Cambridge, 1990, vol. 1.
[54] J. McCarthy and A. Prince, ``Prosodic morphology and templatic morphology,'' in *Perspectives on Arabic linguistics II: papers from the second annual symposium on Arabic linguistics*. John Benjamins Pub. Co. Amsterdam, 1990, pp. 1--54.
[55] J. J. McCarthy and A. Prince, ``Generalized alignment,'' in *Yearbook of morphology 1993*. Springer, 1993, pp. 79--153.
[56] L. Kataja and K. Koskenniemi, ``Finite-state description of semitic morphology: A case study of ancient accadian,'' in *Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics*, 1988.
[57] K. Beesley, ``Finite-state description of arabic morphology,'' in *Proceedings of the Second Cambridge Conference: Bilingual Computing in Arabic and English*, 1990, pp. 5--7.
[58] K. R. Beesley, ``Computer analysis of arabic morphology: A two-level approach with detours,'' in *Perspectives on Arabic Linguistics III: Papers from the Third Annual*

Symposium on Arabic Linguistics. John Benjamin's Publishing Company Amsterdam, 1991, pp. 155--172.

[59] ------, ``Arabic finite-state morphological analysis and generation," in *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.

[60] R. W. Sproat, *Morphology and computation*. MIT press, 1992.

[61] S. G. Pulman and M. R. Hepple, ``A feature-based formalism for two-level phonology: a description and implementation," *Computer Speech & Language*, vol. 7, no. 4, pp. 333--358, 1993.

[62] A. Narayanan and L. Hashem, ``On abstract finite-state morphology," in *Conference of the European Chapter of the Association for Computational Linguistics*, 1993.

[63] K. R. Beesley, ``Arabic morphology using only finite-state operations," in *SEMITIC@COLING*, 1998.

[64] G. A. Kiraz, ``Multitiered nonlinear morphology using multitape finite automata: a case study on syriac and arabic," *Computational Linguistics*, vol. 26, pp. 77--105, 2000.

[65] A. Kornai, ``Formal phonology," 2018.

[66] B. Wiebe, ``Modelling autosegmental phonology with multi-tape finite state transducers," 1992.

Appendix A
Related Work

A. Arabic Morphologizers

The most significant morphological analyzers are those that utilize finite-state transducer formalism, such as Xerox(11), the morphological analyzer and generator for the Arabic dialects (MAGEAD)(12; 13) , and the Arabic Computer Lexicon (AraComLex)(14), and those that utilize a tabular approach, such as Darweesh(15), the Buckwalter Morphological Analyzer for Arabic (BAMA)(16), the Standard Arabic Morphological Analyzer (SAMA)(17), ElixirFM(18), a high-level implementation of functional Arabic morphology, Morphological Analysis and Disambiguation of Arabic (MADA)(19), and MADAMIRA(10).

The Xerox Arabic morphologizers, which are bidirectional morphologizers that take diacritized or undiacritized words as the input and compute the prefix, pattern, root, and suffix, were based on a finite-state transducer (FST) and utilize grammar rules rather than listing the stems. For example, Beesley's(11) Xerox finite-state morphological analyzer, which was built on finite-state transducer advancements to handle morphology and uses Xerox finite-state language modeling tools, is rule-based and has a large coverage. Because the Xerox finite-state morphological analyzer adopts a root-and-pattern approach, it can generate all possible morphological features for each word; 4,930 roots and 400 patterns that generate 90,000 stems; can also reconstruct the vowel marks, and provides an English glossary for each word. The Xerox finite-state morphological analyzer was based on the ALPNET developed earlier by Beesley and Buckwalter, which was founded on PC-KIMMO. This tool was constructed as two-level morphology by Antworth and Karttunen(20; 21).

The MAGEAD morphologizer system, which extended Kiraz's(22) work using AT&T's finite-state machine toolkit(23), decomposes an isolated diacritized word into prefix, pattern, root, and suffix and generates words from input morphemes, that is, it is a bidirectional morphologizer. MAGEAD also provides linguistic features, such as word class, in a hierarchical form; however, it is currently restricted to verbs. MAGEAD, which is based on a multitape finite-state transducer similar to the Xerox-based work of Beesley and Kiraz, has morphophonemic and orthographic rewrite rules that extend Kiraz's analysis by introducing a fifth tier: Tier 1: pattern and affixational morphemes; Tier 2: root; Tier 3: vocalism; Tier 4: phonological representation; and Tier 5: orthographic representation. In the generation direction, tiers 1 through 3 are always the input tiers, Tier 4 is first, an output tier and then, a subsequent input tier, and Tier 5 is always an output tier.

The AraComLex is an open-source data-driven Arabic morphologizer that utilizes a bidirectional FST and uses the lemma as its base form. As a lemma is a marked form of a word without affixes and is not inflected, it has shorter lexicons than stem-based methods and can benefit from generalized rules rather than listings. For Arabic, this is typically the perfective, 3rd person singular verbs or the singular indefinite form for nouns and adjectives. Other inflected forms are derived from the lemma using alteration rules, which is different from the root-based Xerox morphologizer and the stem-based BAMA/SAMA morphologizers.

Darwish's(15) tabular method is a morphological analyzer that uses automatically-derived rules and statistics from the "Build-Model" module in the morphologizer. This module takes a list of word-root pairs as the input, which allows it to extract a list of prefixes, suffixes, and stem templates. The probability of each item's occurrence in these lists is then used to generate the statistical rules. The Darwish Morphologizer has been found to have an 84% success rate(15). Its "Detect-Root" module extracts all possible roots for an input word by generating a prefix, suffix, and stem, removing the prefix and suffix from the stem, and matching the stem against the templates, with the resultant template (along with the stem) being used to determine the root.

The BAMA was first developed by Buckwalter and has since had three versions; BAMA[9] versions 1.0 and 2.0(16); and SAMA [10] 3.1.; all of which are available as source codes from LDC. The input and output are in transliterated Roman letters and the program is written in Perl.

SAMA's input is isolated words, that is, sentence context is not considered in the disambiguation. The input word may be either diacritized on undiacritized, with the output being all possible prefix, stem, and suffix combinations, that is, it is a stemmer rather than a deep morphologizer. As SAMA is non-bidirectional, words may not be generated from the prefix, stem, and suffix inputs. In addition to stemming, the BAMA/SAMA also provides a part of speech tag. Rather than incorporating grammatical rules, BAMA/SAMA uses manually entered lexicons and morphotactic rules as its tables, which makes it difficult to generalize and requires significant manual effort to scale. In addition to the tables that specify the allowable prefix, suffix, and stem set combinations, the lexicons also include prefix, suffix, and stem sets. BAMA 1.0 has 299 prefixes,

[9] https://catalog.ldc.upenn.edu/LDC2002L49, https://catalog.ldc.upenn.edu/LDC2004L02

[10] https://catalog.ldc.upenn.edu/LDC2010L01

618 suffixes, and 82158 stems, 1648 prefix-stem combinations, 1285 stem-suffix combinations, and 598 prefix-suffix combinations. SAMA 3.1 has 1328 prefixes, 945 suffixes, 79318 stems, 40654 stem categories, 2497 prefix-stem combinations, 1632 stem-suffix combinations, and 1180 prefix-suffix combinations. A simple Perl program uses these to segment a word into all possible prefix-stem-suffix set combinations. Although the BAMA/SAMA uses reasonably sized tables, it is quite efficient and compact.

ElixirFM uses SAMA resources and Haskell's functional morphology library to incorporate the interface between morphology and syntax and determine the morphophonemic patterns to identify the roots and templates for the SAMA lexical items. MADA (Morphological Analysis and Disambiguation for Arabic) uses Support Vector Machines to compute nineteen features; five for spelling variations and fourteen for morphological features, such as number, gender, case, and mood. MADAMIRA is a concatenative morphological analyzer that uses the Penn Arabic treebank as part of its training set. MADAMIRA (10) is a combination of the MADA, which was based on SAMA, and AMIRA (24), a morphological system for colloquial Egyptian Arabic).

B. Arabic Phonetizers

Classical Arabic literature provides a rich set of pronunciation rules for classical Arabic(25). Recent publications provide pronunciation rules for modern Arabic that were extracted from traditional sources(26; 27). However, some rules that should be included have been excluded, others that should have been excluded because they relate to spelling have been incorporated, and the effect of text markers has been ignored, all of which have significant consequences on phonetization, which means that the pronunciation produced using these rules can have many errors.

In recent publications, gemination has been handled by doubling a singleton consonant or mapping into its singleton version, which is phonetically inaccurate as demonstrated by geminated plosives that have a single voice onset time and release. Similarly, a long vowel is dealt with by doubling its short version, whereas the spectral characteristics of a long vowel are noticeably different from its short vowel counterpart**(author?)** (5). Also, the rules related to Wasl characters (ب ف ك و) are ignored, even though they frequently occur. Four additional forms for (ء) (أ إ ؤ ى ئ), (آ), end of sentence vowels, mapping (ا) at the beginning of a sentence into a glottal stop, and the pronunciation of (ة) as /h/ at the end of the sentence are also not considered.(26) ignored short vowels at the end of a sentence by removing them and (آ), and(27) did not incorporate situations that require the mapping of (اً).

(28) used a rule-based two-level finite-state automata to develop an orthographic to allophonic mapping, with the first level being grapheme to grapheme changes such as deletion and duplication, and the second level being grapheme to phoneme changes. The allophonic changes are then applied to the phoneme level to produce an allophonic transcription and evaluate the system output on the diacritized words from the Penn Arabic treebank(29). While some of the problems in previous publications were resolved, duplication rather than gemination is employed and Sokoon is excluded from the rules when it uses syllable structure to determine the pronunciation of waw, ya', and Alif Wasl. In addition, ta' marbutah is deleted rather than pronouncing it as /h/ at the end of an utterance.

Appendix B
FSPMD (System) Evaluation

A. Reference Corpus for Evaluation

Tashkeela(6), PADT_UD treebank, and MADAMIRA(10) were used to evaluate and compare the performance of the developed morphology generator, analyzer, and diacritizer. Tashkeela, Wikipedia , and modern standard Arabic orthography to phoneme transcriptions as well as specific examples that highlighted edge cases were used to fully test the orthography to phoneme transcription system.

As FSAM performs generation, analysis, and diacritization tasks that cover both undiacritic and diacritic words, a corpus of diacritized Arabic was needed to evaluate the proposed system. Tashkeela is one of the few available corpora that satisfied our requirements as it is a collection of diacritized passages in Classical and modern standard Arabic. Further, as our system is a deep morphologizer that works at pattern and root levels, the PADT_UD treebank, which was built on the Prague Arabic Treebank (Hajic et al. 2004), was the only resource available for a granular generation and analysis evaluation because alternatives such as the Penn Arabic treebank (Maamouri et al. 2004) lack root information. The PADT_UD is the Universal Dependencies Prague Arabic Treebank of modern standard and colloquial Arabic that contains undiacritized words, with the analysis consisting of the root, the Vform (the diacritized word), gender, number, case, definite, voice, and others. The FSAM was compared with the MADAMIRA (in analysis-only mode), which is a concatenative morphologizer (a morphologizer that gives the features of the words such as number, gender, person, etc. but does not give the composition of the word in terms of its pattern and root) rather than a templatic morphologizer, which partially makes up for the absence of patterns and roots by utilizing the SAMA stem categories to provide some granular analysis.

1) Fully diacritized text and corpus and treebank vocabulary: The diacritized texts in the Tashkeela corpus were utilized to manually test the ability of our orthography to phonemic systems and to test the correctness of our model, IPA was used to transcribe the MSA fully diacritized sentences from Wikipedia.

The undiacritized and diacritized word vocabulary was computed in Tashkeela and PADT_UD. Table VI conveys the word statistics after the punctuation was removed.

2) Gold sandard : A gold standard was generated from the PADT_UD treebank as a reference for the evaluation of the analysis and generation capabilities. A gold standard must be free from punctuation, abbreviations (e.g., كم "km"), foreign words (e.g., واشنطن "Washington"), affixes (e.g., ال), and single-character graphemes (ت); which are not considered words in the Arabic language.

To eliminate words that were colloquial rather than modern standard Arabic, PADT_UD was intersected with Tashkeela, followed by the serial removal of affixes, single-character graphemes (letters), foreign words, and abbreviations. Table VI also details the statistics for the intersection between

Table VI. LEFT: Diacritized vocabulary and the resulting undiacritized words in each resource ignoring punctuation. The PADT_UD diacritized words are those listed as Vform (vocalized form) in its analysis of undiacritized words. RIGHT: Gold Standard Treebank is the intersection of Tashkeela and the PADT_UD followed by the removal of isolated affixes, letters, foreign words, abbreviations, and entries with no analysis (oov)

References Vocabulary					
	undiacritized	diacritized		undiacritized	diacritized
Tashkeela	481,611	982,922	Intersection	16,760	27,097
PADT UD	23,175	33,597	Gold Standard	16,469	26,772
			Gold Standard - no OOV	15,035	24,080

PADT_UD, Tashkeela, and the gold standard, which is the intersection that excludes affixes, foreign words (determined by Foreign = Yes in the PADT_UD analyses), and abbreviations (determined by Abbr = Yes in the PADT_UD analyses).

3) Category correspondence: There is a mismatch in groupings and terminologies between our system, PADT_UD, and MADAMIRA. As the proposed system is based on Arabic language constructs, it uses intrinsic categories; verb, noun, tool word, and proper name. The verbs and nouns are further classified as regular and irregular. In contrast, PADT_UD labels words according to the standard part-of-speech classification scheme in English, and MADAMIRA labels words according to stem classes in the underlying SAMA corpus.

Table VII. PADT_UD label correspondence to the noun, verb, tool word, and proper name categories, and the diacritized (diac) and undiacritized (undiac) statistics for each label

LABEL	CATEGORY	DIAC	UNDIAC
NOUN	noun proper name tool word	14,405	8,424
X	proper name tool word noun	2,693	2,679
VERB	verb	4,603	3,551
PART	tool word	19	21
CCONJ	tool word proper name	83	49
AUX		99	90
PRON	tool word	12	34
ADV	tool word	22	25
DET	tool word	34	37
PROPN	proper name	29	28
ADJ	noun proper name	5199	3587
INTJ	noun tool word	3	3
ADP	tool word proper name noun	94	105

A label can map onto more than one category. For instance, a noun in PADT_UD may be a noun, proper name, or tool word as it contains words such as ميراث (inheritance) "noun," دولار (dollar) "proper name," and كل (all) "tool word." Therefore, a word that is analyzed as a noun in PADT_UD and analyzed as a tool word in the proposed model is marked as a tool word and a match occurs.

Table VIII. MADAMIRA label correspondence to the noun, verb, tool word, and proper name categories

LABEL	CATEGORY	LABEL	CATEGORY	LABEL	CATEGORY
abbrev	proper name	noun quant	noun	part interrog	tool word
noun prop	proper name	noun num	noun	part neg	tool word
verb	verb	noun	noun	part restrict	tool word
verb pseudo	verb	adj	noun	part verb	tool word
adv	noun	adj comp	noun	part voc	tool word
adv interrog	noun	adj num	noun	prep	tool word
adv rel	noun	part	tool word	pron	tool word
conj	noun	part det	tool word	pron dem	tool word
conj sub	noun	part focus	tool word	pron interrog	tool word
interj	noun	part fut	tool word	pron rel	tool word

Table VII details the correspondence between the PADT_UD labels and the categories in the proposed system. Table VIII details the MADAMIRA label correspondence to the various categories: noun, verb, tool word, and proper name.

B. FSAM analysis evaluation

Table XI compares the MADAMIRA's and FSAM's verb and noun analyses. Because of the overlap between Penn Arabic treebank, which is used as the MADAMIRA training corpus, and UD_PADT, the basis of our gold standard, MADAMIRA analyzed around 100% of the gold standard verbs and nouns, whereas FSAM analyzed around 84% of verbs and nouns.

MADAMIRA categorized the word correctly 100% of the time and FSAM categorized it correctly 97% of the time. Both systems had similar performances at around 99% accuracy when computing gender (99.5% vs 99.4%), definitiveness (99.3% vs 98.0%), person (98.2% vs 99.9%), case (99.4% vs 99.8%), aspect (99% vs 99.9%), and voice (99.8% vs 97.9%). FSAM performed better for mood (99.7% vs 93%) and number (97.8% vs 90.5%), and found the root with approximately 92% correctness.

C. FSAP (Phonetic Transducer) Evaluation

The full range of examples to test FSAP are provided in Tables XII, XIII and XIV. In addition to the IPA non-phonemes of continuation (‿), medium duration pause (|), and long duration pause (‖), we used zero duration pause, short duration pause, and not pronounced to more comprehensively reflect the morpho-phonetic relationships. These are the only expected differences between the Expected IPA and Output.

Table XII tests FSAP in all context dependent pronunciation environments, and as can be observed from the table, the system performs with 100% accuracy in those examples. Table XIII uses diacritized text from the Tashkeela corpus(6) to evaluate the system on diverse examples. Both tables XII and XIII don't have the expected IPA transcription as part of the corpus, so we used a language expert to transcribe the sentences into IPA to get the expected output.

Table IX. FSAM-generated stem vocabulary for each sub-category and category (Top), and MADAMIRA tabulated stem vocabulary (Bottom). *UNK means that the reference has no stem categorization. No counterpart in MADAMIRA unless their reference - SAMA (has a listing of stems) - is directly utilized. Noun, verb, tool word, and proper name categories are based on the label correspondence in the MADAMIRA reference table shown in Table VIII. Note that a stem has multiple labels in the reference

Stem Vocabulary					
FSAM	undiacritized	diacritized	MADAMIRA	undiacritized	diacritized
regular verb	579,522	1,882,047	verb	4,269	4,843
irregular verb	98,668	282,611	noun	11,950	12,763
regular noun	716,177	2,192,815	toolword	32	37
irregular noun	157,322	405,834	proper name	544	556
toolword	238	261	UNK*	8,849	11,897
proper name	7,681	8,352			
TOTAL	1,196,895	4,018,302	TOTAL	24,055	29,685

Table X. Overlap count between the synthesized undiacritized stems and the gold standard stems. The intersection is between the gold standard and synthesized stems. Missing is the set gold standard stems, that is, the synthesized stems. As there is no reference to MADAMIRA for the generation of stems from roots, the overlap of stems was checked from the underlying listing for the gold standard stems (8,536 stems)

Generated Stem Overlap with the Gold Standard Stems		
UNDIAC	FSAM	MADAMIRA
Intersection	6,622	5,146
Missing	1,914	3,390
Total	8,536	8,536

Table XI. Analysis accuracy for the undiacritized words for the gold standard treebank. On the left is FSAM (F) and on the right is MADAMIRA (M). To produce roots, the model outperformed in mood, number, and voice properties. MADAMIRA had almost full coverage of the gold reference because of the overlap between the training data and the reference

Analysis Performance FSAM (F) vs MADAMIRA (M)						
UNDIAC	verb		noun		noun+verb	
	F	M	F	M	F	M
analyzed	94.9	99.9	83.4	99.8	83.8	**99.8**
category	99.0	99.9	96.8	100	97.0	100
root	94.0	NA	91.8	NA	**92.3**	NA
case	-	-	99.4	99.8	99.4	99.8
gender	99.4	100	99.6	98.9	99.5	99.4
mood	99.7	92.2	-	-	**99.7**	92.2
definite	-	-	99.3	98.0	99.3	98.0
number	99.3	100	97.4	88.0	**97.8**	90.3
person	98.2	99.9	-	-	98.2	99.9
voice	99.8	97.7	-	-	99.8	97.7
aspect	99.0	99.9	-	-	99.0	99.9

Table XIV tests the system on peer reviewed examples from Wikipedia [11] which contains the diacritized text and the corresponding expected IPA transcription.

[11] https://en.wikipedia.org/wiki/Varieties_of_Arabic#Typological_differences, https://en.wikipedia.org/wiki/Arabic_phonology

TABLE XII. CONTEXT-DEPENDENT PRONUNCIATION EXAMPLES; EXPECTED VS OUTPUT

Test phonetic transcription: characters with pronunciation affected by context are colored red.
Φ: zero duration pause, μ: short duration pause, ω: medium duration pause, α: long duration pause, ‿: continuation, •: not pronounced

Phrase	Expected IPA	Output(s)
Definite Article (ال)		
وَالقَهْرِ؛	walqahr ‖	walqahΦr • α
القَهْرِ؛	ʔalqahr ‖	ʔalqahΦr • α
وَالتَّمَرْ	watːamar	wa • tːamarΦ
التَّمَرْ	ʔatːamar	ʔatːamarΦ
هٰذَا الكِتَابُ	haːða‿lkitaːbu	haːða‿lkitaːbu
دَرَسُوا الكِتَابَ	darasu‿lkitaːba	darasu‿lkitaːba
فِي الكِتَابِ	fi‿lkitaːbi	fi‿lkitaːbi
إِلَى الاِعْتِرَافِ	ʔila‿lʔiʕtiraːfi	ʔil • a‿lʔiʕΦtiraːfi
Hamza and her sisters (ء, أ, إ, ؤ, ئ)		
الْمَسَاء	ʔalmasaːʔ	ʔalmasaːʔ
مَقْرُوءَة؛	maqruːʔah ‖	maqΦruːʔah α
بَرِيء	bariːʔ	bariːʔ
أَكَل	ʔakal	ʔakal
أُكِل	ʔukil	ʔukil
طَـأْطَـأَ	tˤaʔtˤaʔa	liʔanːa
لِأَنَّ	liʔanːa	liʔanːa
الْإِسلَام	ʔalʔislaːm	ʔalʔislaːm
إِسلام	ʔislaːm	ʔislaːm
كُؤُوس	kuʔuːs	kuʔuːs
فُـؤَاد	fuʔaːd	fuʔaːd
بُـؤْبُـؤْ	buʔbuʔ	buʔΦbuʔΦ
فِـئَـة	fiʔah	fiʔah
كَـئِـيب	kaʔiːb	kaʔiːb
بريئَـة؛	bariːʔah ‖	briːʔah α
Lam (ل)		
أَلْـعَب	ʔalʕab	ʔalΦʕab
اِلْـتَمَس	ʔiltamas	ʔilΦtamas
لَـعَب	laʕab	laʕab
تَـل	tal	tal
مَال	maːl	maːl
وَالْـوُد	walwud	walwud
وَالـوُد	walwud	walwud
وَالـتِّين	watːiːn	wa • tːiːn
الـتِّينَ	ʔatːiːna	ʔatːiːna
Waw (و) and Ya (ي)		
تُـوت	tuːt	tuːt
مَـوز	mawz	mawz
وَاحِد	waːħid	waːħid
تِـيـن	tiːn	tiːn
بَـيَّـت	bajːat	bajːat
يَـد	jad	jad

Phrase	Expected IPA	Output(s)
Alif maqsurah (ى)		
هُدَى القُلُوبْ	huda‿lquluːb	hud • a‿lquluːbΦ
زُرْتُ هُدَى؛	zurtu hudaː ‖	zurΦtu μ hud • aː α
لَمَــى بِنْتٌ جَمِيلَة.	lamaː bintun d͡ʒamiːlah ‖	lam • aː μ binΦtun μ d͡ʒamiːlah α
Alif mad (آ)		
آبَارْ	ʔaːbaːr	ʔaːbaːrΦ
مَلآنْ	malʔaːn	malΦʔaːnΦ
Madd (ٰ)		
هٰـذَا	haːðaː	haːð • aː
Alif (ا)		
وَاقْرَأ	waqraʔ	wa • qΦraʔ μ
اِقْرَأ؛	ʔiqraʔ ‖	ʔiqΦraʔ α
الْبَـاب؛	ʔalbaːb ‖	ʔalbaːb α
كَتَبُوا	katabuː	katabuː
مَـا اِسْمُك؟	ma‿smuk ‖	ma‿ • sΦmuk α
كَرِيمًـا	kariːman	kariːman
كَرِيمًـا؟	kariːmaː ‖	kariːmaː α
Harakat (ـَـِـُ)		
بـَابَ	baːba	baːba
بـَرْدٍ	bardin	barΦdin μ
أَبَ؛	ʔab ‖	ʔab • α
أَبِ؛	ʔab ‖	ʔab • α
تـُوتُ	tuːtu	tuːtu
كُلِّ	kulːi	kulːi μ
أَبُ؛	ʔab ‖	ʔab • α
مُدَرِّسًا	mudarːisan	mudarːisan μ
بِمَا	bimaː	bim • aː
حِينَ	ħiːna	ħiːna
Tanween (ـًـٍـٌ)		
بَابـًا	baːban	μ baːban μ
بَابـًا.	baːbaː ‖	baːbaː α
بَابٌ	baːbun	μ baːbun μ
بَابٌ.	baːb ‖	baːbΦ α
بَابٍ	baːbin	μ baːbin μ
بَابٍ .	baːb ‖	baːbΦ α
أَيٍّ	ʔajːin	ʔajːin
مِنًى	minan	minan
مِنًى؟	minaː	minaː α
Ta' Marbutah (ة)		
زُرْتُ المَدْرَسَـةَ وَفَرِحْتُ.	zurtu‿lmadrasata wafariħt ‖	zurΦtu‿lmadΦrasata μ wafariħΦt • α
الْمَدْرَسَـةْ.	ʔalmadrasah ‖	ʔalmadΦrasahΦ α
زُرْتُ المَدْرَسَـةَ.	zurtul‿madrasah ‖	zurΦtu‿lmadΦrasah • α

TABLE XIII. EVALUATION OF FULLY DIACRITIZED TASHKEELA SENTENCES. IPA IS THE EXPECTED PHONETIC TRANSCRIPTION FROM A LANGUAGE SPECIALIST, OUT IS THE SYSTEM OUTPUT, AND DIFFER EXPLAINS THE VARIANCE BETWEEN IPA AND OUT. AS ILLUSTRATED IN THE EXAMPLES, IT CAN BE SEEN THAT THE SYSTEM PERFORMED WELL ON A LARGE VARIETY OF TEXTS. NOTE THAT WE REMOVED /•/ AND /Φ/ FOR READABILITY; THE DIFFERENCES ARE IN RED

Φ: zero duration pause, μ: short duration pause, ω: medium duration pause, α: long duration pause, ‿: continuation, •: not pronounced; In IPA |means short stop, and ‖means long stop	
Input	وَيُعَانِي الْوَطَنُ الْعَرَبِيُّ بِشِدَّةٍ مِنْ هَذِهِ الظَّاهِرَةِ بِسَبَبِ وُقُوعِهِ ضِمْنَ النِّطَاقِ الصَّحْرَاوِيِّ وَشِبْهِ الصَّحْرَاوِيِّ الْمُمْتَدِّ مِنْ شَمَالِ أَفْرِيقْيَا إِلَى آسْيَا. وَتُشَكِّلُ نِسْبَةُ الْمِسَاحَاتِ الْمُتَصَحِّرَةِ وَالْأَرَاضِي الْقَاحِلَةِ فِي الْمَنْطِقَةِ حَوَالَيْ مِنْ إِجْمَالِيِّ الْمِسَاحَةِ الْكُلِّيَّةِ، أَيْ حَوَالَيْ مِنْ إِجْمَالِيِّ الْمَنَاطِقِ الْمُتَصَحِّرَةِ عَلَى مُسْتَوَى الْعَالَمِ.
IPA	wajuʕa:ni‿lwatˤanu‿lʕarabij:u biʃid:atin min haðihi‿ðˤ:a:hirati bisababi wuqu:ʕihi dˤimna‿n:itˤa:qi‿sˤ:aħra:wij:i waʃibhi‿sˤ:aħra:wij:i‿lmumtad:i min ʃama:li ʔafri:qja: ʔila: ʔa:sja: α watuʃak:ilu nisbatu‿lmisa:ħa:ti‿lmutasˤaħ:irati walʔara:dˤi‿lqa:ħilati fi‿lmantˤiqati ħawa:laj min ʔid͡ʒma:lij:i‿lmisa:ħati‿lkul:ij:ah |ʔaj ħawa:laj min ʔid͡ʒma:lij:i‿lmana:tˤiqi‿lmutasˤaħ:irati ʕala: mustawa‿lʕa:lam ‖
Out	μ wajuʕa:ni‿lwatˤanu‿lʕarabij:u μ biʃid:atin μ min μ haðihi‿ðˤ:a:hirati μ bisababi μ wuqu:ʕihi μ dˤimna‿n:itˤa:qi‿sˤ:aħra:wij:i μ waʃibhi‿sˤ:aħra:wij:i‿lmumtad:i μ min μ ʃama:li μ ʔafri:qja: μ ʔila: μ ʔa:sja: α μ watuʃak:ilu μ nisbatu‿lmisa:ħa:ti‿lmutasˤaħ:irati μ walʔara:dˤi‿lqa:ħilati μ fi‿lmantˤiqati μ ħawa:laj μ min μ ʔid͡ʒma:lij:i‿lmisa:ħati‿lkul:ij:ah ω μ ʔaj μ ħawa:laj μ min μ ʔid͡ʒma:lij:i‿lmana:tˤiqi‿lmutasˤaħ:irati μ ʕala: μ mustawa‿lʕa:lam α
Differ	None
Input	وَالطَّحَالِبُ نَبَاتَاتٌ بَحْرِيَّةٌ بَسِيطَةُ التَّكْوِينِ، مُعْظَمُهَا قَادِرٌ عَلَى إِجْرَاءِ عَمَلِيَّةِ التَّمْثِيلِ الضَّوْئِيِّ، حَيْثُ تَسْتَطِيعُ أَنْ تُنْتِجَ زَيْتًا نَبَاتِيًّا، تَتِمُّ مُعَالَجَتُهُ كِيمْيَائِيًّا لِلْحُصُولِ عَلَى الدِّيزَل الْحَيَوِيِّ، الْقَادِرِ عَلَى تَشْغِيلِ كَثِيرٍ مِنَ الْمُحَرِّكَاتِ.
IPA	watˤ:aħa:libu naba:ta:tun baħrij:atun basi:tˤatu‿t:akwi:n |muʕðˤamuha: qa:dirun ʕala: ʔid͡ʒra:ʔi ʕamalij:ati‿t:amθi:li‿dˤ:awʔij: |ħajθu tastatˤi:ʕu ʔan tuntid͡ʒa zajtan naba:tij:a: |tatim:u muʕa:lad͡ʒatuhu ki:mja:ʔij:an lilħusˤu:li ʕala‿d:i:zal‿lħajawij: |ʔalqa:diri ʕala: taʃʁi:li kaθi:rin mina‿lmuħar:ika:t ‖
Out	watˤ:aħa:libu μ naba:ta:tun μ baħrij:atun μ basi:tˤatu‿t:akwi:n ω μ muʕðˤamuha: μ qa:dirun μ ʕala: μ ʔid͡ʒra:ʔi μ ʕamalij:ati‿t:amθi:li‿dˤ:awʔij: ω μ ħajθu μ tastatˤi:ʕu μ ʔan μ tuntid͡ʒa μ zajtan μ naba:tij:a: ω μ tatim:u μ muʕa:lad͡ʒatuhu μ ki:mja:ʔij:an μ lilħusˤu:li μ ʕala‿d:i:zal μ lħajawij: ω μ ʔalqa:diri μ ʕala: μ taʃʁi:li μ kaθi:rin μ mina‿lmuħar:ika:t α
Differ	Due to the lack of a word-final diacritic in الدِّيزَل, our system does not connect it to the next word even though it pronounces it correctly as seen. Please note that الدِّيزَل is a loan word "the diesel" ʕala‿d:i:zal μ lħajawij: ʕala‿d:i:zal‿lħajawij:
Input	يُذْكَرُ أَنَّ هٰذِهِ الْقَائِمَةَ تُسَلِّطُ الضَّوْءَ كُلَّ عَامٍ عَلَى الْأَشْخَاصِ الَّذِينَ أَسْهَمُوا فِي مَجَالَاتِ السِّيَاسَةِ وَالرِّيَاضَةِ وَالْفَنِّ وَالْأَعْمَالِ وَغَيْرِهَا فِي جَمِيعِ أَنْحَاءِ الْعَالَمِ.
IPA	juðkaru ʔan:a ha:ðihi‿lqa:ʔimata tusal:itˤu‿dˤ:awʔa kul:a ʕa:min ʕala‿lʔaʃχa:sˤi‿l:aði:na ʔashamu: fi: mad͡ʒa:la:ti‿s:ija:sati wari:ja:dˤati walfan:i walʔaʕma:li waʁajriha: fi: d͡ʒami:ʕi ʔanħa:ʔi‿lʕa:lam ‖
Out	juðkaru μ ʔan:a μ ha:ðihi‿lqa:ʔimata μ tusal:itˤu‿dˤ:awʔa μ kul:a μ ʕa:min μ ʕala‿lʔaʃχa:sˤi‿l:aði:na μ ʔashamu: μ fi: μ mad͡ʒa:la:ti‿s:ija:sati μ wari:ja:dˤati μ walfan:i μ walʔaʕma:li μ waʁajriha: μ fi: μ d͡ʒami:ʕi μ ʔanħa:ʔi‿lʕa:lam α
Differ	None
Input	وَيُوجَدُ بِجَامِعِ الْقَرَوِيِّينَ بَابًا، مِنْ أبْرَزِهَا بَابُ الشَّمَّاعِينَ وَهُوَ الْبَابُ الرَّئِيسِيُّ، وَبَابُ الْحَفَاةِ، وَبَابُ الْوَرْدِ.
IPA	waju:d͡ʒadu bid͡ʒa:miʕi‿lqarawij:i:na ba:ba: |min ʔabraziha: ba:bu‿ʃ:am:a:ʕi:na wahuwa‿lba:bu‿r:aʔi:sij: |waba:bu‿lħufa:h |waba:bu‿lward ‖
Out	waju:d͡ʒadu μ bid͡ʒa:miʕi‿lqarawij:i:na μ μ ba:ba: ω μ min μ ʔbraziha: μ ba:bu‿ʃ:am:a:ʕi:na μ wahuwa‿lba:bu μ r:aʔi:sij: ω μ waba:bu‿lħufa:h ω μ waba:bu‿lward α
Differ	The reason behind the difference in أبْرَزِهَا pronunciation of the system and the expected pronunciation was the lack of a diacritic on the Alif أ thus making it an incomplete diacritization of the word, which was beyond our scope
Input	. الْكَلْبُ أكْبَرُ مِنَ الْقِطِّ
IPA	ʔalkalbu ʔakbaru mina‿lqitˤ:
Out	ʔalkalbu μ ʔkbaru μ mina‿lqitˤ: α
Differ	ʔakbaru vs ʔkbaru due to the lack of diacritic on أ
Input	طَائِرَةٌ رَائِعَةُ التَّصْمِيمِ تَخْتَرِقُ سُرْعَتُهَا حَاجِزَ الصَّوْتِ وَتَتَجَاوَزُهُ لِلضِّعْفِ. اِخْتِصَارُهَا الْمُدَّةَ الزَّمَنِيَّةَ الْمُسْتَغْرِقَةَ فِي الطَّيَرَانِ إِلَى أَقَلَّ مِنَ النِّصفِ أَدَّى إِلَى إِحْدَاثِهَا ثَوْرَةً فِي مَجَالِ النَّقْلِ الْجَوِّيِّ. وَلَكِنْ، تَوَقُّفُ الْعَمَلِ بِهَا نِهَائِيًّا عَامَ ، فَمَا هُوَ السَّبَبُ؟
IPA	tˤa:ʔiratun ra:ʔiʕatu‿t:asˤmi:mi taχtariqu surʕatuha: ħa:d͡ʒiza‿sˤ:awti watatad͡ʒa:wazuhu lidˤ:iʕf ‖ʔiχtisˤa:ruha‿lmud:ata‿z:amanij:ata‿lmustaʁriqata fi‿tˤ:ajara:ni ʔila: ʔaqal:i mina‿n:isˤfi ʔad:a: ʔila: ʔiħda:θiha: θawratan fi: mad͡ʒa:li‿n:aqli‿ld͡ʒaw:ij: ‖wala:kin |tawaq:afa‿lʕamalu biha: niha:ʔij:an ʕa:ma |fama: huwa‿s:abab ‖
Out	tˤa:ʔiratun μ ra:ʔiʕatu‿t:asˤmi:mi μ taχtariqu μ surʕatuha: μ ħa:d͡ʒiza‿sˤ:awti μ watatad͡ʒa:wazuhu μ lidˤ:iʕf α μ χtisˤa:ruha‿lmud:ata‿z:amanij:ata‿lmustaʁriqata μ fi‿tˤ:ajara:ni μ ʔila: μ ʔaqal:i μ mina‿n:isˤfi μ ʔad:a: μ ʔila: μ ʔiħda:θiha: μ θawratan μ fi: μ mad͡ʒa:li‿n:aqli‿ld͡ʒaw:ij: α μ walakin ω μ tawaq:afa‿lʕamalu μ biha: μ niha:ʔij:an μ ʕa:ma μ ω μ fama: μ huwa‿s:abab α
Differ	The lack of mad in the spelling of لَكِن is the reason our output did not pronounce the long vowel /a:/ in the word and instead pronounced it as the short vowel /a/
Input	. الدَّانِمَارْكُ رَفَضَتْ هَذِهِ السَّنَةَ اِسْتِقْبَالَ لَاجِئًا كَانُوا ضِمْنَ حِصَّتِهَا لِهَذَا الْعَامِ
IPA	ʔad:a:nima:rku rafadˤat ha:ðihi‿s:anata‿stiqba:la la:d͡ʒiʔan ka:nu: dˤimna ħisˤ:atiha: liha:ða‿lʕa:m ‖
Out	ʔad:a:nima:rku μ rafadˤat μ haðihi‿s:anata‿stiqba:la μ μ la:d͡ʒiʔan μ ka:nu: μ dˤimna μ ħisˤ:atiha: μ lihaða‿lʕa:m α
Differ	The same word هَذَا is misspelled twice and lacks the mad which gives the long vowel pronunciation ha:ðihi vs haðihi
Input	وَأَكَّدَ الْأَكَادِيمِيُّ الْفِلَسْطِينِيُّ الْمُنَسِّقُ الْعَامُّ لِلْمُؤْتَمَرِ يَحْيَى جَبْر أَنَّهُ يَأْتِي بِتَنْظِيمٍ مِنْ جَامِعَةِ النَّجَاحِ الْوَطَنِيَّةِ بِنَابُلُسَ وَدَائِرَةِ شُؤُونِ الْمُغْتَرِبِينَ بِمُنَظَّمَةِ التَّحْرِيرِ الْفِلَسْطِينِيَّةِ.
IPA	waʔak:ada‿lʔaka:di:mij:u‿lfilastˤi:nij:u‿lmunas:iqu‿lʕa:m:u lilmuʔtamari jaħja: d͡ʒabr ʔan:ahu jaʔti: bitanðˤi:min min d͡ʒa:miʕati‿n:ad͡ʒa:ħi‿lwatˤanij:ati bina:bulusa wada:ʔirati ʃuʔu:ni‿lmuʁtaribi:na bimunaðˤ:amati‿t:aħri:ri‿lfilastˤi:nij:ah ‖
Out	waʔak:ada‿lʔaka:di:mij:u‿lfilastˤi:nij:u‿lmunas:iqu‿lʕa:m:u μ lilmuʔtamari μ jaħja: μ d͡ʒabr μ ʔan:ahu μ jaʔti: μ bitanðˤi:min μ min μ d͡ʒa:miʕati‿n:ad͡ʒa:ħi‿lwatˤanij:ati μ bina:bulusa μ wada:ʔirati μ ʃuʔu:ni‿lmuʁtaribi:na μ bimunaðˤ:amati‿t:aħri:ri‿lfilastˤi:nij:ah α
Differ	None
Input	تَدْرُسُ شَرِكَةُ غُوغِل السَّمَاحَ بِإِنْشَاءِ حِسَابَاتٍ عَلَى الْإِنْتَرْنِت لِلْأَطْفَالِ تَحْتَ سِنِّ عَامًا، وَمَنْحَ آبَائِهِم الْقُدْرَةَ عَلَى التَّحَكُّمِ فِي كَيْفِيَّةِ اسْتِخْدَامِ هَذِهِ الْخِدْمَةِ.
IPA	tadrusu ʃarikatu ʁu:ʁil‿s:ama:ħa biʔinʃa:ʔi ħisa:ba:tin ʕala‿lʔintarnit lilʔatˤfa:li taħta sin:i ʕa:ma: |wamanħa ʔa:ba:ʔihima‿lqudrata ʕala‿t:aħak:umi fi: kajfij:ati‿stiχda:mi ha:ðihi‿lχidmah ‖
Out	tadrusu μ ʃarikatu μ ʁu:ʁl μ s:ama:ħa μ biʔinʃa:ʔi μ ħisa:ba:tin μ ʕala‿lʔintarnit μ lilʔatˤfa:li μ taħta μ sin:i μ μ ʕa:ma: ω μ wamanħa μ ʔa:ba:ʔihim μ lqudrata μ ʕala‿t:aħak:umi μ fi: μ kajfij:ati‿stiχda:mi μ haðihi‿lχidmah α
Differ	The main reason for the differences between expected and out in this example is a lack of diacritics, the existence of loan words (غُوغِل "google") and the lack of mad (') in هَذِهِ

TABLE XIV. EVALUATION OF THE FULLY DIACRITIZED WIKIPEDIA SENTENCES. IPA IS THE EXPECTED PHONETIC TRANSCRIPTION FROM WIKIPEDIA, OUT IS THE SYSTEM OUTPUT, AND DIFFER EXPLAINS THE VARIANCE BETWEEN IPA AND OUT. AS ILLUSTRATED BELOW, THE SYSTEM ACHIEVED A PERFECT SCORE ON THE EXAMPLES. PLEASE NOTE WE REMOVED /Φ/ AND /•/ FROM THE OUTPUT TO IMPROVE READABILITY

<table>
<tr><td colspan="2">Φ: zero duration pause, μ: short duration pause, ω: medium duration pause, α: long duration pause, ‿: continuation, •: not pronounced; In IPA |means short stop and ||means long stop</td></tr>
<tr><td>Input</td><td>كَانَتْ رِيحُ الشَّمَالِ تَتَجَادَلُ وَالشَّمْسَ فِي أَيٍّ مِنْهُمَا كَانَتْ أَقْوَى مِنَ الأُخْرَى، وَإِذْ بِمُسَافِرٍ يَطْلُعُ مُتَلَفِّعًا بِعَبَاءَةٍ سَمِيكَةٍ. فَاتَّفَقَتَا عَلَى اعْتِبَارِ السَّابِقِ فِي إِجْبَارِ المُسَافِرِ عَلَى خَلْعِ عَبَاءَتِهِ الأَقْوَى. عَصَفَتْ رِيحُ الشَّمَالِ بِأَقْصَى مَا اسْتَطَاعَتْ مِن قُوَّةٍ. وَلَكِنْ كُلَّمَا ازْدَادَ العَصْفُ ازْدَادَ المُسَافِرُ تَدَثُّرًا بِعَبَاءَتِهِ، إِلَى أَنْ أُسْقِطَ فِي يَدِ الرِّيحِ فَتَخَلَّتْ عَنْ مُحَاوَلَتِهَا. بَعْدَئِذٍ سَطَعَتِ الشَّمْسُ بِدِفْئِهَا، فَمَا كَانَ مِنَ المُسَافِرِ إِلَّا أَنْ خَلَعَ عَبَاءَتَهُ عَلَى التَّوِّ. وَهَكَذَا اضْطُرَّتْ رِيحُ الشَّمَالِ إِلَى الاِعْتِرَافِ بِأَنَّ الشَّمْسَ كَانَتْ هِيَ الأَقْوَى.</td></tr>
<tr><td>IPA</td><td>kaːnat riːħu‿ʃːamaːli tatad͡ʒaːdalu wa‿ʃːamsa fiː ʔajːin minhumaː kaːnat ʔaqwaː mina‿lʔuxraː |wa‿ʔið bimusaːfirin jatˤluʕu mutalafːiʕan biʕabaːʔatin samiːkah ||fatːafaqataː ʕala‿ʕtibaːri‿sːaːbiqi fiː ʔid͡ʒbaːri‿lmusaːfiri ʕalaː xalʕi ʕabaːʔatihi‿lʔaqwaː ||ʕasˤafat riːħu‿ʃːamaːli biʔaqsˤaː ma‿statˤaːʕat min quwːah ||wa‿laːkin kulːama‿zdaːda‿lʕasˤfu‿zdaːda‿lmusaːfiru tadaθːuran biʕabaːʔatih |ʔilaː ʔan ʔusqitˤa fiː jadi‿rːiːħi fataxalːat ʕan muħaːwalatihaː ||baʕdaʔiðin satˤaʕati‿ʃːamsu bidifʔihaː |fa‿maː kaːna mina‿lmusaːfiri ʔilːaː ʔan xalaʕa ʕabaːʔatahu ʕala‿tːaw: ||wa‿haːkaða‿dˤtˤurat riːħu‿ʃːamaːli ʔila‿lʔiʕtiraːfi biʔanːa‿ʃːamsa kaːnat hija‿lʔaqwaː</td></tr>
<tr><td>Out</td><td>kaːnat μ riːħu‿ʃːamaːli μ tatad͡ʒaːdalu μ waʃːamsa μ fiː μ ʔajːin μ minhumaː μ kaːnat μ ʔaqwaː μ mina‿lʔuχraː ω μ waʔið μ bimusaːfirin μ jatˤluʕu μ mutalafːiʕan μ biʕabaːʔatin μ samiːkah α μ fatːafaqataː μ ʕala‿ʕtibaːri‿sːaːbiqi μ fiː μ ʔid͡ʒbaːri‿lmusaːfiri μ ʕalaː μ χalʕi μ ʕabaːʔatihi‿lʔaqwaː α μ ʕasˤafat μ riːħu‿ʃːamaːli μ biʔaqsˤaː μ ma‿statˤaːʕat μ min μ quwːah α μ walaːkin μ kulːama‿zdaːda‿lʕasˤfu‿zdaːda‿lmusaːfiru μ tadaθːuran μ biʕabaːʔatih ω μ ʔilaː μ ʔan μ ʔusqitˤa μ fiː μ jadi‿rːiːħi μ fataχalːat μ ʕan μ muħaːwalatihaː α μ baʕdaʔiðin μ satˤaʕati‿ʃːamsu μ bidifʔihaː ω μ famaː μ kaːna μ mina‿lmusaːfiri μ ʔilːaː μ ʔan μ χalaʕa μ ʕabaːʔatahu μ ʕala‿tːaw: α μ wahaːkaða‿dˤtˤurat μ riːħu‿ʃːamaːli μ ʔila‿lʔiʕtiraːfi μ biʔanːa‿ʃːamsa μ kaːnat μ hija‿lʔaqwaː α</td></tr>
<tr><td>Differ</td><td>None</td></tr>
</table>

<table>
<tr><td>Input</td><td>عِنْدَمَا ذَهَبْتُ إِلَى المَكْتَبَة.</td></tr>
<tr><td>IPA</td><td>ʕindamaː ðahabtu ʔila‿lmaktabah ||</td></tr>
<tr><td>Out</td><td>ʕindamaː μ ðahabtu μ ʔila‿lmaktabah α</td></tr>
<tr><td>Differ</td><td>None</td></tr>
</table>

<table>
<tr><td>Input</td><td>عِنْدَمَا ذَهَبْتُ إِلَى المَكْتَبَة.</td></tr>
<tr><td>IPA</td><td>ʕindamaː ðahabtu ʔila‿lmaktabah ||</td></tr>
<tr><td>Out</td><td>ʕindamaː μ ðahabtu μ ʔila‿lmaktabah α</td></tr>
<tr><td>Differ</td><td>None</td></tr>
</table>

<table>
<tr><td>Input</td><td>لَمْ أَجِد سِوَى هٰذَا الكِتَابِ القَدِيم</td></tr>
<tr><td>IPA</td><td>lam ʔad͡ʒid siwaː haːða‿lkitaːbi‿lqadiːm</td></tr>
<tr><td>Out</td><td>lam μ ʔad͡ʒid μ siwaː μ haːða‿lkitaːbi‿lqadiːm</td></tr>
<tr><td>Differ</td><td>None</td></tr>
</table>

<table>
<tr><td>Input</td><td>كُنْتُ أُرِيدُ أَنْ أَقْرَأَ كِتَابًا عَن تَارِيخِ المَرأَةِ فِي فَرَنسَا</td></tr>
<tr><td>IPA</td><td>kuntu ʔuriːdu ʔan ʔaqraʔa kitaːban ʕan taːriːχi‿lmarʔati fiː faransaː</td></tr>
<tr><td>Out</td><td>kuntu μ ʔuriːdu μ ʔan μ ʔaqraʔa μ kitaːban μ ʕan μ taːriːχi‿lmarʔati μ fiː μ faransaː</td></tr>
<tr><td>Differ</td><td>None</td></tr>
</table>

<table>
<tr><td>Input</td><td>أَنَا أُحِبُّ القِرَاءَةَ كَثِيرًا</td></tr>
<tr><td>IPA</td><td>ʔanaː ʔuħibːu‿lqiraːʔata kaθiːran</td></tr>
<tr><td>Out</td><td>ʔanaː μ ʔuħibːu‿lqiraːʔata μ kaθiːran</td></tr>
<tr><td>Differ</td><td>None</td></tr>
</table>

Supplement 1: Arabic Orthography, Phonology, and Morphology Compendium

***Abstract*—Concise and specific information on modern standard Arabic (MSA), which can also be used as a standalone MSA reference, is given as a background to the main text and covers topics, such as textual marks and fermatas that significantly affect the phonetic transcription of text transformations that are not normally discussed in other texts in English. Minimal consonant pairs, geminates, and vowels to indicate phonemic contrasts are also given. To avoid confusion, the original Arabic symbols were retained rather than using Roman transliteration. Other than minimal pairs, the compendium was compiled from various Arabic references (7; 30).**

A Orthography

Modern Standard Arabic (MSA) is written cursively from right to left, with the letter shapes changing according to the position. MSA has forty-six characters; twenty-eight alphabet letters, ten non-alphabet letters, and eight diacritics; and ligatures (لإ,لا,لآ,لأ), which are letter combinations used when writing but are not counted as characters because they are only graphical representations. Tables XV and XVI show the MSA script alphabet and non-alphabet letters and the diacritics. Table XVI presents the alphabet letter shapes based on their position in a word, an example word containing the alphabet letter, its IPA transcription using segmental phonology, and the word meaning.

Table XVI introduces the ten non-alphabet letters, four (أ, إ, ئ, ؤ) of which are part of the five-member "Hamza sisters," and the fifth being the alphabet letter (ء). Five of the non-alphabet letters (ا, ى, ة, ٰ, آ) produce complexities as they map to one or more sounds based on their position in the word and the surrounding words; this is discussed in more detail in later sections. Kasheeda (ـ) is used for graphical justification and elongation and has no underlying pronunciation. The letter mad (ٓ) is part of MSA but is non-existent on computer keyboards and, therefore, is missing in modern texts and needs to be added to the thirteen words that contain it before any processing.

Table XVII lists the diacritics, which may not be at the start of a word, and divides the diacritics into three categories: Harakah (ـِ, ـُ, ـَ), Tanween (ـٍ, ـٌ, ـً), and other (Shaddah ـّ; Sukoon ـْ). A Harakah character is pronounced as a short vowel, a Tanween character is pronounced as a combination of a short vowel and /n/ and can only occur at the end of a word, Sukoon (ـْ) is a zero-length pause, and as Shaddah (ـّ) indicates the gemination of the sound it follows, it cannot be preceded by a diacritic.

Diacritics cannot be consecutive, except for Shaddah (ـّ), which is generally followed by Harakah and sometimes by Tanween or Sukoon. Other rules restrict a sequence of characters; for example, إ can only be followed by a kasra, أ is only followed by Sukoon, dhamma, or fatha, ؤ may not be followed by kasra, shaddah, and ya' and can only be preceded by dhama, ئ cannot be followed by a shaddah and is only preceded by kasra, and ء cannot start a word.

Five MSA letter groupings are related to pronunciation: Hamza, Wasl, Solar, and Lunar. The Hamza set {أ إ ؤ ء ئ} are pronounced as glottal stops, and the Wasl symbols {و ب ت ل ك} are characters that affect the pronunciation of ا; for example, if any of the Wasl symbols precede ا when ا is at the start of the word, then ا is not pronounced.

The Solar set {ت ث د ذ ر ز س ش ص ض ط ظ ن ل} and the Lunar set {أ إ ء ي و ه م ك ق ف غ ع خ ح ج ب} are grouped because the Solar and Lunar letters affect the pronunciation of ل in a very specific character environment (when it is part of the definite article in Arabic: ال); for example, when a Solar letter follows ل, then ل is not pronounced, but when a Lunar letter follows ل, it is pronounced.

It is also important to describe the text markings as they not only correspond to "sounds" but also regulate the contextual pronunciation of the consonants as detailed in the rules presented in the main manuscript. Table XV details the marks and their corresponding fermatas.

Table XV. Marks and their Corresponding Pauses and Continuations

Description	Mark	Mark symbol	Fermata	IPA symbol	System symbol
Between paragraphs, sentences	End of File, Start of File tab, new line	eof, sof \t, \n	Long duration pause	‖ or (...)	α
Between phrases	. ! : " ' -	!;.? - ' ",:	Medium duration pause	| or (.)	ω
Between words	space	\s	Short duration pause	(.)	μ
Within word	Not applicable	Not applicable	Zero duration pause		•
Connected words	space	\s	Continuation	‿	‿

Table XVI. Alphabetic Letters and Their Shapes in Different Word Positions. I: Isolated, B: Beginning, M: Middle, and E: End. Example Words and Meanings (gloss) are given in IPA with their Broad Segmental Pronunciation Transcriptions

Alphabet Letters

I	B	M	E	Word	IPA	Gloss
ء			ء	شعراء	/ʃuʕaraːʔ/	poets
ب	بـ	ـبـ	ـب	بَيْتْ	/bajt/	house
ت	تـ	ـتـ	ـت	تل	/tal/	hill
ث	ثـ	ـثـ	ـث	ثاني	/θaːniː/	second
ج	جـ	ـجـ	ـج	جلد	/d͡ʒild/	leather
ح	حـ	ـحـ	ـح	حليب	/ħaliːb/	milk
خ	خـ	ـخـ	ـخ	خل	/χal/	vinegar
د	د	ـد	ـد	دب	/dub/	bear
ذ	ذ	ـذ	ـذ	ذرة	/ðurah/	corn
ر	ر	ـر	ـر	رز	/ruz/	rice
ز	ز	ـز	ـز	زعتر	/zaʕtar/	thyme
س	سـ	ـسـ	ـس	سُلَّم	/sulːam/	ladder
ش	شـ	ـشـ	ـش	شلال	/ʃalːal/	water fall
ص	صـ	ـصـ	ـص	صبْر	/sˤabr/	patience

Alphabet Letters

I	B	M	E	Word	IPA	Gloss
ض	ضـ	ـضـ	ـض	ضفدع	/dˤufdaʕ/	frog
ط	طـ	ـطـ	ـط	طائرة	/tˤaːʔirah/	plane
ظ	ظـ	ـظـ	ـظ	ظالم	/ðˤaːlim/	who wrongs
ع	عـ	ـعـ	ـع	عين	/ʕajn/	eye
غ	غـ	ـغـ	ـغ	غدّة	/ʁudːah/	gland
ف	فـ	ـفـ	ـف	فيل	/fiːl/	elephant
ق	قـ	ـقـ	ـق	قلم	/qalam/	pen
ك	كـ	ـكـ	ـك	كتاب	/kitaːb/	book
ل	لـ	ـلـ	ـل	لون	/lawn/	color
م	مـ	ـمـ	ـم	موز	/mawz/	banana
ن	نـ	ـنـ	ـن	نمر	/nimr/	tiger
ه	هـ	ـهـ	ـه	هلال	/hilaːl/	crescent
و	و	ـو	ـو	ولد	/walad/	boy
ي	يـ	ـيـ	ـي	يوم	/jawm/	day

Non-alphabet Letters

I	B	M	E	Word	IPA	Gloss
آ	آ	ـآ,آ	ـآ,آ	آسيا	/ʔaːsjaː/	Asia
ؤ		ـؤ، ؤ	ـؤ، ؤ	مؤمن	/muʔmin/	faithful
ئ		ـئـ، ئـ	ـئ، ئ	طائرة	/tˤaːʔirah/	plane
أ	أ	ـأ, أ	ـأ, أ	أكتب	/ʔaktub/	I write
إ	إ	ـإ, إ	ـإ, إ	إحسان	/ʔiħsaːn/	goodness

Non-alphabet Letters

I	B	M	E	Word	IPA	Gloss
ا	ا	ـا ، ا	ـا ، ا	اختبار	/ʔiχtibaːr/	exam
ى			ـى، ى	أنثى	/ʔunθaː/	female
ة			ـة، ة	طائرة	/tˤaːʔirah/	plane
			ـٰ	هـٰذا	/haːðaː/	this 'male'
			ـ	بـــسم	/bism/	in the name of

Table XVII. Diacritics with Example usage, their Broad Segmental Pronunciation Transcriptions in IPA, and Meanings

Diacritics		Usage	IPA	Gloss
Fatha	َ	وَلَدْ	/walad/	a boy
Tanween Fath	ً	اِسْمًا	/ʔisman/	a name
Dhamma	ُ	دُبْ	/dub/	a bear
Tanween Dham	ٌ	رُزٌّ	/ruzun/	rice

Diacritics		Usage	IPA	Gloss
Kasra	ِ	فِعِلْ	/fiʕil/	a verb
Tanween Kasr	ٍ	فِعْلٍ	/fiʕlin/	a verb
Shaddah	ّ	غُدَّةْ	/ʁudːah/	a gland
Sukoon	ْ	فِعِلْ	/fiʕil/	a verb

B Segmental Phonology

MSA is a Semitic language with 55 consonants and six vowels. In addition to labio-dental and velar sounds, MSA is rich in glottal, uvular, and pharyngeal sounds such as /ʕ/, and has regular and velarized or pharyngealized pairs, such as /d/

and /dˤ/. The geminated counterparts of the phonemes are also MSA phonemes.

MSA has eight plosives; one bilabial /b/, four alveolars /t/, /d/, /tˤ/, /dˤ/, one velar /k/, one uvular /q/, and one glottal /ʔ/); two nasals; bilabial /m/ and alveolar /n/; one alveolar trill; /r/; thirteen fricatives; one labiodental /f/, three dental /θ/, /ð/, /ðˤ/, three alveolars /s/, /z/, /sˤ/, one postalveolar /ʃ/, two uvular /χ/, /ʁ/, two pharyngeal /ħ/, /ʕ/, and one glottal /h/; two approximants; one bilabial /w/ and one palatal /j/; one postalveolar affricate; /d͡ʒ/; and an alveolar lateral approximant; /l/. MSA has only six long and short vowels: /aː/, /iː/, /uː/, /a/, /i/, /u/. Table XIX shows the consonant chart, with the consonants based on voicing, pharyngealization, place of articulation, and manner of articulation. Table XX enumerates the geminated consonantal phonemes with examples. Table XVIII presents the vowel chart. Minimal pairs that validate the phonemes are in Appendix A.

Phonemes are also grouped based on the pronunciation of their related character: coronals; dental, alveolar, and postalveolar, with /θ/, /ð/, /ðˤ/, /r/, /n/, /t/, /d/, /tˤ/, /dˤ/, /s/, /z/, /sˤ/, /l/, /ʃ/, /d͡ʒ/ being the phonemes the Solar letters map to; and non-coronals; bilabial, labiodental, palatal, velar, uvular, pharyngeal, and glottal, with /b/, /m/, /w/, /f/, /j/, /k/, /q/, /χ/, /ʁ/, /ħ/, /ʕ/, /ʔ/, /h/ being the phonemes Lunar letters map to. Hamzah sisters map to the glottal stop, Harakah map to short vowels, Tanween maps to a short vowel followed by /n/ depending on the context, and Shaddah causes gemination.

Table XVIII. Vowel Chart: Upper Left is Normal Unrounded, Upper Right is Normal Rounded, Lower Left is Lengthened and Unrounded, and the Lower Right is Lengthened and Rounded

	Front	Central	Back
Close (high)	i iː		u uː
Mid			
Open (low)	a aː		

Table XIX. Consonant Chart : The Symbol on the Top Right is Normal Voiced, the Symbol on the Top Left is Normal Unvoiced, the Symbol on the Bottom Right is Pharyngealized Voiced, and the Symbol on the Bottom Left is Pharyngealized Unvoiced

	Bilabial	Labio-dental	Dental	Alveolar	Post-alveolar	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	b			t d tˤ dˤ			k	q		ʔ
Nasal	m			n						
Trill				r						
Fricative		f	θ ð ðˤ	s z sˤ	ʃ			χ ʁ	ħ ʕ	h
Affricate					d					
Approximant	w					j				
Lateral Approximant				l						

C Morphology

Morphology deals with the internal structure of words. More specifically, it dictates the composition of a word from smaller meaningful units called morphemes. There are two approaches to morphology; form-based and functional. Form-based morphology considers the form of the units making up a word, their interactions, and how they relate to the word's overall form. Functional morphology is about the function of the units inside a word and how they affect its overall syntactic and semantic behavior.

Table XX. MSA Geminated Phonemes. All Consonants Except for the Glottal Stop. The Examples below Show Words with Geminated Consonants

Phone	Word	IPA	Gloss	Phone	Word	IPA	Gloss
[bː]	تبًّا	/tabːan/	perish!	[tˤː]	الطَّبيب	/ʔtˤːabiːb/	the doctor
[tː]	التَّل	/ʔtːal/	the hill	[ðˤː]	الظَّل	/ʔðˤːil/	the shadow
[θː]	الثُّلاثاء	/ʔθːulaːθaːʔ/	Tuesday	[ʕː]	لعَّبَ	/laʕːaba/	he played with
[d͡ʒː]	أجَّل	/ʔd͡ʒːala/	delayed	[ʁː]	صغَّر	/saʁːara/	he made smaller
[ħː]	وحَّد	/waħːda/	united	[fː]	أفٍّ	/ʔufːin/	expressing impatience/contempt
[χː]	أخَّر	/ʔaχːara/	delayed	[qː]	وقَّت	/waqːata/	he timed
[dː]	الدَّمار	/ʔdːamaːr/	the destruction	[kː]	أكَّل	/ʔkːala/	he fed
[ðː]	الذَّنب	/ʔðːanb/	the sin	[lː]	اللَّمس	/ʔlːams/	the touch
[rː]	الرُّز	/ʔrːuz/	the rice	[mː]	نمَّام	/namːam/	he who spreads scandals
[zː]	الزَّوج	/ʔzːawd͡ʒ/	the husband	[nː]	النَّمر	/ʔnːimr/	the tiger
[sː]	السَّماء	/ʔsːamaːʔ/	the sky	[hː]	وهَّم	/wahːam/	puzzled somebody
[ʃː]	الشَّمس	/ʔʃːams/	the sun	[wː]	توَّاب	/tawːab/	repentant
[sˤː]	الصَّباح	/ʔsˤːabaːħ/	the morning	[jː]	جيِّد	/d͡ʒajːit/	good
[dˤː]	الضُّفدع	/ʔdˤːufdaʕ/	the frog				

Fig. 5. Arabic word morpheme breakdown. A word is a concatenation of a prefix, stem, and suffix (concatenative). A stem is a meaning-bearing unit that can be further decomposed into its root and pattern (templatic). The root gives the core meaning and the pattern provides the part of speech (POS, category) and other linguistic properties, such as number, tense, and gender. This image uses the Buckwalter transliteration scheme (www.qamus.org/transliteration.htm).

Fig. 5 illustrates the structure of Arabic words. Arabic utilizes form-based morphology and has concatenative and templatic morphemes (smallest units in a word). Concatenative morphology is centered on stems and affixes (prefixes, suffixed, circumfixes), and the morphemes are generally concatenated in a sequence to produce a surface form (word). Morphological grammar that constructs stems from the interdigitation (interleaving) of the root and pattern is called templative morphology. In Arabic, morphological form and function are independent although most templatic processes are derivational and most concatenative processes are inflectional. Derivational functional morphology is concerned with creating new words from other words, and in inflectional morphology, the meaning and part of speech remain the same (31).

The two broad morpheme classes in concatenative morphology are stems and affixes. Stems are the core meaning-bearing units, and affixes are added before and after stems to alter the meaning and function. An affix may be a prefix (concatenated before the stem), a suffix (concatenated after the stem), or a circumfix, with parts added before and after a stem. The stem can be templatic (derived) or non-templatic (fixed). Templatic stems are stems that can be formed using templatic morphemes, whereas non-templatic word stems are not derivable from templatic morphemes and tend to be of foreign origin or names.

References

[1] D. Jurafsky and J. H. Martin, Eds., *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2020.

[2] F. Seifart, ``Orthography development,'' *Essentials of language documentation*, pp. 275--299, 2006.

[3] R. Hetzron, Ed., *The Semitic Languages*. Routledge, 1997.

[4] M. Hulden, ``Foma: a finite-state compiler and library,'' in *Proceedings of the Demonstrations Session at EACL 2009*, 2009, pp. 29--32.

[5] N. Halabi, ``Modern standard arabic phonetics for speech synthesis,'' Ph.D. dissertation, University of Southampton, 2016.

[6] T. Zerrouki and A. Balla, ``Tashkeela: Novel corpus of arabic vocalized texts, data for auto-diacritization systems,'' *Data in Brief*, vol. 11, pp. 147 -- 151, 2017.

[7] A. Dahdah and G. M. Abdulmassih, *A dictionary of Arabic grammar in charts and tables*. Librairie du Liban, 1981.

[8] M. b. A. B. Al-Razi, ``Mukhtar al-sihah,'' *Beirut: Dar al-Namudzajiyah*, 1999.

[9] A. El-Dahdah, E. Matar, and G. M. Abdul-Massih, ``majam qawaaid al-arabiiat al-aalamiiat (a dictionary of universal arabic grammar)/معجم قواعد العربية العالمية,'' 1990.

[10] A. Pasha, M. Al-Badrashiny, M. T. Diab, A. El 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 *Lrec*, vol. 14, no. 2014, 2014, pp. 1094--1101.

[11] 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*, vol. 1. Citeseer, 2001, pp. 1--8.

[12] N. Y. Habash and O. C. Rambow, ``Magead: A morphological analyzer and generator for the arabic dialects,'' 2006.

[13] K. Darwish, M. Diab, and N. Habash, ``Proceedings of the acl workshop on computational approaches to semitic languages,'' in *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, 2005.

[14] 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 the 9th International Workshop on Finite State Methods and Natural Language Processing*, 2011, pp. 125--133.

[15] K. Darwish, ``Building a shallow arabic morphological analyser in one day,'' in *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*, 2002.

[16] T. Buckwalter, ``Buckwalter arabic morphological analyzer version 1.0,'' *Linguistic Data Consortium, University of Pennsylvania*, 2002.

[17] D. Graff, M. Maamouri, B. Bouziri, S. Krouna, S. Kulick, and T. Buckwalter, ``Standard arabic morphological analyzer (sama),'' *Linguistic Data Consortium LDC2009E73*, 2010.

[18] O. Smrz, ``Elixirfm--implementation of functional arabic morphology,'' in *Proceedings of the 2007 workshop on computational approaches to Semitic languages: common issues and resources*, 2007, pp. 1--8.

[19] R. Roth, O. Rambow, N. Y. Habash, M. Diab, and C. Rudin, ``Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking,'' 2008.

[20] E. L. Antworth, ``Pc-kimmo: a two-level processor for morphological analysis,'' *Summer Institute of Linguistics*, 1990.

[21] L. Karttunen, *Finite-state lexicon compiler*. Xerox Corporation, Palo Alto Research Center, 1993.

[22] G. Kiraz, ``Multi-tape two-level morphology: a case study in semitic non-linear morphology,'' *arXiv preprint cmp-lg/9407023*, 1994.

[23] M. Mohri, F. Pereira, and M. Riley, ``Weighted finite-state transducers in speech recognition,'' *Computer Speech Language*, vol. 16, no. 1, pp. 69--88, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230801901846

[24] M. T. Diab, ``Second generation amira tools for arabic processing : Fast and robust tokenization , pos tagging , and base phrase chunking,'' 2009.

[25] A. A. Al-Nassir, ``Sibawayh the phonologist: A critical study of the phonetic and phonological theory of sibawayh as presented in his treatise? al kitab?'' Ph.D. dissertation, University of York, 1985.

[26] Y. A. El-Imam, ``Phonetization of arabic: rules and algorithms,'' *Computer Speech & Language*, vol. 18, no. 4, pp. 339--373, 2004.

[27] F. Biadsy, N. Habash, and J. Hirschberg, ``Improving the arabic pronunciation dictionary for phone and word recognition with linguistically-based pronunciation rules,'' in *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, 2009, pp. 397--405.

[28] A. Ramsay, I. Alsharhan, and H. Ahmed, ``Generation of a phonetic transcription for modern standard arabic: A knowledge-based model,'' *Computer Speech & Language*, vol. 28, no. 4, pp. 959--978, 2014.

[29] M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki, ``The penn arabic treebank: Building a large-scale annotated arabic corpus,'' in *NEMLAR conference on Arabic language resources and tools*, vol. 27. Cairo, 2004, pp. 466--467.

[30] J. Åkesson, ``Arabic morphology and phonology: Based on the marāḥ al-arwāḥ by aḥmad b.ʿaī b. mas ʿūd,'' in *Arabic Morphology and Phonology*. Brill, 2017.

[31] A. A. S. Farghaly, ``Arabic computational linguistics,'' *(No Title)*, 2010.

[32] M. Sipser, *Introduction to the Theory of Computation*. Cengage Learning, 2012.

[33] R. M. Kaplan and M. Kay, ``Regular models of phonological rule systems,'' *Computational linguistics*, vol. 20, no. 3, pp. 331--378, 1994.

[34] K. Koskenniemi, ``Two-level morphology,'' Ph.D. dissertation, Ph. D. thesis, University of Helsinki, 1983.

[35] L. Karttunen *et al.*, ``Kimmo: a general morphological processor,'' in *Texas Linguistic Forum*, vol. 22. Texas, USA, 1983, pp. 163--186.

[36] J. Bear, ``A morphological recognizer with syntactic and phonological rules,'' in *COLING*, vol. 86, no. 10.3115, 1986, pp. 991 365--991 445.

[37] ------, ``Morphology with two-level rules and negative rule

features," in *Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics*, 1988.

[38] A. W. Black, G. Ritchie, S. Pulman, and G. Russell, ``Formalisms for morphographemic description," in *Third Conference of the European Chapter of the Association for Computational Linguistics*, 1987.

[39] K. Beesley, T. Buckwalter, and S. Newton, ``Two-level finite-state analysis of arabic morphology," in *Proceedings of the Seminar on Bilingual Computing in Arabic and English*, 1989, pp. 6--7.

[40] H. Trost, ``The application of two-level morphology to non-concatenative german morphology," 1990.

[41] G. D. Ritchie, *Computational morphology: practical mechanisms for the English lexicon*. MIT press, 1992.

[42] E. L. Antworth, ``Morphological parsing with a unification-based word grammar," in *Proceedings of the North Texas Natural Language Processing Workshop*. Citeseer, 1994, pp. 24--32.

[43] H. Ruessink, *Two-level formalisms*. Katholieke Universiteit, 1989.

[44] D. Carter, ``Rapid development of morphological descriptions for full language processing systems," *arXiv preprint cmp-lg/9502006*, 1995.

[45] E. Grimley-Evans, G. A. Kiraz, and S. G. Pulman, ``Compiling a partition-based two-level formalism," *arXiv preprint cmp-lg/9605001*, 1996.

[46] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[47] C. D. Johnson, *Formal aspects of phonological description*. Walter de Gruyter GmbH & Co KG, 2019, vol. 3.

[48] K. R. Beesley and L. Karttunen, ``Finite-state morphology: Xerox tools and techniques," *CSLI, Stanford*, pp. 359--375, 2003.

[49] S. Bird and T. M. Ellison, ``One-level phonology: Autosegmental representations and rules as finite automata," *Computational Linguistics*, vol. 20, no. 1, pp. 55--90, 1994.

[50] K. R. Beesley and L. Karttunen, ``Finite-state nonconcatenative morphotactics," *arXiv preprint cs/0006044*, 2000.

[51] J. J. McCarthy, ``A prosodic theory of nonconcatenative morphology," *Linguistic inquiry*, vol. 12, no. 3, pp. 373--418, 1981.

[52] M. Kay, ``Nonconcatenative finite-state morphology," in *Third Conference of the European Chapter of the Association for Computational Linguistics*, 1987.

[53] J. A. Goldsmith, *Autosegmental and metrical phonology*. Basil Blackwell Cambridge, 1990, vol. 1.

[54] J. McCarthy and A. Prince, ``Prosodic morphology and templatic morphology," in *Perspectives on Arabic linguistics II: papers from the second annual symposium on Arabic linguistics*. John Benjamins Pub. Co. Amsterdam, 1990, pp. 1--54.

[55] J. J. McCarthy and A. Prince, ``Generalized alignment," in *Yearbook of morphology 1993*. Springer, 1993, pp. 79--153.

[56] L. Kataja and K. Koskenniemi, ``Finite-state description of semitic morphology: A case study of ancient accadian," in *Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics*, 1988.

[57] K. Beesley, ``Finite-state description of arabic morphology," in *Proceedings of the Second Cambridge Conference: Bilingual Computing in Arabic and English*, 1990, pp. 5--7.

[58] K. R. Beesley, ``Computer analysis of arabic morphology: A two-level approach with detours," in *Perspectives on Arabic Linguistics III: Papers from the Third Annual Symposium on Arabic Linguistics*. John Benjamin's Publishing Company Amsterdam, 1991, pp. 155--172.

[59] ------, ``Arabic finite-state morphological analysis and generation," in *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.

[60] R. W. Sproat, *Morphology and computation*. MIT press, 1992.

[61] S. G. Pulman and M. R. Hepple, ``A feature-based formalism for two-level phonology: a description and implementation," *Computer Speech & Language*, vol. 7, no. 4, pp. 333--358, 1993.

[62] A. Narayanan and L. Hashem, ``On abstract finite-state morphology," in *Conference of the European Chapter of the Association for Computational Linguistics*, 1993.

[63] K. R. Beesley, ``Arabic morphology using only finite-state operations," in *SEMITIC@COLING*, 1998.

[64] G. A. Kiraz, ``Multitiered nonlinear morphology using multitape finite automata: a case study on syriac and arabic," *Computational Linguistics*, vol. 26, pp. 77--105, 2000.

[65] A. Kornai, ``Formal phonology," 2018.

[66] B. Wiebe, ``Modelling autosegmental phonology with multi-tape finite state transducers," 1992.

Appendix A

A. Minimal Pairs/near-minimal Pairs (Evidence for the MSA Phonemic Inventory)

The sounds of Modern Standard Arabic (MSA) could be classified as phonemes or allophones. In order to differentiate between phonemes and allophones, a list of minimal or near-minimal pairs have been found for all the phonemes in MSA. Two things to note include: (1) The diacritics convey the vowel sounds, which is why the correct diacritics are important to correctly phonetically transcribe orthography. (2) There are MSA characters that convey the vowel sounds.

Table XXI lists the minimal / near minimal pairs for non-geminated consonants. Table XXII lists the minimal / near minimal pairs for geminated consonants that are compared to the non-geminated version of the consonant. Table XXIII list the minimal / near minimal pairs for the short and long vowels, and Table XXIV conveys that the long and short vowels are contrastive.

TABLE XXI. MINIMAL CONSONANT PAIRS

Contrastive Phones			Minimal Pair / Near-Minimal Pair		
Phone 1	**Phone 2**	**Shared Property**	**Word**	**IPA**	**Gloss**
/t/	/d/	alveolar stops	دُبْ	[dub]	bear
			تُبْ	[tub]	repent
/s/	/sˤ/	voiceless alveolar fricatives	سُوْرَةْ	[surah]	chapter of the Qur'an
			صُوْرَةْ	[sˤurah]	picture
/l/	/r/	liquids	لَانَ	[la:n]	relent, soften
			رَانَ	[ra:n]	seize; overcome; prevail
/ʕ/	/ħ/	pharyngeal fricatives	عُلِبَ	[ʕul:ib]	was boxed
			حُلِبَ	[ħul:ib]	was milked
/q/	/k/	unvoiced plosives	قَلْبْ	[qalb]	heart
			كَلْبْ	[kalb]	dog
/m/	/n/	voiced nasals	مَالَ	[mala]	swayed
			نَالَ	[nala]	gained
/ʃ/	d͡ʒ/	postalveolar	أشمل	[ʔaʃmal]	more general
			أجمل	[ʔad͡ʒmal]	more beautiful
/ð/	/ðˤ /	voiced dental fricatives	ذلك	[ðalik]	that 'male'
			ظَالم	[ðˤalim]	unjust/ oppressive
/z/	/s/	alveolar fricatives	زَاهِرْ	[za:hir]	blooming
			سَاهِرْ	[sa:hir]	up late into the night
/q/	/χ/	voiceless uvular fricatives	قَدَمْ	[qadam]	leg
			خَدَمْ	[χadam]	servants
/ʃ/	/f/	voiceless fricatives	شَعَلَ	[ʃaqala]	lit
			فَعَلَ	[faqala]	'he' did
/ʁ/	/ʕ/	voiced fricatives	غالي	[ʁa:li:]	expensive
			عالي	[ʕa:li:]	high
/b/	/m/	voiced bilabials	بَالَ	[ba:la]	urinated
			مَالَ	[ma:la]	tilted
/ʔ/	/h/	unvoiced glottals	مُؤمِنْ	[muʔmin]	faithful
			مُهمِلْ	[muhmil]	careless
/θ/	/ð/	dental fricatives	ثَمَرْ	[θamar]	fruit
			ذَهَبْ	[ðahab]	gold
/j/	/w/	voiced approximants	بَيْتْ	[bajt]	house
			مَوْتْ	[mawt]	death
/t/	/tˤ/	voiceless alveolar plosives	تَبِيتْ	[tabi:t]	she sleeps over
			طَبِيبْ	[tˤabi:b]	doctor
/d/	/dˤ/	voiced alveolar plosives	دَمَارْ	[dama:r]	destruction
			ضَمِيرْ	[dˤami:r]	conscience

TABLE XXII. GEMINATED CHARACTERS ARE LANGUAGE PHONEMES AND CONTRAST THE NONGEMINATED PHONEMES

Phoneme	Word (geminated)	IPA	Gloss	Word (un-geminated)	IPA	Gloss
[b]	حبَّ	/ħab:a/	had loved	حب	/ħab/	a seed
[t]	التَّل	/ʔat:al/	the hill	تل	/tal/	hill
[θ]	الثُّلاثاء	/ʔaθ:ula:θa:ʔ/	Tuesday	ثمر	/θamar/	fruit
[d͡ʒ]	أجَّل	/ʔad͡ʒ:ala/	delayed	أجل	/ʔad͡ʒal/	time
[ħ]	وحَّد	/waħ:da/	had united	وَحَدَ	/waħada/	united
[χ]	أخَّر	/ʔaχ:ara/	delayed	آخر	/ʔa:χar/	other
[d]	الدَّمار	/ʔad:ama:r/	the distruction	دمار	/dama:r/	distruction
[ð]	الذَّنب	/ʔað:anb/	the sin	ذنب	/ ðanb/	sin
[r]	الرُّز	/ʔar:uz/	the rice	رز	/ruz/	rice
[z]	الزَّوج	/ʔz:awd͡ʒ/	the husband	زوج	/zawd͡ʒ/	husband
[s]	السَّماء	/ʔas:ama:ʔ/	the sky	سماء	/sama:ʔ/	sky
[ʃ]	الشَّمس	/ʔaʃ:ams/	the sun	شمس	/ ʃams/	sun
[sˤ]	الصَّباح	/ʔasˤ:aba:ħ /	the morning	صباح	/ sˤaba:ħ/	morning
[dˤ]	الضُّفدع	/ʔadˤ:ufdaʕ/	the frog	ضفدع	/ dˤufdaʕ/	frog
[tˤ]	الطَّبيب	/ʔatˤ:abi:b/	the doctor	طبيب	/tˤabi:b/	doctor
[ðˤ]	الظِّل	/ʔaðˤ:il/	the shadow	ظل	/ðˤil/	shadow
[ʕ]	لعَّب	/laʕ:aba/	played with	لعَب	/laʕab/	played
[ʁ]	صغَّر	/saʁ:ara/	made smaller	صغر	/saʁar/	became smaller
[f]	أفٌّ	/ʔuf:in/	expressing impatience or contempt	لف	/luf/	wrap (command)
[q]	وقَّت	/waq:ata/	timed	وقت	/waqt/	time
[k]	أكَّل	/ʔak:ala/	fed	أكل	/ʔakala/	ate
[l]	اللَّمس	/ʔl:ams/	the touch	لمس	/lams/	touch
[m]	نمَّام	/nam:a:m/	person who spreads scandals	سمع	/samaʕ/	heard
[n]	النَّمر	/ʔan:imr/	the tiger	نمر	/nimr/	tiger
[h]	وهَّم	/wah:am/	puzzled somebody	وهم	/wahm/	assumed
[w]	توَّاب	/taw:ab/	repentant; remorseful; regretful	مواد	/mawa:d/	material
[j]	أيَّد	/ʔaj:ad/	supported	يد	/jad/	hand

TABLE XXIII. MINIMAL VOWEL PAIRS

Contrastive Phones			Minimal Pair / Near-Minimal Pair		
Phone 1	**Phone 2**	**Shared Property**	**Word**	**IPA**	**Gloss**
/i:/	/u:/	vowel - high	فُول	[fu:l]	type of Levantine bean dish
			فِيل	[fi:l]	elephant
/i/	/a/	vowel - front	أُمَّك	[ʔum:ak]	your mother - 'to male'
			أُمِّك	[ʔum:ik]	your mother - 'to female'
/i:/	/a:/	vowel - front - long	قِيل	[qi:l]	it was said
			قَال	[qa:l]	he said
/i:/	/u/	vowel - high	قِيل	[qi:l]	it was said
			قُل	[qul]	say 'command'

TABLE XXIV. MINIMAL PAIRS TO SHOW THAT LONG AND SHORT VOWELS ARE CONTRASTIVE

Contrastive Phones		Minimal Pair / Near-Minimal Pair		
Phone 1	**Phone 2**	**Word**	**IPA**	**Gloss**
/i:/	/i/	فيل	/fi:l/	elephant
		فلم	/film/	movie
/a:/	/a/	بادر	/ba:dara/	took initiative
		بدر	/badr/	full moon
/u:/	/u/	ثوم	/θu:m/	garlic
		ثمّ	/θum:a/	then

Supplement 2: Finite-state Machines for Linguistics

A Automata Hierarchy and Power

In comparison to other automata, finite-state machines (FSMs) have the least computing power beyond finite languages and can process regular expressions. More powerful complex automata and languages are push-down automata for context-free languages; embedded push-down automata for mildly context-sensitive (linear indexed) language; nested stack automata for indexed language; linear bounded automata for context-sensitive language; always halting Turing automata for recursive language; and the Turing machine for recursively enumerable language (all formal languages). Fig. 6 illustrates the hierarchy of formal languages (32).

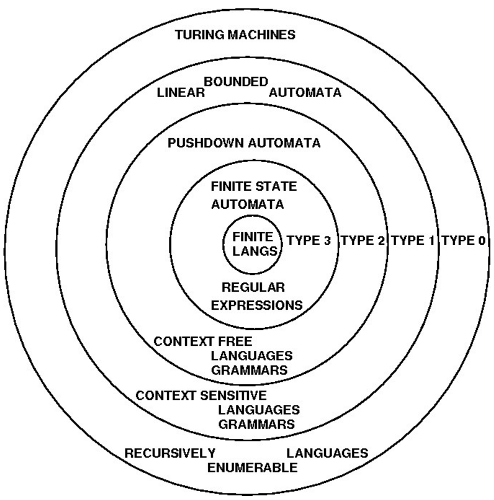

Fig. 6. Formal languages and associated automata; https://www.cs.rochester.edu.

B Development FSM and Compilation Tools

Finite-State Machines (FSMs) are either finite-state automata (FSA), which are acceptors of strings constructed to define sets of characters, or finite-state transducers (FSTs), which convert an input string into an output string using contextual or non-contextual replacement, insertion, or deletion. FSAs and FSTs are written using regular expressions and are closed under operations such as concatenation and union. FST is bidirectional and hence input and output can be inverted for the same FST.

The author in (33) constructed cascaded FSTs, with an FST mapping one character at a time between input and output. (34) developed an FST in which the rules were executed in parallel and obligatory or optional single-character mapping rules were allowed. This approach was implemented in various systems such as KIMMO and PC-KIMMO (35; 21; 20).

Bear introduced a unification-based grammar for morphotactic parsing and used diacritics coded in lexical entries to allow the rules to apply to a subset of the lexicon (36; 37). This formalism was adapted by various implementations, which allowed a rule to map between equal-sized input and output string subsequences rather than single characters (38; 39; 40; 41; 42).

Ruessink's formalism allows unequal size sequences and explicit contexts; however, this results in some invalid combinations (43). Pullman and Hepple extended Ruessink's formalism by adding rule features; however, their proposal had problems with the interpretation of obligatory rules. Carter suggested that Pullman and Hepple's formalism was impractical for specifying the mappings between unequal-length sequences (44). Grimley-Evans, Kiraz, and Pulma redefined the obligatory rules in Ruessink's formalism (45).

Algorithms exist for the compilation of rules written as regular expressions into automata (33). Computationally, parsers and compilers for regular expressions are O(n), where n is the length of the input string. Widely available FST compiler tools include Lex, Flex, xfst from Xerox, HFST, Foma, and OpenFST. Foma (4) is an open-source library for unweighted FSTs that has an interface similar to the proprietary XFST from Xerox. OpenFST is suitable for dealing with weighted transducers and has been considered a better tool than the FSM Library of AT&T. In addition, more powerful automata are available in the Natural Language Toolkit (46).

C FST for Computational Linguistics

A finite-state transducer (FST) can model most phonological rules, possibly with exceptions related to some stress and tone rules (47), which has been independently verified by (33) (21; 48). Many finite-state models are also available for phonology (49). Cascade and other extensions of finite-state technology are also available (50).

An important FST class is a two-level finite-state formalism that allows the mapping rules between input and output strings to be implemented with finite-state transducers. The same automaton can be used for analysis (decomposition) and synthesis (generation), thereby providing bidirectionality. The two-level formalism and its generalization to multi-levels are used for the phonological analyzer and the concatenative and templatic morphology analyzer.

D Multi-level Finite-state Formalism

(51) described a root-and-pattern morphology FST. (35) proposed a two-level system for language morphology. (52) proposed a framework in which each of the autosegmental tiers was assigned a tape in a multitape finite-state machine, with an additional tape for the surface form. Kay's approach followed the CV model and used four-tape automata, which was an extension of the traditional FST. (52) also proposed a framework for handling templatic morphology in which each templatic morpheme was assigned a tape in a multitape finite-state machine and an additional tape for the surface form.

The two-level formalism has been extended to multiple levels, as illustrated in Fig. 7, in which the templatic morphemes are roots, patterns, and vocalisms. The vocalism morpheme specifies the short vowels to use with a pattern; in contrast, traditional accounts of Arabic morphology collapse the vocalism into the pattern.

The advancement to templatize morphology was achieved by having multiple inputs (tapes) to the FSTs based on linguistic abstractions of Semitic nonlinear morphology. However, such constructs handle only a subset of Arabic words, such as verbs, nouns, or broken plurals.

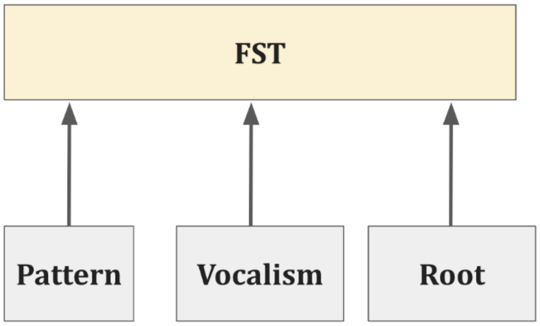

Fig. 7. Three-level FST where the pattern, root, and vocalism are the input tapes.

McCarthy's CV-based model was presented for Arabic morphology under an autosegmental phonology framework to handle verbs (51; 53). A stem is represented by three tiers: the root, vocalism, and a CV pattern. Associations are made based on well-formed conditions, association conventions, and additional rules. The Moraic model uses a different vocabulary to represent the pattern morph based on the noun prosody (54), while the Affixational model derives several templates using affixation under prosodic circumscription for verbs (55).

Infixation and reduplication are handled within the standard two-level morphology using diacritics (20). Kay's approach followed the CV model using a four-tape automaton, which was an extension of the traditional FST (52). (56) used a lexical component that takes the intersection of rules and pattern expressions and produces verbal stems, with the stems being the input for a standard two-level system. (56) also presented a system for handling Akkadian root-and-pattern morphology by adding an additional lexicon component to Koskenniemi's two-level morphology (34). Beesley's intersection approach is probably the largest system for Arabic morphology (57; 39; 58). Beesley later compiled all combinations into a transducer (59).

E Concatenative Morphological Formalism

The state-of-the-art concatenative morphological formalism consists of three components: lexical automata, morphotactic rules, and rewrite rules (60). Finite-state automata are constructed to represent prefixes, stems, and suffixes. The prefix, stem, and suffix are concatenated with markers separating them to form a lexical form based on morphotactic rules that specify valid combinations. Orthographic changes that need to be made to the lexical form to yield the surface form (word) are coded using rewrite rules incorporating contextual mappings and are implemented using an FST.

Morphotactic rules can be implemented in an FST with continuation classes using filters (34; 20; 21). Continuation classes are, however, inappropriate for handling separated dependencies, interdigitation, infixation, and reduplication, and, therefore, flag diacritics are used to address the separated dependencies, discontinuous dependencies, and long-distance dependencies within the FST framework. However, as using FSTs can be awkward, context-free grammar (with feature unification if necessary) is used to address the complex dependencies (36; 39; 40; 41; 42; 49).

F Templatic Morphological Formalism

(61) embedded pattern and vocalism morphs in the surface expression of the rules, (62) extended the two-level model by adding a third abstract level for inflection patterns, and Kiraz (1994) developed a two-level formalism based on Kay's approach that could handle CV, Moraic, and Affixational models. The first large-scale Arabic morphology implementation within finite-state method constraints, which was conducted by (39), included a 'detouring' mechanism to access multiple lexica, which was the forerunner to other studies by (63). (33) constructed cascading FSTs, in which an FST mapped one character at a time between the input and output. Subsequent advancements in this approach can be found in (63), (50), and (16). Cascade and other extensions of finite-state technology are also available (16). (64) extended Kay's approach and implemented a multi-tape system for MSA.

G Phonology Formalism

(65) modeled autosegmental phonology using FSTs, in which the autosegmental phonology was coded as linear strings, (49) used a one-level phonological approach to code autosegmental representations as a triangular prism, and (66) used multilinear coding that was processed using state labeled finite automata, which were shown to be more powerful than FSTs.

References

[1] D. Jurafsky and J. H. Martin, Eds., *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2020.

[2] F. Seifart, ``Orthography development," *Essentials of language documentation*, pp. 275--299, 2006.

[3] R. Hetzron, Ed., *The Semitic Languages*. Routledge, 1997.

[4] M. Hulden, ``Foma: a finite-state compiler and library," in *Proceedings of the Demonstrations Session at EACL 2009*, 2009, pp. 29--32.

[5] N. Halabi, ``Modern standard arabic phonetics for speech synthesis," Ph.D. dissertation, University of Southampton, 2016.

[6] T. Zerrouki and A. Balla, ``Tashkeela: Novel corpus of arabic vocalized texts, data for auto-diacritization systems," *Data in Brief*, vol. 11, pp. 147 -- 151, 2017.

[7] A. Dahdah and G. M. Abdulmassih, *A dictionary of Arabic grammar in charts and tables*. Librairie du Liban, 1981.

[8] M. b. A. B. Al-Razi, ``Mukhtar al-sihah," *Beirut: Dar al-Namudzajiyah*, 1999.

[9] A. El-Dahdah, E. Matar, and G. M. Abdul-Massih, ``majam qawaaid al-arabiiat al-aalamiiat (a dictionary of universal arabic grammar)/معجم قواعد العربية العالمية," 1990.

[10] A. Pasha, M. Al-Badrashiny, M. T. Diab, A. El 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 *Lrec*, vol. 14, no. 2014, 2014, pp. 1094--1101.

[11] 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*, vol. 1. Citeseer, 2001, pp. 1--8.
[12] N. Y. Habash and O. C. Rambow, ``Magead: A morphological analyzer and generator for the arabic dialects," 2006.
[13] K. Darwish, M. Diab, and N. Habash, ``Proceedings of the acl workshop on computational approaches to semitic languages," in *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, 2005.
[14] 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 the 9th International Workshop on Finite State Methods and Natural Language Processing*, 2011, pp. 125--133.
[15] K. Darwish, ``Building a shallow arabic morphological analyser in one day," in *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*, 2002.
[16] T. Buckwalter, ``Buckwalter arabic morphological analyzer version 1.0," *Linguistic Data Consortium, University of Pennsylvania*, 2002.
[17] D. Graff, M. Maamouri, B. Bouziri, S. Krouna, S. Kulick, and T. Buckwalter, ``Standard arabic morphological analyzer (sama)," *Linguistic Data Consortium LDC2009E73*, 2010.
[18] O. Smrz, ``Elixirfm--implementation of functional arabic morphology," in *Proceedings of the 2007 workshop on computational approaches to Semitic languages: common issues and resources*, 2007, pp. 1--8.
[19] R. Roth, O. Rambow, N. Y. Habash, M. Diab, and C. Rudin, ``Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking," 2008.
[20] E. L. Antworth, ``Pc-kimmo: a two-level processor for morphological analysis," *Summer Institute of Linguistics*, 1990.
[21] L. Karttunen, *Finite-state lexicon compiler*. Xerox Corporation, Palo Alto Research Center, 1993.
[22] G. Kiraz, ``Multi-tape two-level morphology: a case study in semitic non-linear morphology," *arXiv preprint cmp-lg/9407023*, 1994.
[23] M. Mohri, F. Pereira, and M. Riley, ``Weighted finite-state transducers in speech recognition," *Computer Speech Language*, vol. 16, no. 1, pp. 69--88, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230801901846
[24] M. T. Diab, ``Second generation amira tools for arabic processing : Fast and robust tokenization , pos tagging , and base phrase chunking," 2009.
[25] A. A. Al-Nassir, ``Sibawayh the phonologist: A critical study of the phonetic and phonological theory of sibawayh as presented in his treatise? al kitab?" Ph.D. dissertation, University of York, 1985.
[26] Y. A. El-Imam, ``Phonetization of arabic: rules and algorithms," *Computer Speech & Language*, vol. 18, no. 4, pp. 339--373, 2004.
[27] F. Biadsy, N. Habash, and J. Hirschberg, ``Improving the arabic pronunciation dictionary for phone and word recognition with linguistically-based pronunciation rules," in *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, 2009, pp. 397--405.
[28] A. Ramsay, I. Alsharhan, and H. Ahmed, ``Generation of a phonetic transcription for modern standard arabic: A knowledge-based model," *Computer Speech & Language*, vol. 28, no. 4, pp. 959--978, 2014.
[29] M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki, ``The penn arabic treebank: Building a large-scale annotated arabic corpus," in *NEMLAR conference on Arabic language resources and tools*, vol. 27. Cairo, 2004, pp. 466--467.
[30] J. Åkesson, ``Arabic morphology and phonology: Based on the marāḥ al-arwāḥ by aḥmad b.ʻaī b. mas ʻūd," in *Arabic Morphology and Phonology*. Brill, 2017.
[31] A. A. S. Farghaly, ``Arabic computational linguistics," *(No Title)*, 2010.
[32] M. Sipser, *Introduction to the Theory of Computation*. Cengage Learning, 2012.
[33] R. M. Kaplan and M. Kay, ``Regular models of phonological rule systems," *Computational linguistics*, vol. 20, no. 3, pp. 331--378, 1994.
[34] K. Koskenniemi, ``Two-level morphology," Ph.D. dissertation, Ph. D. thesis, University of Helsinki, 1983.
[35] L. Karttunen *et al.*, ``Kimmo: a general morphological processor," in *Texas Linguistic Forum*, vol. 22. Texas, USA, 1983, pp. 163--186.
[36] J. Bear, ``A morphological recognizer with syntactic and phonological rules," in *COLING*, vol. 86, no. 10.3115, 1986, pp. 991 365--991 445.
[37] ------, ``Morphology with two-level rules and negative rule features," in *Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics*, 1988.
[38] A. W. Black, G. Ritchie, S. Pulman, and G. Russell, ``Formalisms for morphographemic description," in *Third Conference of the European Chapter of the Association for Computational Linguistics*, 1987.
[39] K. Beesley, T. Buckwalter, and S. Newton, ``Two-level finite-state analysis of arabic morphology," in *Proceedings of the Seminar on Bilingual Computing in Arabic and English*, 1989, pp. 6--7.
[40] H. Trost, ``The application of two-level morphology to non-concatenative german morphology," 1990.
[41] G. D. Ritchie, *Computational morphology: practical mechanisms for the English lexicon*. MIT press, 1992.
[42] E. L. Antworth, ``Morphological parsing with a unification-based word grammar," in *Proceedings of the North Texas Natural Language Processing Workshop*. Citeseer, 1994, pp. 24--32.
[43] H. Ruessink, *Two-level formalisms*. Katholieke Universiteit, 1989.
[44] D. Carter, ``Rapid development of morphological descriptions for full language processing systems," *arXiv preprint cmp-lg/9502006*, 1995.
[45] E. Grimley-Evans, G. A. Kiraz, and S. G. Pulman, ``Compiling a partition-based two-level formalism," *arXiv preprint cmp-lg/9605001*, 1996.
[46] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.
[47] C. D. Johnson, *Formal aspects of phonological description*. Walter de Gruyter GmbH & Co KG, 2019, vol. 3.
[48] K. R. Beesley and L. Karttunen, ``Finite-state morphology:

Xerox tools and techniques," *CSLI, Stanford*, pp. 359--375, 2003.

[49] S. Bird and T. M. Ellison, ``One-level phonology: Autosegmental representations and rules as finite automata," *Computational Linguistics*, vol. 20, no. 1, pp. 55--90, 1994.

[50] K. R. Beesley and L. Karttunen, ``Finite-state nonconcatenative morphotactics," *arXiv preprint cs/0006044*, 2000.

[51] J. J. McCarthy, ``A prosodic theory of nonconcatenative morphology," *Linguistic inquiry*, vol. 12, no. 3, pp. 373--418, 1981.

[52] M. Kay, ``Nonconcatenative finite-state morphology," in *Third Conference of the European Chapter of the Association for Computational Linguistics*, 1987.

[53] J. A. Goldsmith, *Autosegmental and metrical phonology*. Basil Blackwell Cambridge, 1990, vol. 1.

[54] J. McCarthy and A. Prince, ``Prosodic morphology and templatic morphology," in *Perspectives on Arabic linguistics II: papers from the second annual symposium on Arabic linguistics*. John Benjamins Pub. Co. Amsterdam, 1990, pp. 1--54.

[55] J. J. McCarthy and A. Prince, ``Generalized alignment," in *Yearbook of morphology 1993*. Springer, 1993, pp. 79--153.

[56] L. Kataja and K. Koskenniemi, ``Finite-state description of semitic morphology: A case study of ancient accadian," in *Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics*, 1988.

[57] K. Beesley, ``Finite-state description of arabic morphology," in *Proceedings of the Second Cambridge Conference: Bilingual Computing in Arabic and English*, 1990, pp. 5--7.

[58] K. R. Beesley, ``Computer analysis of arabic morphology: A two-level approach with detours," in *Perspectives on Arabic Linguistics III: Papers from the Third Annual Symposium on Arabic Linguistics*. John Benjamin's Publishing Company Amsterdam, 1991, pp. 155--172.

[59] ------, ``Arabic finite-state morphological analysis and generation," in *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.

[60] R. W. Sproat, *Morphology and computation*. MIT press, 1992.

[61] S. G. Pulman and M. R. Hepple, ``A feature-based formalism for two-level phonology: a description and implementation," *Computer Speech & Language*, vol. 7, no. 4, pp. 333--358, 1993.

[62] A. Narayanan and L. Hashem, ``On abstract finite-state morphology," in *Conference of the European Chapter of the Association for Computational Linguistics*, 1993.

[63] K. R. Beesley, ``Arabic morphology using only finite-state operations," in *SEMITIC@COLING*, 1998.

[64] G. A. Kiraz, ``Multitiered nonlinear morphology using multitape finite automata: a case study on syriac and arabic," *Computational Linguistics*, vol. 26, pp. 77--105, 2000.

[65] A. Kornai, ``Formal phonology," 2018.

[66] B. Wiebe, ``Modelling autosegmental phonology with multi-tape finite state transducers," 1992.