Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 3834–3841
Marseille, 11–16 May 2020

Finite State Machine Pattern-Root Arabic Morphological Generator, Analyzer and Diacritizer

Maha Alkhairy*, Afshan Jafri, David A. Smith*
*College of Computer and Information Science Northeastern University,
College of Computer and Information Science King Saud University
alkhairy.m@husky.neu.edu, ajafri@ksu.edu.sa, dasmith@ccs.neu.edu

Abstract

We describe and evaluate the Finite-State Arabic Morphologizer (FSAM) – a concatenative (prefix-stem-suffix) and templatic (root-pattern) morphologizer that generates and analyzes undiacritized Modern Standard Arabic (MSA) words, and diacritizes them. Our bidirectional unified-architecture finite state machine (FSM) is based on morphotactic MSA grammatical rules. The FSM models the root-pattern structure related to semantics and syntax, making it readily scalable unlike stem-tabulations in prevailing systems. We evaluate the coverage and accuracy of our model, with coverage being percentage of words in Tashkeela (a large corpus) that can be analyzed. Accuracy is computed against a gold standard, comprising words and properties, created from the intersection of UD_PADT treebank and Tashkeela. Coverage of analysis (extraction of root and properties from word) is 82%. Accuracy results are: root computed from a word (92%), word generation from a root (100%), non-root properties of a word (97%), and diacritization (84%). FSAM's non-root results match or surpass MADAMIRA's, and root result comparisons are not made because of the concatenative nature of publicly available morphologizers.

Keywords: Morphological Analysis, Concatenative Morphology, Templatic Morphology, Root-Pattern Analysis, Morphological Generation, Diacritization, Finite State Machine

1. Introduction

Morphological analyzers support part-of-speech tagging, syntactic parsing, and semantic processing in computational linguistics and have applications in natural language processing from information retrieval to speech synthesis to machine translation.

Arabic utilizes form-based morphology, which considers the form of units making up a word, their interactions with each other and how they relate to the word's overall form, and has concatenative and templatic morphemes (Farghaly, 2010; Habash, 2010). Concatenative morphology is centered on stem and affix (prefixes, suffixes, circumfixes) morphemes, which are generally concatenated in a sequence to produce a surface form. A morphological grammar that constructs stems from interdigitation (interleaving) of the root and pattern is called templative morphology. The root provides the core meaning of a word whereas the pattern provides the properties (gender, number, etc.), category (regular/irregular verb and noun, proper noun, function word), and syntactic position. Figure 1 illustrates the structure of Arabic words using the word $fasmiEahaA$[1] where the prefix is fa (so), the suffix is haA (her), and the stem is $samiEa$ (he heard) with the root of $s\ m\ E$, which gives the core meaning of hearing, and pattern $faEila$ which gives the properties of regular verb, past, singular, masculine to the word giving the final meaning of "so he heard her".

Our model (FSAM) is an acceptor (recognizer), synthesizer (generator), and analyzer (parser) of Modern Standard Arabic (MSA) words. It is also a diacritizer that infers short vowels and other diacritics from undiacritic words. The acceptor specifies whether an input word is morphologically valid or invalid.

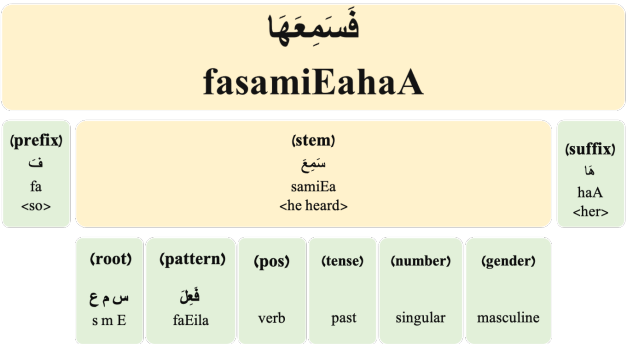

Figure 1: Arabic word morpheme breakdown. A word is a concatenation of a prefix, stem and suffix (concatenative). A stem is meaning-bearing unit which is further decomposed into its root and pattern (templatic). The root gives the core meaning and the pattern provides the part of speech (POS, category) and other linguistic properties such as number, tense, and gender. This paper uses the Buckwalter transliteration scheme (`www.qamus.org/transliteration.htm`) for examples.

The input to the generator is a root with optional specification of pattern or affix, or "print lower-words" which outputs all licit combinations of roots, patterns and affixes. A word that cannot be decomposed into a pattern and root is a fixed word (e.g Washington) and is represented by the root being the fixed word without affixes and the pattern being the identity.

The input to the analyzer is a word and the output is valid alternative morpheme decompositions (prefix, root, suffix), pattern, part-of-speech (category), and morphosyntactic features such as number and gender. The input to the diacritizer is an undiacritized word and the output is valid

[1]This paper uses the Buckwalter transliteration scheme (`www.qamus.org/transliteration.htm`) for examples

vocalizations.
The automaton utilizes morphotactic MSA grammatical rules that governs allowable concatenations of affixes and stems; and Arabic grammar licit templatic combinations of morphological patterns and roots, thereby ensuring absence of invalid words.
The finite state machine (FSM) morphologizer may be used as a generator in the downwards direction and as an analyzer in the upwards direction for both diacritic and undiacritic words. The benefit of FSM is bidirectionality and the ability to hard wire patterns; this allows us to synthesize, analyze and diacritize words with a unified architecture and without the need of a lookup table.
The designed architecture incorporates roots and a large variety of patterns, thus providing a rich set of valid generated forms and an average of around 28 analyses per undiacritic word. This contrasts favorably to table-based unidirectional universal machines of leading morphologizers that provide single analysis and do not have root-based generation capabilities.
FSAM has multiple features: (1) it analyzes words into roots; (2) it identifies sub-categories such as irregular verb and nouns from the computed pattern; (3) it generates words from roots; (4) it diacritizes, analyzes, and generates from a unified architecture; (5) the architecture is transferable to other Semitic languages such as Hebrew, Aramaic, etc; and their dialects.
We evaluate coverage against Tashkeela, a large diacritized word corpus (Zerrouki and Balla, 2017). To evaluate the accuracy of the analyser, generator, and diacritizer; we create a gold standard of words, their properties, categories, and diacritized versions. It is an intersection of the Prague Arabic Universal Dependencies treebank [2] (PADT_UD) and the Tashkeela to ensure that there are no invalid or dialectal words.
We compare FSAM's results to that of the leading Arabic morphologizer, MADAMIRA [3] . It is a concatenative morphological analyzer, that uses Penn Arabic treebank as part of its training set which has an overlap with UD_PADT. MADAMIRA (Pasha et al., 2014) combines MADA (Morphological Analysis and Disambiguation of Arabic) which is built on SAMA (Standard Arabic Morphological Analyser) and AMIRA (a morphological system for colloquial Egyptian Arabic). In contrast to MADAMIRA, FSAM's rule-based system focuses on MSA templatic morphological analysis yielding root and pattern, generation and diacritization.
The non-root evaluation results match or supersede MADAMIRA's: accuracy of root analysis from word (92% vs NA), generation of words from a root (100% vs NA), non-root properties of a word (97% vs 91%), and diacritization of a word (84% vs 59%); and coverage of analysis (82% vs 82%). There are no comparison results for roots because publicly available morphologizers are concatenative. Also our analysis of words into their irregular/regular verbs and nouns sub-categories is not compared because UD_PADT, MADAMIRA, or other systems or resources lack these finer level categorizations. Tables 1, 2, 3, 4, 5, 6, and 7 summarize the results.

[2] https://github.com/UniversalDependencies/UD_Arabic-PADT

[3] https://camel.abudhabi.nyu.edu/madamira/

2. Architecture

Our goal is to design a morphological system (morphologizer) that acts as a synthesizer, analyzer, and diacritizer solely based on Arabic language rules that avoids very large listings of stems and training on data. Towards achieving this goal, our morphologizer is built using Finite State Machines (FSMs) (Mohri, 1997; Sipser, 2005) .
FSMs are either finite state automata (FSA) that are acceptors of strings that we build to define sets of characters or finite state transducers (FSTs) that convert an input string into an output string using contextual or non-contextual replacement, insertion, or deletion. FSA and FSTs are defined 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, thereby constructing a unified synthesizer and analyzer.
An FST allows writing of generation rules without a need to write the more complex analysis rules, that are exercised automatically due to the bidirectional nature of FST. Consequently, we describe the constructs in generational language rather than decomposition language. Another advantage of finite state automata is that FST and FSA corresponds to regular expressions that closely match Arabic concatenative and templatic morphological rules. This obliviates the need for higher level formal grammars such as context free grammars, that are not bidirectional.
Algorithms exist for the compilation of rules into automata (Kaplan and Kay, 1994). Computationally, parsers and compilers for regular expressions are O(n) where n is the length of the input string. We decided to use Foma (Hulden, 2009) for this project since it is non-probabilistic and open-source. Other tools used to compile rules into automata (Bird et al., 2009) are: Lex, Flex, xfst (Beesley and Karttunen, 2003) from Xerox; HFST (Lindén et al., 2013); and OpenFST (Allauzen et al., 2007).

2.1. Morphological Automata

The morphological automata, serving as an acceptor, synthesizer and analyzer, has 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 are derived from multiple linguistic sources (Dahdah and Abdulmassih, 1981; Al-Razi, 1981; El-Dahdah et al., 1990).
FSAM's analyser/generator is a composite of three main layers of automata as shown in Figure 2: (1) templatic rule-based automaton that governs combinations of patterns and roots into a word, (2) concatenative rule-based automaton that governs combinations of prefix - stem - suffix into a word, and (3) rewrite rule transducer that applies orthographic and morpho-phonemic rules to the raw words.
The templatic automata is a union of pattern specific FSTs where each FST is a hardwired pattern. For each pattern, we have allowable combinations of prefix, suffix and root sets - implemented as FSAs. An example of a word (verb)

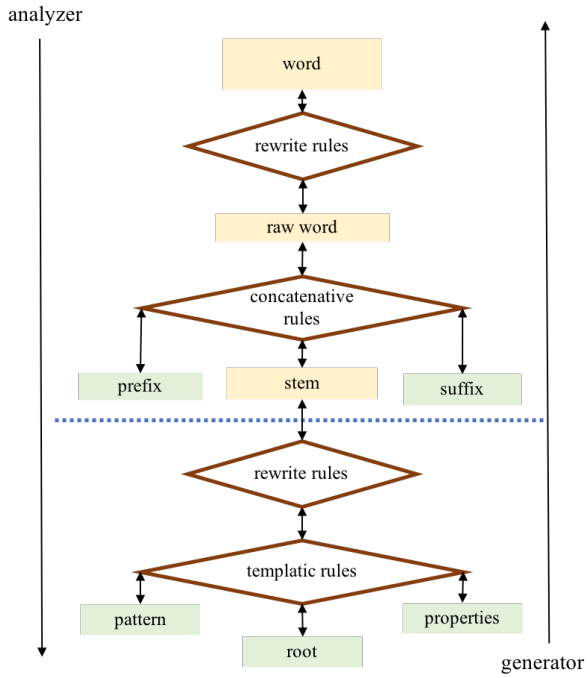

Figure 2: Architecture of the bi-directional Finite State Machine based morphological system: downwards direction is analyzer and upwards direction is synthesizer. Top portion is a rule based concatenative morphologizer and bottom is rule based templatic morphologizer that produces the root, morphological pattern, and properties which include category (part of speech), and morpho-phonemic features. In the generation (synthesis) direction all of these are optional inputs.

which uses the templatic form is demonstrated in Figure 1 where the root smE (to hear) is combined with the pattern $faEila$ (he did) to create the stem $samiEa$ (he heard), and when attached to the prefix fa (so) and the suffix haA (her) we get the word $fasamiEahaA$ (so he heard her).

The automata for fixed words such as $OamorikaA$ (America) has root as the raw word $'amorikaA$ which is the word prior to application of rewrite rules, since there is no corresponding pattern. The underlying pattern for fixed words is the identity FST where the input is the same as the output. This listing of fixed words is constructed as an FSA.

The concatenative automata is a union of FSTs that indirectly govern prefix - stem - suffix combinations through implementation of valid prefix - pattern - suffix combination rules as our morphologizer never lists stems. For each pattern FSAs are created for the sets of prefixes, suffixes, and roots that apply to it.

The rewrite transducer fixes the orthography of neighboring characters arising from concatenation of affixes with stems and interdigitization of patterns with roots. These are based on Arabic morpho-phonemic rules such as assimilation, and replacement. An example is the hamza ligature (combination of lam and hamza characters e.g lI, lO)) due to neighboring characters.

2.2. Expanding Coverage

The morphological automata described in the previous section strictly enforces allowable prefixes, suffixes and roots that may combine with a morphological pattern according to morpheme compiled. Obviously, there are morpheme combinations that do occur and should be added to the sets in the system. To allow for this expansion, a version of the morphological automata is constructed by removing restrictions on root, or prefix and suffix that combine with a pattern, while retaining hard-wired patterns because they are precisely known.

An example of a pattern is $faEala$ ('has done') that could have the restricted sets $'kl$, drs, fhm as roots; wa, fa as prefixes and haA, to as suffixes. So, if we have the word $kataba$ inputted into our system which does not have the root ktb in the sets related to the $faEala$ pattern we can analyse it with our open system and add it to our closed system thus improving our coverage and only allowing for valid words to be analysed by our closed system.

If we allow a trilateral pattern to correspond to any three letter root we have an unrestricted subsystem which would allow us to analyse valid words and expand the list of roots in the restricted system. At the same time, it admits many invalid words and hence may only be used by a language specialist to expand the list of morphemes.

2.3. Diacritizer

FSAM's diacritizer is illustrated in Figure 3. A sample input and output to this system is $wdrshA \leftrightarrow wadarasahaA$, $wadar{\sim}asahaA$.

Using the diacritized patterns of MSA as the basis of the diacritizer, we create a finite state transducer (FST) which diacritizes pattern-based words. The model used is an insertion FST which inserts diacritics to an undiacritic pattern to create its diacritic counter parts e.g $\Omega\Gamma\Lambda \leftrightarrow \Omega a\Gamma a\Lambda a$, $\Omega a\Gamma{\sim}a\Lambda a$, $\Omega a\Gamma i\Lambda a, \Omega u\Gamma i\Lambda a$, where Ω, Γ, and Λ are placeholders for the root.

For fixed words and affixes, we use a simple diacritizer which is based on a listing of diacritized fixed words and affixes instead of patterns because they don't follow any pattern.

The segmentor decomposes the word to its prefix-stem-suffix components for which the stem could be a pattern-based word or a fixed word. After we diacritize the components of the word, we concatenate them to form the diacritic word.

3. Data Sets and References

We utilize Tashkeela, PADT_UD treebank, and MADAMIRA to evaluate and compare the performance of the developed generator, analyzer, and diacritizer.

As FSAM performs the three tasks of generation, analysis, and diacritization covering both undiacritic and diacritic words, we need a corpus of diacritized Arabic to evaluate the proposed system. Tashkeela is one of the few available that satisfies our requirements as it is a collection of diacritized passages in Classical and Modern Standard Arabic. Furthermore, since our system is a deep morphologizer that works at the level of patterns and roots, the PADT_UD treebank which is built on the Prague Arabic Treebank (Hajič

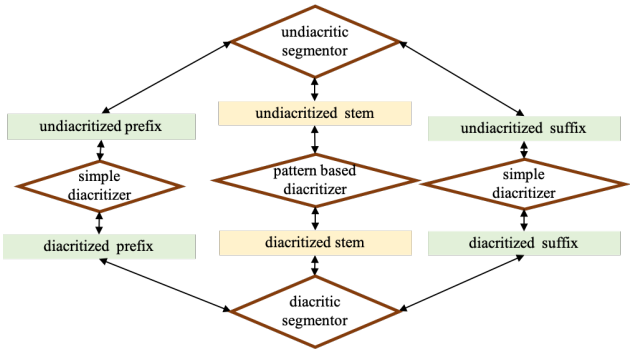

Figure 3: Architecture of the finite state machine based diacritizer system: downward direction is diacritizer. The segmentor decomposes the word into its prefix-stem-suffix. The pattern based diacritizer is an insertion finite state transducer which inserts the diacritics to the undiacritic pattern to produce the corresponding diacritic patterns e.g. $fEl \leftrightarrow faEala, fuEilo, fiEilo$. In order to diacritize the stem using the pattern diacritizer the stem is matched with the corresponding undiacritic pattern to produce the diacritic stem. The simple diacritizer inserts the diacritics directly to the undiacritic affix.

et al., 2004) is the only resource available for granular evaluation of generation and analysis, as alternatives such as the Penn Arabic treebank (Maamouri et al., 2004) lacks information about roots. PADT_UD[4] is the Universal Dependencies Prague Arabic Treebank of modern standard and colloquial Arabic containing the undiacritized word and its analysis consisting of: root, Vform (the diacritized word), gender, number, case, definite, voice, in addition to others.
We compare FSAM against MADAMIRA (in analysis only mode), a concatenative morphologizer (a morphologizer which 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 templatic morphologizer, which partially makes up for absence of patterns and roots by utilizing the SAMA stem categories to provide granular analysis.

3.1. Corpus and Treebank Vocabulary

We compute the vocabulary of undiacritized and diacritized words in Tashkeela and PADT_UD. Table 1 lists the statistics of words after removal of punctuation.

3.2. Gold Standard

We generate a gold standard from the PADT_UD treebank to serve as a reference for evaluation of analysis and generation capabilities. The gold standard must be free from punctuation, abbreviation (e.g. km), foreign words (e.g. washington), affixes (e.g Al) and single character graphemes (e.g. t); which are not considered words of the Arabic language.
To eliminate words which are colloquial rather than modern standard arabic, we intersect PADT_UD with Tashkeela, followed by serial removal of affixes, single character graphemes (letters), foreign words and abbreviations. Table 1 describes the statistics of the intersection between PADT UD and Tashkeela in addition to that of the Gold Standard which is the intersection excluding affixes, foreign words (determined by Foreign=Yes in the PADT_UD analyses) and abbreviations (determined by Abbr=Yes in the PADT_UD analyses).

References Vocabulary

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

Table 1: TOP: Vocabulary of diacritized and resulting undiacritized words in each resource ignoring punctuation. PADT_UD diacritized words are those listed as Vform (vocalized form) in its analysis of undiacritized words. BOTTOM: Gold Standard Treebank is the intersection of Tashkeela and PADT_UD followed by removal of isolated affixes, letters, foreign words, abbreviations, and entries with no analysis (oov).

3.3. Category Correspondence

There is mismatch in groupings and terminologies between our system, PADT_UD, and MADAMIRA. As our system is based on Arabic language constructs, it uses intrinsic categories, which are verb, noun, function 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 8 in the appendix details the correspondence between PADT_UD labels and the categories of our system.
A label can map onto more than one category. For instance, a NOUN in PADT_UD may be a noun, proper name or function word as it contains words such as myrAv (inheritance) “noun”, dwlAr (dollar) “proper name”, and kl (all) “function word”. Thus, a word that is analyzed as a noun in PADT_UD and analyzed as a function word in our model is marked as a function word and a match occurs.

4. Evaluation Methodology

We evaluate the developed generator, analyzer, and diacritizer on three levels: on its own, against the gold-standard treebank, and against the Tashkeela corpus. The gold-standard treebank has granular analysis with a small vocabulary and Tashkeela has a large vocabulary with no analysis; thus, each offers a different breadth and depth of evaluation.
Evaluation of the system on its own is done by checking the statistics of synthesized words in each category showing the scope of the model. Against the gold standard, we

[4] https://github.com/UniversalDependencies/UD_Arabic-PADT

check the analysis, generation, and diacritization by comparing FSAM's output to the treebank annotation. Against the Tashkeela corpus, we check the percentage analyzed. The evaluations are discussed in detail in Sections 5., 6., and 7.. Tables 2, 3, 4, 6, 5, and 7 show the statistics of the evaluations.

The gold standard evaluation favors MADAMIRA, which is based on SAMA tables having a large listing of stems and is trained on the Penn Arabic treebank which overlaps with UD_PADT. In contrast, FSAM does not have a listing of stems nor has been trained on any treebank.

5. Synthesizer Evaluation

FSAM synthesizes words in two ways: 1) an input of prefix-root-suffix to the system outputs all words resulting from the multitude of pattern and root combinations; 2) a command of "print lower-words" to the transducer synthesizes all stems that are valid combinations of patterns and roots or synthesizes all words that are valid combinations patterns, roots, prefixes and suffixes.

5.1. Stem Vocabulary

Stem vocabulary is synthesized using the command "print lower-words" to the transducer resulting in the full list of stems. Table 2 contains the statistics for the number of stems FSAM can synthesize. By construction, all these stems are valid words. We focus on stems (base words) rather than words because of the huge vocabulary that arise from additional prefix suffix combinations.

No counterpart in MADAMIRA unless their reference - SAMA (has a listing of stems) - is directly utilized. The stem vocabulary of MADAMIRA statistics is shown in Table 2. Noun, verb, function word and proper name categories are based on the label correspondence in the MADAMIRA reference table shown in Table 9 in the appendix. Note that a stem has multiple labels in the reference.

Stem Vocabulary

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

Table 2: FSAM generated stem vocabulary for each subcategory and category (Top), and MADAMIRA tabulated stem vocabulary (Bottom). *UNK means that the reference has no categorization for the stem.

When comparing the undiacritized stem vocabulary to the undiacritized words in the Tashkeela corpus, we find the overlap between FSAM's generated stems and Tashkeela words to be 88,784 stems which contrasts favorably (6 times more) against the 14,951 stems of MADAMIRA which overlaps with Tashkeela.

5.2. Gold Standard Reference

5.2.1. Synthesis Vocabulary

FSAM synthesizes word vocabulary corresponding to the gold standard by inputting the various root, prefix, and suffix combinations. These root, prefix, and suffix combinations are decompositions of the gold standard words contained in the treebank. Consequently our vocabulary is larger than that of the gold standard because of the additional patterns applicable to the prefix-root-suffix combinations. Table 3 shows the tremendous effect that the patterns have. Since there is no reference to MADAMIRA for generation of stems from roots, we check the overlap of stems with respect to the gold-standard stems (8,536 stems).

Generated Stem Overlap
wrt Gold Standard Stems

UNDIAC	FSAM	MADAMIRA
Intersection	6,622	5,146
Missing	1,914	3,390
Total	8,536	8,536

Table 3: Count of overlap between synthesized undiacritized stems and Gold Standard stems. Intersection is between Gold Standard and synthesized stems. Missing is the set Gold Standard stems - synthesized stems .

5.2.2. Generation from Root

In order to evaluate the ability of generation from a root, we use the root provided by the gold standard and prefix and suffix provided by my segmentation of the gold standard word to generate the words possible from the combinations of prefix, root, pattern, and suffix.

Table 4 shows that we have 100% accuracy and 92% coverage when generating the word from a root and its prefix and suffix. No correspondence to MADAMIRA since it is not a synthesizer.

Generated Words from Roots

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

Table 4: FSAM generated words from gold standard roots. Column 'generated' is percentage of roots that model can generate words from; e.g root k which is not considered a root in Arabic does not yield any words. Column 'correct' is the percentage of words that match the gold standard if the model generates them.

6. Analyzer Evaluation

The input to an analyzer is a word and the output of FSAM is root, pattern, category, and other linguistic information, such as number, gender, case, definiteness, and aspect of the word. The output for MADAMIRA does not include the root and pattern. We run MADAMIRA in analyses only mode.

6.1. Coverage

Using a large corpus (Tashkeela), we evaluate the coverage of our model by computing the percentage of analyzed words.

Examples of words that cannot be analysed by both systems and are invalid are: $rnblAly, Alsnbwsk, fw$ykws and words that had not been separated by whitespace and are thus considered as one word such as : EbAs-AlfwAH$, bAllyl-wAlIbAHp, AlmSlyn-wAlwjh, AlEdw-Elykm, Al$EvA'-fy, Alsdy-wxrj, AlOwAh-Al*y, Almslmyn-HAl, wgyrhA-wIny, mAlk-wAl$AfEy, AlmsAjd-IlA, fOxbrny-mHmd, yEny-AlbynAt, ESyr-wAlwjh, qwlh-wh*A, sfyAn-On, OwlA-HtbAshA; where the dash (-) indicates where the words should be separated. We refrained from using the backoff mode of MADAMIRA because it admits invalid words such as above.

Percentage Analyzed

	undiacritized
FSAM	81.83%
MADAMIRA	82.24%

Table 5: Percentage analyzed of Tashkeela. Using MADAMIRA in analyses only mode and no backoff.

6.2. Gold Standard Reference

6.2.1. Analysis

A full analyzer evaluation may only be conducted against the gold standard reference. Tashkeela, however, is only a collection of morphologically valid Arabic words, whereas the gold standard treebank has root, category, and other linguistic information. For undiacriticized evaluation, we input all words of the treebank into the analyzer and match against its analysis. We are using the Gold Standard - no OOV treebank to evaluate the systems.

Table 6 shows the comparison between MADAMIRA's and FSAM's analyses for verbs and nouns. 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 analyses around 100% of the gold standard verbs and nouns. FSAM analyses around 84% of verbs and nouns.

The advantage our system has is its ability to extract the root and pattern of a word which gives us the ability to get the shallow analyses of the word without the use of a table of stems and their properties but rather is derived based on the pattern. The properties both systems produce are the category, case, gender, mood, definiteness, number, person, voice, and aspect.

MADAMIRA categorizes the word correctly 100% of the time whereas FSAM categorizes it correctly 97% of the time. Both system have similar performances at around 99% accuracy when computing: gender (99.5% vs 99.4%), definite (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 performs better in the case of mood (99.7% vs 93%) and number (97.8% vs 90.5%). FSAM finds the root with approximately 92% correctness.

Analysis Performance
FSAM (F) vs MADAMIRA (M)

UNDIAC	verb		noun		noun+verb	
	F	M	F	M	F	M
analysed	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 6: Accuracy of analysis of undiacritized words with respect to gold standard treebank. Left is FSAM (F), Right is MADAMIRA (M). In addition, to producing roots, our model outperforms in mood, number and voice properties. MADAMIRA has almost full coverage of gold reference because of the overlap between its training data and the reference.

7. Diacritizer Evaluation

The input to a diacritizer is a fully undiacritized word and the output is a set of valid diacritizations. The evaluation is conducted with respect to the gold standard.

7.1. Gold Standard Reference

We evaluate the diacritizer by selecting all undiacritized words in the treebank then passing them to the diacritizer and checking the output against the diacritized word contained in Vform. We evaluate the output of the diacritizer according to standard Arabic spelling rules and note that the gold standard meets these standards with some exceptions that are only apparent upon visual inspection.

Diacritization

WORD	FSAM	MADAMIRA
verb	85.91	80.99
noun	83.67	53.49
function word	83.34	53.43
proper name	82.46	50.78
noun+verb	84.01	58.76
all	**83.65**	58.59

Table 7: Accuracy of diacritization with respect to the treebank. Our pattern-based model has significantly higher accuracy.

The gold standard has inconsistencies between the spelling of the diacritized words and the undiacritized version, that deflates the performance as apparent in Table 7. Examples include: using Y instead of y ($mdnY \leftrightarrow madaniy \sim N$, $fY \leftrightarrow fiy$, $HwAlY \leftrightarrow HawaAlay$, $OlgY \leftrightarrow Oulgiya$), A instead of | ($Ab \leftrightarrow |ba$, $AlAstAnp \leftrightarrow Aal|sitaAnapi$, $Axr \leftrightarrow |xaru$), and A instead of I ($AlY \leftrightarrow IilaY$, $ATlAq \leftrightarrow IiTlaAqi$, $A\$Arp \leftrightarrow Ii\$aArapK$) which can have a large effect on the evaluation.

The evaluation in Table 7 demonstrates that FSAM's diacritizer works better than that of MADAMIRA in full diacritization (84% vs 59%) because FSAM does not learn the diacritization from a corpus but rather deduces it based on the patterns that exist in the Arabic language whereas MADAMIRA trains its model on corpora which would have more partial diacritization than full.

8. Conclusion

We have developed a unified architecture MSA word generator, analyzer, and diacritizer based on Modern Standard Arabic morphotactic rules and finite-state automata; and have evaluated it with respect to a large corpus (Tashkeela) and gold standard treebank (the intersection between Tashkeela and UD_PADT) and compare the results against MADAMIRA.

We evaluate the model on coverage (percentage analysed) based on Tashkeela and accuracy of results based on gold standard and compare to MADAMIRA. The coverage of Tashkeela is around 82% for both models. FSAM performs on par or better than MADAMIRA when finding the various properties of the word (category, gender, definiteness, person, case, aspect, voice, mood, and number), in addition to having the ability to compute root and patterns of a word.

The advantages of our model are as follows: It (1) has a unified architecture for analyses, generation, and diacritization; (2) has 135 times larger synthesis vocabulary than that of MADAMIRA's reference vocabulary; (3) generates words from roots with 100% accuracy; (4) analyses words to roots with 92% accuracy; (5) the architecture is scalable; (6) the architecture is transferable to other Semitic languages such as Hebrew, Aramaic, etc; and their dialects.

The proposed FSM model may be extended to a pronunciation system that utilizes the noun-verb categorization of a word by the morphologizer. In addition, it can be extended to incorporate morphosyntactic rules that is integrated with a syntactic parser. In another direction, the root and pattern analysis of the morphologizer can be utilized in semantic analyses.

9. Acknowledgements

We would like to thank the reviewers for all their comments and acknowledge the Northeastern University tier 1 grant for Building a Digital Archive of Cherokee.

Appendix: Correspondence Tables

Tables 8 and 9 shows the correspondence between PADT UD label and categories of our developed system.

LABEL	CATEGORY	DIAC	UNDIAC
NOUN	noun proper name function word	14,405	8,424
VERB	verb	4,603	3,551
CCONJ	function word proper name	83	49
PRON	function word	12	34
PART	function word	19	21
DET	function word	34	37
ADJ	noun proper name	5199	3587
ADP	function word proper name noun	94	105
AUX	verb function word	99	90
ADV	function word	22	25
PROPN	proper name	29	28
INTJ	noun function word	3	3
X	proper name function word noun	2693	2679

Table 8: PADT UD labels' correspondence to our categories: noun, verb, function word, proper name. The statistics of diacritized (diac) and undiacritized (undiac) for each label

10. References

Al-Razi, M. b. A. B. (1981). Mukhtar al-sihah. *Beirut: Dar al-Fikr*.

Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., and Mohri, M. (2007). OpenFst: A general and efficient weighted finite-state transducer library. In *International Conference on Implementation and Application of Automata*, pages 11–23. Springer.

Beesley, K. R. and Karttunen, L. (2003). Finite-state morphology: Xerox tools and techniques. *CSLI, Stanford*.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python: Analyzing text with the Natural Language Toolkit*. O'Reilly Media, Inc.

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

El-Dahdah, A., Matar, E., and Abdul-Massih, G. M. (1990). Majam qawaaid al-arabiiat al-aalamiiat (A Dictionary of Universal Arabic Grammar).

Farghaly, A. A. S. (2010). *Arabic computational linguistics*. CSLI Publications, Center for the Study of Language and Information.

Habash, N. Y. (2010). Introduction to Arabic natural language processing. *Synthesis Lectures on Human Language Technologies*, 3(1):1–187.

Hajič, J., Smrz, O., Zemánek, P., Šnaidauf, J., and Beška, E. (2004). Prague Arabic dependency treebank: Development in data and tools. In *Proc. of the NEMLAR Intern.*

LABEL	CATEGORY
abbrev	proper name
adj	noun
adj_comp	noun
adj_num	noun
adv	noun
adv_interrog	noun
adv_rel	noun
conj	noun
conj_sub	noun
interj	noun
noun	noun
noun_num	noun
noun_prop	proper name
noun_quant	noun
part	function word
part_det	function word
part_focus	function word
part_fut	function word
part_interrog	function word
part_neg	function word
part_restrict	function word
part_verb	function word
part_voc	function word
prep	function word
pron	function word
pron_dem	function word
pron_interrog	function word
pron_rel	function word
verb	verb
verb_pseudo	verb

Table 9: MADAMIRA labels' correspondence to our categories: noun, verb, function word, proper name

Conf. on Arabic Language Resources and Tools, pages 110–117.

Hulden, M. (2009). Foma: A finite-state compiler and library. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*, pages 29–32. Association for Computational Linguistics.

Kaplan, R. M. and Kay, M. (1994). Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.

Lindén, K., Axelson, E., Drobac, S., Hardwick, S., Kuokkala, J., Niemi, J., Pirinen, T. A., and Silfverberg, M. (2013). HFST: A system for creating NLP tools. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 53–71. Springer.

Maamouri, M., Bies, A., Buckwalter, T., and Mekki, W. (2004). The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *NEMLAR conference on Arabic language resources and tools*, volume 27, pages 466–467. Cairo.

Mohri, M. (1997). Finite-state transducers in language and speech processing. *Computational linguistics*, 23(2):269–311.

Pasha, A., Al-Badrashiny, M., Diab, M. T., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., and Roth, R. (2014). Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *LREC*, volume 14, pages 1094–1101.

Sipser, M. (2005). Introduction to the theory of computation (2nd eds.).

Zerrouki, T. and Balla, A. (2017). Tashkeela: Novel corpus of Arabic vocalized texts, data for auto-diacritization systems. *Data in brief*, 11:147.