Abdulrahman Alosaimy, Eric Atwell

Tagging Classical Arabic Text using Available Morphological Analysers and Part of Speech Taggers

Abstract

Focusing on Classical Arabic, this paper in its first part evaluates morphological analysers and POS taggers that are available freely for research purposes, are designed for Modern Standard Arabic (MSA) or Classical Arabic (CA), are able to analyse all forms of words, and have academic credibility. We list and compare supported features of each tool, and how they differ in the format of the output, segmentation, Part-of-Speech (POS) tags and morphological features. We demonstrate a sample output of each analyser against one CA fully-vowelized sentence. This evaluation serves as a guide in choosing the best tool that suits research needs. In the second part, we report the accuracy and coverage of tagging a set of classical Arabic vocabulary extracted from classical texts. The results show a drop in the accuracy and coverage and suggest an ensemble method might increase accuracy and coverage for classical Arabic.

1. Introduction

Arabic morphological analysis is essential to Arabic Natural Language Processing (NLP) tasks, and part-of-speech (POS) tagging is usually done in the first steps of advanced NLP tasks such as machine translation and text categorization. It derives its importance as its accuracy impacts other subsequent tasks. Arabic morphology is one of the most studied topics in Arabic NLP. POS tagging can be defined as the procedure of identifying the morphosyntactic class for each lexical unit using its structure and contextual information. Due to the nature of the language, being highly inflectional, and the lack of short vowels, morphological analysis of Arabic is not an easy task. The analysis involves handling of a high degree of ambiguity.

POS tagging usually uses the information provided from the morphological analyser. A morphological analyser (MA) is a context-independent tagger that provides all possible solutions based on a lexicon or dictionary. While POS taggers and MAs tag the word morphosyntactically, some POS taggers uses the context to either choose one tag or provide an ordered list of tags.

A survey of the literature shows that multiple morphological analysers and POS taggers exist. The accuracy and features of those taggers vary and errors are generated for every tagger. No tagger shows a perfect performance and no tagger has been adopted as a standard. Therefore, choosing between available taggers can be challenging.

Classical Arabic is the "liturgical" language that Muslims around the world use in religious practice. CA is also known as "Fussha" (the clearest), which Arabic Grammarians build their rules upon. One variant of CA is the Quranic Arabic, which is worded from CA, but differs in the sense that it is believed by Muslims to be the direct word of Allah. As time passes, different spoken variants of Classical Arabic emerged and people needed a standard form of communication: the Modern Standard Arabic (MSA). MSA is recognised as the formal and standard written Arabic. MSA is the language currently employed in media and education Bin-Muqbil (2006).

Even though the morphology of MSA is inherited from CA, two studies showed that CA is not compatible with MSA taggers and vice versa. S. Rabiee (2011) tried to adapt several taggers by training them on a classical Arabic Corpus: the Quranic Arabic Corpus (QAC), and then tested them on MSA. The accuracy achieved in tagging a 66-word MSA sample was "not impressive", 73% was achieved. Alrabiah et al. (2014) compared MADA Habash et al. and AlKhalil Boudchiche et al. (2016) both designed for MSA in order to annotate the KSUCCA corpus. Using five samples from different genres of CA, an evaluation of these two systems showed a drop in their accuracy by 10-15%. This shows that current taggers need to be adapted for CA and their dictionaries need to include more classical vocabulary. We extend this evaluation to examine the coverage and accuracy of the surveyed tools.

Next section reviews relevant work. The third and fourth sections list evaluated POS taggers and MAs in detail. The fifth section compares those tools by their features and demonstrates such differences on one tagged sentence. The last section reports the accuracy and coverage on a collection of classical vocabulary.

2. Related work

Several previous studies surveyed the linguistic resources available for researchers in the field of Arabic NLP. Atwell et al. (2004) conducted a survey on the available MAs and came up with 10 different analysers. Authors concluded their survey pointing out that most of those analysers are not freely available or they are hard to use. Maegaard (2004) surveyed the state-of-art language resources including MAs and POS taggers. Basic Language Resource Kit (BLARK) project (2010) listed 7 MAs, three of which are commercial software. Sawalha (2011) listed 6 MAs with his proposal of a new fine-grained morphological analyser, three of which are freely available. Albared et al. (2009) surveyed the "POS tagging" techniques with a focus on Arabic: MSA and dialects. None was designed for classical Arabic. Those techniques were criticized as assuming closed-vocabulary which might not be the case with classical Arabic. Al-Sughaiyer and Al-Kharashi (2004) conducted a survey of Arabic "morphological analysis" techniques and classified the efforts in analysing Arabic morphology into four categories: table-lookup, linguistic (using finite state automaton or traditional grammar), combinatorial and pattern-based.

Focusing on *available* MAs and POS taggers, we performed a comprehensive search, that adds to previous surveys, an in-depth literature review of available MAs and POS taggers. We limited the search to MAs and POS taggers that:

are designed for MSA or CA, i.e. either designed for Arabic but not intended for dialectal Arabic or has a model for MSA or CA.

are able to analyse all forms of words, i.e. not designed for verb only for example.

are available freely for research purposes, and

have academic credibility, with at least one published academic paper

The result of this survey are seven MAs and eight POS taggers listed in table 1.

POS tagger	Sub-category	Paper
Mada (MD)	knowledge-based: SAMA OR ALMOR. SVM using SVMTools for disambigation	Habash et al.
AMIRA (AM)	data-driven: Support Vector Machines (SVM) using YAMCHA toolkit	Diab
MadaAmira (MA)	knowledge-based: using a lexicon: SAMA OR AL. SVM for disambigation	Pasha et al. (2014)
Stanford (ST)	data-driven: Cyclic dependency network	Toutanova et al. (2003)
ATKS' POS Tagger (MT)	N/A	Kim et al. (2015)
Marmot (MR)	data-driven: CRF	Mueller et al. (2013)
SAPA (WP)	data-driven: CRF	Gahbiche-Braham et al. (2012)
Farasa (FA)	joint segmentation/POS tagging/ Parsing	Zhang et al. (2015)
Morphological Analyser	Sub-category	Paper
AraComLex (AR)	Finite state transducer	Attia et al. (2014)
ElixirFM (EX)	Haskell, functional programming	Smrz (2007)
BAMA,AraMorph (BP)	Dictionary	Buckwalter (2002)
Almorgeana (AL)	Dictionary	Habash et al.
ATKS' Sarf (MS)	N/A	N/A
AlKhalil (KH)	Dictionary	Boudchiche et al. (2016)
Qutuf (QT)	Dictionary	Altabbaa et al. (2010)
Excluded Tools	Sub-category	Reason
MORPH2	MA: knowledge-based: XML lexicon	Kammoun et al. (2010)(2)
Khoja ArabicTagger	POS-tagger: Hybrid: Statistical and Rule-based. Vetrabi for disambiutation	Khoja (2001)(2)
SAMA	MA: Dictionary	Maamouri et al. (2010)(1)
SALMA	MA: N/A	Sawalha et al. (2013)(2)
Xerox	MA: FST	Beesley (1998)(3)

Table 1: The list of MAs and POS Taggers that have been studied. Reasons of exclusion: (1) Only available to LDC members. (2) Authors did not response to our request of their system. (3) The demo website is working but its web service produces 501 error.

3. Available Morphological Analysers

3.1. AraMorph (BP)

AraMorph (a.k.a BAMA) is free GNU-licenced software originally written in Perl by Tim Buckwalter in 2002 and published in `www.qamus.org`. The software was later optimized by Jon Dehdari on 2005 to support UTF-8 encoding and speed up the processing time. AraMorph has been ported to Java by Pierrick Brihaye and published on `http://www.nongnu.org/`. In addition, AraMorph has received more work in 2012 by Hulden and Samih (2012)[1] that converts original table-based procedural AraMorph software into a finite-state transducer (FST) parser using Foma(Hulden, 2009)[2]. The authors claim that it is faster and more flexible, i.e. a wider range of applications can use the FST such as spell checkers. Tim Buckwalter released BAMA 2 and later SAMA 3, but they need Linguistic Data Consortium (LDC) licence to be used; therefore, they have been excluded from our list. AraMorph uses a list of prefixes, suffixes, and a compatible table. By extracting all possible compatible substrings that match these affixes, it returns all matched candidates. However, infixes are common in Arabic, and thus it fails to identify them correctly (e.g. identify the plurality of a "broken" plural noun).

TAGSET: About 70 basic subtags (Habash, 2010). They are mixed with morphological features to form more complex tag such as: `IV_PASS` (imperfective passive verb).

3.2. AlKhalil (KH)

AlKhalil (Boudchiche et al., 2016) is a morphosyntactic analyser of MSA shipped with a large set of lexicon and rules. It is an open-source free software written in Java and in Perl. The latest version 2 was released on 2016 [3] which improved the lexicon and added lemma and its pattern to the list of features. The standard way to interact with AlKhalil is using its graphical user interface that accepts raw text in UTF8 encoding. El-haj and Koulali (2013) reported that AlKhalil (v1.1) reached an accuracy of 96%.

OUTPUT: The system results can be either shown in browser or saved as a comma-separated file. For a given word, AlKhalil returns a list of solutions of possible tag of the stem with features. Noun features are its nature, root and pattern in addition to functional features of noun: gender and number. Verb features are aspect, form and voice in addition to syntactic features: form, root, permittivity[4], transitivity and conjugation's gender, person and number. For every solution, the system determines its voweled form, and its prefix and/or suffix whenever those exist.

[1] `https://code.google.com/p/buckwalter-fst/`

[2] Foma is a software for constructing finite-state automata and transducers for multiple purposes. `https://code.google.com/p/foma/`.

[3] `http://oujda-nlp-team.net/?p=1299&lang=en`

[4] Verbs are traditionally classified into two categories: "primitive" which all of its characters are primitive and "derived" where one or more characters have been added to the original primitive verb

TAGSET: AlKhalil is not consistent in identifying the possible tags of the word and its results are not in readily reusable form: Morphological and grammatical features are embedded within a plain text that describes the analysis. To the best of our knowledge, AlKhalil does not have a predefined set of tags. For example, for some functional words that have different possible analyses it returns one analysis with a description like: "conditional or negative particle", instead of returning two analyses: "conditional particle" and "negative particle". We estimate the possible tags for the base form of the word to be at least 118 tags.

3.3. AraComLex (AR)

AraComLex (Attia et al., 2014) is a morphological analyser and generator that uses finite state technology shipped with a contemporary dataset of news articles. It uses rule-based approach with stem as the base form in its lexicon. The last version published is 2.1[5]. The analyser uses Foma(Hulden, 2009) to construct a model and then lookup for matches.

A distinguishing feature in AraComLex is the identification of multi-word expressions. However, since AraComLex assumes a tokenized input provided by author's tokenizer which was not working[6], we could not find a suitable tokenizer that make it able to detect and identify multi-word expressions.

INPUT: With the lack of technical documentation and after some trial-and-error: AraComLex expects non-diacritized UTF8-encoded text with each word in a line. The system fails to find proper analysis if diacritics are present.

OUTPUT: The output of AraComLex is a set of solutions for every given word in a custom format as can be seen in Section B in the appendices. No description of the tagset is provided: "fut" tag for example

3.4. ALMORGEANA (AL)

ALMORGEANA (Habash, 2007) is a lexeme-based morphological analyser and generator. It uses Buckwalter's lexicon with a different engine that can additionally generate the proper inflected word given a feature-set. In the analysis task, it differ from AraMorph in the output lexeme-and-feature representation. In addition, it has a back-off step where it looks for compatible substrings of prefix and suffix and if found, the stem is considered a degenerate lexeme.

ALMORGEANA is used in MADA and presumably MADAMIRA suits to generate all possible morphological analysis of a given text. This step follows the preprocessing step of normalization. ALMORGEANA can be used with either Buckwalter Arabic

[5] `sourceforge.net/projects/aracomlex/`

[6] The author also published a set of relevant tools in his web page `http://www.attiaspace.com/getrec.asp?rec=htmFiles/fsttools` including a guesser and a tokenizer in a compiled format for Mac and Windows. However, they did not work on current operating systems (at least on MAC OSX 10.10). One tool is Arabic Morphological Guesser, with back-off feature, that is, if a word is not found in the lexicon, it guesses a correct morphology rather than returning none.

Morphological Analyser (BAMA) or Standard Arabic Morphological Analyser (SAMA). The latter is only available to LDC members, so we used BAMA instead. MADA authors reported that using BAMA instead of SAMA will result in a slight drop (2-4%) in word disambiguation.

3.5. Elixir FM (EX)

Elixir Functional Morphology (Smrz, 2007) is an analyser and generator that reuse and extends the functional morphology library for Haskell. Elixir has two interfaces to the core Haskell system written in Perl and Python. Its lexicon is designed to be abstracted from the actual program which allows easy addition to the lexicon. It was initially derived from Buckwalter dictionary but it has been enriched with syntactic annotations from Prague Arabic Dependency Treebank (PADT).

TAGSET: Elixir uses the same tagset of PADT (23 basic tags). The tags consist of a 10-position string with first two characters reserved for POS tag and the remaining eight includes morphological and grammatical features like gender, person, case and mood.

3.6. Sarf from Arabic Toolkit Service (MS)

Microsoft Research Lab in Cairo has developed a set of linguistic tools targeting Arabic language. Among eight tools, they provide free of charge access to a morphological analyser (SARF) and a POS tagger for academic researchers, professors and students only. We could not find an academic paper the describes how the two tools work. The toolkit can be accessed using SOAP web service.

The morphological analyser (SARF) provides all possible analyses of a given word: affixes, stem, diacritized form and morphological features like gender. One distinguishing feature of SARF is that it rank its solutions based on the actual language usage of each analysis.

TAGSET: contains 109 possible complex tags, making it the second largest tagset. The tagset has some combination of morphological features in it. For example, it has three type of pronouns: first-person (with suffix *_MOTAKALLEM*) pronouns, second-person and third-person. The tagset has about 70 basic tags.

3.7. Qutuf (QT)

Altabbaa et al. (2010) proposed an NLP framework written in Python that has a morphological analysis component. The latest version of Qutuf is 1.01; but it is currently in an idle state. Qutuf used Alkhalil dictionary after enriching it. Qutuf extends Alkhalil by making the output easy to be reusable and by assigning each solution with a probability.

TAGSET: A tag has 10 slot separated by comma that represents the base POS tag and some morphological and syntactical features. Some slots serve different meanings depeding on the main POS tag. For example, slot 2 represents the punctation mark (if the main POS is "other"), particle (if "particle") type or gender (if "verb" or "noun").

4. Available POS Taggers

POS taggers assign one POS tag to every word-form or to every word's segments. Unlike MAs, POS taggers assign a tag that is contextually suitable. Some POS taggers returns only one tag, a ranked list of possible POS tags or a list with each tag assigned with a probability. Some POS taggers use MAs as a preprocessing step (e.g. MADA, MADAMIRA, MarMot .. etc) and thus they disambiguate and rank different proposed analyses. Some POS taggers use MAs even in the tokenization process, e.g. MADA and MADAMIRA.

While there are some POS taggers that do word-based tagging (e.g. Mohamed et al. (2010)), all POS-tagger in our list do morpheme-based tagging. Because of Arabic's rich morphology, word sparsity is high and consequently word segmentation becomes important. Studies have shown that word segmentation lowers data sparseness and achieves better performance (Diab et al., 2004; Benajiba and Zitouni, 2010). POS tagger usually has a component that does the segmentation or relies on the user to provide a segmented input. However, this segmentation increases the ambiguity as a word may be segmented into multiple candidate sets of segments.

4.1. MADA+TOKAN suite (MD)

MADA (Habash et al.) is a popular suite that has multiple tools for Arabic NLP. MADA processes raw Arabic text to provide a list of applications: POS tagging, diacritization, lemmatization, stemming and glossing. MADA is written in Perl and uses Support Vector Machines (SVM) model trained on Penn Arabic Treebank (PATB) to select a proper analysis from the list provided by Buckwalter Arabic Morphological Analyser (BAMA). MADA uses 19 features, 14 of which are morphological features, to rank the list of possible analysis. The reported accuracy of predicting the correct POS tag is 96.1(Pasha et al., 2014).

TAGSET: MADA "targets the finest possible POS tagset" (Habash et al.). It supports the mapping to four different possible tagsets: ALMORGEANA, CATiV, PATB, or Buckwalter. However, we used the tagset used internally which has a size of 36 tags for tagging the base of the word. In addition, five, eighteen, seven, and two tags are dedicated for article, preposition, conjunction and questions *proclitics* respectively; and twenty-two tags for *enclitics*. The tagset used by MADA is well documented in the manual shipped with the suite.

4.2. AMIRA Toolkit (AM)

AMIRA (Diab) is a toolkit of three main tools: tokenizer, POS tagger, and base phrase chunker. The POS tagger uses YamChi toolkit, a SVM-based sequence classification toolkit. The toolkit does not depend on deep morphology information, instead it learns from the surface data. AMIRA was trained on PATB. The reported accuracy of predicting the correct POS tag using default tagset is 96 (Diab).

TAGSET: AMIRA can output the tags in one of three tagsets: RTS, Extended RTS, Extended RTS with person information. Extended RTS with person information has about 72 tags and those tags encodes gender, number and definiteness. After removing features from the tag, we had about 25 basic tags.

4.3. MADAMIRA suite (MA)

MADAMIRA (Pasha et al., 2014) is a suite that combines two previously mentioned systems: MADA and AMIRA. MADAMIRA ported the two systems into JAVA programming language allowing it to be portable, extensible and even faster. MADAMIRA supports MSA and Egyptian Arabic. One added feature to MADAMIRA is the server mode feature, which allows the user to run MADAMIRA in the background and then send http requests for tokenization, tagging, ... etc. While the accuracy has not improved, the speed of tagging has improved over MADA substntially (16-21x faster). The reported accuracy of predicting the correct POS tag is 95.9%(Pasha et al., 2014).

TAGSET: The tagset used by MADAMIRA extends MADA tagset by having some tags for Egyptian Arabic processing.

4.4. Stanford POS tagger and segmenter (ST)

Stanford NLP group released a list of Arabic NLP tools including a POS tagger (Toutanova et al., 2003) and Arabic word segmenter (Diab et al., 2013). The POS tagger is shipped with a model for Arabic trained on the Penn Arabic Treebank (PATB). It uses Maximum Entropy approach to assign a POS tag to a segmented text (using Stanford Arabic Word Segmenter). Stanford Arabic Word Segmenter uses Conditional Random Fields (CRF) classifier to normalize the text and split off clitics from base words in a similar segmentation schema to one used in the PATB. El-haj and Koulali (2013) reported that Stanford Tagger reached an accuracy of 96.5%.

TAGSET (augmented) Bies tags of 25 basic tags. Authors augmented the tagset by adding DT (determiner) to the beginning of nominal tags.

4.5. MarMoT (MR)

MarMoT (Mueller et al., 2013) is a generic CRF morphological tagger written in Java. MarMoT provides a pre-trained model that was trained on the PATB provided by SPMRL2013 shared task. MarMoT does backward-forward computations by incrementally increased order to prune the size of possible morphological analyses. MarMoT is efficient in training high order CRF classifiers even with large tagset and does some approximation using coarse-to-fine decoding. MarMoT assumes a transliterated and tokenized input according to the PATB transliteration and tokenization. We used TOKAN segmentation tool to pre-process the input. The reported accuracy of predicting the correct POS tag is 96.43%.

TAGSET The same 25-tag RTS tagset used in PATB. Additionally MarMoT provides morphological features identical from AraMorph.

4.6. Arabic Toolkit Service POS Tagger (MT)

Arabic Toolkit Service (ATKS) Kim et al. (2015) also have a tagger that identifies the part-of-speech of each word in a text. It is not clear whether it uses the morphological analyser in the process of tagging. This tool identifies the grammatical features like mood and case; in addition, it resolves the nunation, the addition of nun sound that indicates noun's indefinite case. Instead of normalizing, the tagger uses spelling corrector as a preprocessing step. This helps in decreasing the ambiguity caused by normalizing Hamza and Alif letters.

TAGSET: Has a detailed tagset: (>3000 tags [7]). However, this tagset is not published as MS's tags; it is estimated to have

4.7. Segmentor and Part-of-speech tagger for Arabic (WP)

Segmentor and Part-of-speech tagger for Arabic (Gahbiche-Braham et al., 2012) is a tool that uses a CRF model trained on PATB using Wapiti toolkit[8]. The tool has two components: one to predict POS tag and and the second is to split the enclitics. The reported accuracy of predicting the correct POS tag is 96.38%.

TAGSET: WP used the list of main 24 POS tags of PATB, with 3, 6, and 2 for conjunction, prepostion, and determiner prefixes respectively.

4.8. Farasa (FA)

Farasa (Zhang et al., 2015) is a toolkit for segmentation/tokenization module, POS tagger, Arabic text Diacritizer, and Dependency Parser. Farasa is different from other POS taggers as it can jointly segment, pos-tag, and parse the text which avoids error propagation in the pipelined structure and should exploit syntactic information for POS tagging. This is particularly useful for tagging CA as CA is different in vocabulary from MSA but it shares similar syntax. The reported accuracy of predicting the correct POS tag of MSA is 97.43% and of CA is 84.44%.

TAGSET: FARASA has a tagset of 16 basic tags.

5. Discussion

While POS taggers and morphological analysers predict the main POS tag, they vary in fine-grainness of tagset and segmentation. In agreement with points made by Jaafar and Bouzoubaa (2014); Alosaimy and Atwell (2015), taggers differ in many aspects:

[7] https://www.microsoft.com/en-us/research/project/part-of-speech-pos-tagger/

[8] `https://wapiti.limsi.fr/`

tagset used, output format, method used, and tokenization. Most taggers adapt their own tagset, and they subsequently assume its tokenization scheme. Table 2 and 3 lists supported features by each morphological analysers and POS tagger. Most taggers produce their results in their customized format as shown in section B in the appendix.

Name	AR	EX	BP	AL	MS	KH	XE	QT
Base POS tag	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Aspect	Yes*	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Person	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Gender	Yes	Yes	Yes	Yes	Yes	Yes[a]	Yes	Yes
Number	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Transitivity	Yes	-	-	-	-	Yes	0	Yes
Voice	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
State	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Mood	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Case	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Pattern	-	Yes	-	-	Yes	Yes	Yes	-
Root	Yes	Yes	-	-	Yes	Yes	Yes	-
Stem	-	Yes	Yes	Yes	Yes	Yes	-	-
Lemma	-	-	Yes	Yes	-	Yes	-	-
Diacritization	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Glossing	-	Yes	Yes	Yes	-	-	Yes	-
Tokenization	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Segment-based°	-	Yes	-	-	-	-	Yes	Yes

Table 2: For each given word/segment, the result of each morphological analysers. Exceptions: * Tense (past, present, and future) is used instead of the aspect of the verb but they are highly related. ° whether morphosyntactic features are for each morpheme or not. [a] only for nominals

To show the differences in context, Appendix A presents one Hadith (an utterance attributed to prophet Mohammed often called "prophet sayings") sentence annotated by each tagger. The sentence was extracted from the prophet Mohammed sayings (classical Arabic): لَا يُؤمِنُ أَحَدُكُمْ حَتَّى يَكُونَ هَوَاهُ تَبَعًا لِمَا جِئْتُ بِهِ *lā yuʾminu ʾaḥadukum ḥattaā*

Name	MD	AM	MA	ST	MS	MR	WP	FA
Base POS tag	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Glossary	Yes	-	Yes	-	-	-	-	-
Aspect	Yes	Yes	Yes	Yes*	Yes	-	-	-
Person	Yes	Yes	Yes	-	Yes	-	-	-
Gender	Yes	Yes	Yes	-	Yes	-	-	Yes^a
Number	Yes	Yes	Yes	Yes^o	Yes	-	-	Yes^a
Transitivity	-	-	-	-	-	-	-	-
Voice	Yes	Yes	Yes	Yes	Yes	-	-	-
State	Yes	-	Yes	-	Yes	-	-	-
Mood	Yes	-	Yes	-	Yes	-	-	-
Case	Yes	-	Yes	-	Yes	-	-	-
Pattern	-	-	-	-	-	-	-	-
Root	-	-	-	-	-	-	-	-
Stem	Yes	-	Yes	-	-	-	-	-
Lemma	Yes	-	Yes	-	-	-	-	-

Table 3: For each given word, the result of POS Taggers. Exceptions: * Yes unless it is passive: verb mood can not be determined. ^{o} Number is either singular or plural. ^{a} only for nominals.

yakuwna hawaāhu tabaʿan limaā ǧiʾtu bihi (None of you [truly] believes until his desires are subservient to that which I have brought). The sentence is fully vowelized, including the ending vowel. However, some taggers (ST, MR, AR, BP, KH) performed better when vowels are completely removed, as they were trained on unvowelized texts or the ending vowel is not expected.

We used a revised CoNLL-U format to represent the tagged sentence using MAs and POS taggers. We added one column (the 1st) to represent the tagger name and dropped CoNLL-U's 3,7,8,9 columns as irrelevant. Since MAs do not disambiguate, we manually picked the most-correct analysis. Last column shows the selected analysis and the number of alternative analyses.

This conversion is not straightforward. We had to deal with a number of different output-formats. In addition, the morphological features values were unified for straight comparison. We had to deal with different transliterations and representations: e.g. we extracted clitics from word-based taggers, we extracted morphological features from compound-tag (e.g. word #5 and `IV3MS`) taggers. Our open-source parser Alosaimy and Atwell (2016) that converts these variety of formats to CoNLL-U format, and JSON is available freely[9].

The analyses of the tagged sentence in appendix A shows that:

- Not only POS tags are different, but the word segmentation as well (word #2).

[9] http://sawaref.al-osaimy.com

- Word #10 shows that the definition of the lemma/stem is not standard: is it the PREP or the PRON. This can cause problems when evaluating different lemmatizers/stemmers for example.

- Some taggers do not recover a word's clitics. Instead it reports the POS tag of such clitics. Aligning such taggers with others can not be done intuitively.

- Two tokens sometimes are given one tag (KH analysis of word #10) even though the tag explains the two tokens: "a preposition and its pronoun".

- Some segmentation is for affixes not clitics (word #7), INDEF tag is related to the first segment though.

- In many cases, the first suggested analysis is the correct one: this is because some MAs sort alternative analyses. However, this should not be confused with POS taggers as POS taggers use the *context* to rank alternative analyses.

- The convention of diacritization is not standard. This includes short vowels before long vowels (word #1) and *tanween* location (before or after Alif letter) (word #2). A normalization is required if a comparison is to be performed.

6. Tagging Classical Texts

Most surveyed tools are designed primarily for MSA: the dataset used for training and testing is PATB which is an annotated corpus of news articles and stories. As mentioned earlier, Alrabiah et al. (2014) showed that CA has a worse POS tagging accuracy for MD and KH tools. We would like to compare between these taggers on a sample of CA. However, since taggers are different in their tagsets and segmentation conventions, a direct automatic evaluation is not possible (Paroubek, 2007).

Instead, we analysed 500 words that was extracted from classical books and are not common nowadays. Using OpenArabic Corpus (Dmitriev, 2016) which categorized these books into centuries and provided word frequencies for each book with and without normalization, we sum up non-normalized word frequencies of books that are written in the first 7 centuries (1075 books). We then truncated the word list to the top 500 words and drop any word that appeared at least once in the Corpus of Contemporary Arabic(Al-Sulaiti and Atwell, 2006). The final result was a list of 586 classical words.

Table 4 shows the rate of out of vocabulary (OOV) words, analysis time, average number of analyses per word, and average number of lemmas per word. Next, we compute their accuracy of tagging a sample of 50 words: We check the meaning of the 50 words by finding 10 concordances from the reduced corpus, and check if targeted POS tags were given by the analyser. Second column in table 4 shows the accuracy of each MA.

Then, we evaluate the performance of POS taggers. For each word in the list, we extracted three lines that contains the word, and pass it the POS tagger. Then, we evaluate the tagging of that word *in context*. Table 5 shows the overall accuracy, and the accuracy when we limit the word list to proper nouns.

Since each tagger has its own labelling schema, marking the tag as either correct or not is not easy. The marking was done by the first author. He had to manually check each tagger's output and decide. A tagger has to identify all clitics properly. We allow some tolerance for some tags (e.g. a proper noun with `noun` tag is correct, a verb with any verb tenses) to ensure fair comparison between taggers as not all of them are fine-grained.

We found that 30% of the words are proper nouns. They were rarely tagged as nouns by MAs. Alkhalil seems to have a list of classical proper nouns and performed the best in this matter. We also found that some words are common in contemporary Arabic, but make it to this list as they appeared with some affixes.

The word frequencies reported by OpenArabic are simple word frequencies, instead of TF/IDF, which raised some words that are highly frequent but only on certain books (e.g. dictionaries like بضم*bḍm* (with a Dammaah vowel), prophet sayings like ثنَا*ṯnā* (he reported), bibliography like some proper nouns).

Some sources of mistagging:

- One common adverb was only properly tagged by one analyser, as this adverb is obsolete.

- Normalization of converting Yaa Maqsourah to Yaa, a proper noun was not tagged properly.

- Different classical tokenization such as يَا أيهَا *yā ʾyhā* (O (mankind)) which was written jointly.

- Some words were not identified as the broken plural pattern is obsolete (like القرءة*ālqrʾh* (the readers))

Table 5 gives evidence that one POS tagger performs better in some tags than the other. MADAMIRA toolkit (MA) performed poorly with classical proper nouns as those words either are not covered in its ALMORGEANA lexicon or are mistagged as another word in its lexicon. However, it outperforms other taggers in tagging other words. This suggests that an ensemble POS tagger could increase the accuracy of POS tagging. Other works came to the same conclusion which suggested the same conclusion Aliwy (2015); Alabbas and Ramsay (2012); Alosaimy and Atwell (2016).

Tool	AR	AL	KH	EX	BP	MS	QT
OOV	0.228	0	0.058	0.076	0.084	0.052	0.82
Accuracy	0.560	0.88	0.9	0.84	0.88	0.82	N/A
Analysis Time (in secs)	0.255	4.324	3.453	177.465	1.061	N/A°	0.766
Avg. Analysis/Word	2.06	7.32	14.25	17.89	2.44	1.86	4.27
Avg. Lemmas/Word	1.5	2.53	4.51	2.61	2	1.53	1

Table 4: The rate of Out of Vocabulary (OOV), analysis time, average number of analyses/lemmas of tagging 500 common classical words. Accuracy was computed on a sample of 50 words. AL used backoff when no analysis was found in the dictionary (OOV is zero). QT does not provide lemmatization. ° not available as it is web based service.

	MD	MA	ST	MR	WP	AM	MT	FA
Overall	.696	.706	.784	.667	.686	.794	.676	.745
No Prop Nouns	.8	.785	.714	.528	.585	.742	.871	.742
Prop. Nouns	.468	.531	.937	.968	.906	.906	.250	.750

Table 5: The accuracy of POS taggers of tagging 50 classical words within three sentences per word extracted from classical books.

7. Conclusion

POS taggers and morphological analysers differ in many aspects. While they both predict the main part of speech tag, they vary on what morphological and word features they also predict. Most taggers adapt their own tagset, and they subsequently assume its tokenization scheme. In our experiment, the accuracy and coverage has dropped to low level when applying these taggers on CA texts.

For future work, we think that standrization in Arabic POS tagging is still not tackled. This includes standarization in diacritization, lemmatization, POS tagset, and morphological features. We think at least newly released resources should be backward compatible with one other resource. Some linguistic issues like the definition of lemma, root, and stem should be standarized as well. We noticed as well that some newly techniques such as neural networks have not been employed.

In regard to CA, the annotation of classical text should either adapt its own new morphological analyser or improve current ones to support classical Arabic. One alternative solution is to combine those taggers in one system which should increase the coverage and accuracy levels, as we noticed that errors from analysers differ and combining them will increase the coverage and subsequently improve the accuracy. However, this approach is not easy as taggers implement different tagsets and tokenization schemes.

8. Acknowledgement

We would like to thank the anonymous reviewers for their careful reading of our manuscript and their many insightful comments and suggestions. Abdulrahman Alosaimy owes his deepest gratitude to Al-Imam University in Riyadh, Saudi Arabia for funding this research and his PhD study. We thank also the developers of morphological analysers and POS taggers and gold standard corpora for making their tools and resources available for use.

Appendices

Appendix A Tagged Sentence

This section shows a full sentence of one Hadith (prophet sayings) annotated in parallel by several morphological analysers and POS taggers. Columns represent the abbrevation of the tool, word id with morpheme id (if detected), lemma, assigned POS-tag, and analysed morphological features such as gender (if available).

A.1 Morphological Analysers

```
AL      1       lA      lA_1    part_neg        -       ANALSIS#=1/1
AR      1       lA      -       part_neg        -       ANALSIS#=2/2
BP      1       lA      -       NEG_PART        -       ANALSIS#=1/1
EX      1       laA     laA     F-      -       ANALSIS#=1/3
KH      1       laA     laA     Hrf nfy -       ANALSIS#=2/3
MS      1       laA     laA     HARF_NAFY       -       ANALSIS#=1/1
QT      1       lAa     -       pc      -       ANALSIS#=1/2

AL      2       yu&omin |man_1 verb   Gender=M|Number=S|Aspect=IMPF|Voice=ACT|Person=3  ANALSIS#=3/4
AR      2       >Amn    -       verb   Gender=M|Number=S|Aspect=IMPF|Voice=ACT|Person=3  ANALSIS#=2/2
BP      2-0     yu      -       IV3MS  Gender=M|Number=S|Aspect=IMPF|Voice=ACT|Person=3  ANALSIS#=2/4
BP      2-1     &omin   |man_1 VERB_IMPERFECT -      ANALSIS#=2/4
EX      2       yu&minu |man    VI     Gender=M|Number=S|Mood=IND|Aspect=IMPF|Voice=ACT|Person=3 ANALSIS#=1/1
KH      2       yu&am~inu       >am~ana fEl mDArE mbny llmElwm Case=NOM|Aspect=IMPV|Person=3 ANALSIS#=1/47
MS      2-0     -       -       PREFIX_YA2_ANAIT_MA3LOOM_MAGHOOL    -       ANALSIS#=1/8
MS      2-1     yu&omin yu&omin FE3L_MODARE3_MAZEED Aspect=IMPF   ANALSIS#=1/8
QT      2       UNK-WORD

AL      3-0     >aHadkum        >aHad_1 noun   Gender=M|Number=S|Case=-     ANALSIS#=1/9
AL      3-1     -       -       2mp_poss        -       ANALSIS#=1/9
AR      3-0     >Hd     -       noun   Gender=M|Number=S     ANALSIS#=1/8
AR      3-1     _km     -       genpron Gender=M|Number=P|Person=2  ANALSIS#=1/8
BP      3-0     >aHad   >aHad_1 NOUN   -       ANALSIS#=2/9
BP      3-1     kum     -       POSS_PRON_2MP -       ANALSIS#=2/9
EX      3-0     >aHadu  >aHad   N-     Number=S|Case=NOM     ANALSIS#=1/4
EX      3-1     kum     huwa    SP     Gender=M|Number=P|Case=ACC|Person=2 ANALSIS#=1/4
KH      3-0     >aHadakumo      >aHad  Asm jAmd       Gender=M|Number=S|Case=ACC   ANALSIS#=3/37
KH      3-1     -       -       kumo: Dmyr AlmxATbyn -      ANALSIS#=3/37
MS      3-0     >aHad~akumo     >aHad~a AF3AL_TA3AGOB -      ANALSIS#=1/1
MS      3-1     -       -       SUFFIX_KUM_MOKHATAB_GAM3_MOTHAKAR   Number=P|Person=2     ANALSIS#=1/1
QT      3       UNK-WORD

AL      4       Hat~aY Hat~aY_1       prep    -       ANALSIS#=1/3
AR      4       HtY     -       prep    -       ANALSIS#=1/1
BP      4       Hat~aY -        PREP    -       ANALSIS#=1/3
EX      4       Hat~aY Hat~aY P-        -       ANALSIS#=1/3
KH      4       Hat~aY Hat~aY Hrf ETf -       ANALSIS#=2/2
MS      4       Hat~aY Hat~aY HARF_GARR       -       ANALSIS#=1/1
QT      4       HatY~a -        pp      -       ANALSIS#=1/3

AL      5       yakuwn kAn_1   verb   Gender=M|Number=S|Aspect=IMPF|Voice=ACT|Person=3  ANALSIS#=1/3
AR      5       -       kaw~an verb   Gender=M|Number=S|Aspect=IMPF|Voice=PASS|Person=3 ANALSIS#=2/5
BP      5-0     ya      -       IV3MS  Gender=M|Number=S|Aspect=IMPF|Voice=ACT|Person=3  ANALSIS#=2/4
BP      5-1     kuwn    kAn_1  VERB_IMPERFECT -      ANALSIS#=2/4
EX      5       yakuwna kaAn   VI     Gender=M|Number=S|Mood=SUBJ|Aspect=IMPF|Voice=ACT|Person=3 ANALSIS#=1/2
KH      5       yukowun~a       >akowaY fEl mDArE m&kd mbny llmElwm Aspect=IMPV|Person=3  ANALSIS#=1/18
MS      5-0     -       -       PREFIX_YA2_ANAIT_MA3LOOM      Voice=ACT      ANALSIS#=1/5
MS      5-1     yakuwn yakuwn FE3L_MODARE3_MOGARRAD Aspect=IMPF   ANALSIS#=1/5
QT      5       UNK-WORD

AL      6-0     hawAh   hawaY_1 noun   Gender=M|Number=S|Case=-     ANALSIS#=1/5
AL      6-1     -       -       3ms_poss        -       ANALSIS#=1/5
AR      6-0     hwY     -       noun   Gender=M|Number=S     ANALSIS#=1/1
AR      6-1     _h      -       genpron Gender=M|Number=S|Person=3      ANALSIS#=1/1
BP      6-0     hawA    hawaY_1 NOUN            -       ANALSIS#=4/4
BP      6-1     hu      -       POSS_PRON_3MS   -       ANALSIS#=4/4
EX      6-0     hawaY   hawaY   N-      Number=S|Case=NOM       ANALSIS#=3/5
EX      6-1     hu      huwa    SP      Gender=M|Number=S|Case=ACC|Person=3     ANALSIS#=3/5
KH      6-0     hawaAhu hawFY   Asm jAmd        Gender=M|Number=S|Case=NOM      ANALSIS#=1/8
KH      6-1     -       -       hu: Dmyr AlgA}b -       ANALSIS#=1/8
MS      6-0     hawaAhu hawaY   MASDAR_MOGARRAD -       ANALSIS#=1/1
MS      6-1     -       -       SUFFIX_HA2_MODAF_GHA2EB_MOTHAKKAR       Gender=M|Person=3       ANALSIS#=1/1
QT      6       UNK-WORD

AL      7       tabaEAF tabaEAF_1       adv     Gender=M|Number=S|Case=ACC      ANALSIS#=1/3
AR      7       tbEAF   -       adv     -       ANALSIS#=4/4
BP      7-0     tabaE   tabaEAF_1       ADV     -       ANALSIS#=3/3
BP      7-1     AF      -       NSUFF_MASC_SG_ACC_INDEF -       ANALSIS#=3/3
EX      7       tabaEFA tabaE   N-      Number=S|Case=GEN       ANALSIS#=3/3
KH      7       tiboEFA tiboE   Asm jAmd        Gender=M|Number=S|Case=ACC      ANALSIS#=2/26
MS      7-0     tabaEFA tabaEFA MASDAR_MOGARRAD -       ANALSIS#=2/2
MS      7-1     -       -       SUFFIX_ALEF_TANWEEN     -       ANALSIS#=2/2
QT      7       UNK-WORD


AL      8-0     li      -       prep    -       ANALSIS#=4/4
AL      8-1     mA      mA_1    pron_rel        Gender=M|Number=S|Case=-        ANALSIS#=4/4
AR      8-0     l_      -       prep    -       ANALSIS#=2/8
AR      8-1     mA      -       rel     Number=S        ANALSIS#=2/8
BP      8-0     li      -       PREP    -       ANALSIS#=2/4
BP      8-1     mA      limA_1  REL_PRON        -       ANALSIS#=2/4
EX      8-0     li      li      P-      -       ANALSIS#=2/3
EX      8-1     maA     maA     S-      -       ANALSIS#=2/3
KH      8-0     -       -       li : Hrf Aljr   -       ANALSIS#=11/11
KH      8-1     limaA   maA     Asm mwSwl       -       ANALSIS#=11/11
MS      8-0     -       -       PREFIX_LAM_GARR -       ANALSIS#=1/2
MS      8-1     limaA   maA     ESM_MAWSOOL     -       ANALSIS#=1/2
QT      8       limaA   -       nc      Case=GEN        ANALSIS#=1/2

AL      9       ji}ota  jA’_1   verb    Gender=M|Number=S|Mood=IND|Aspect=PERF|Voice=ACT|Person=2 ANALSIS#=1/3
AR      9       jA’     -       verb    Aspect=PERF|Voice=ACT|Person=1 ANALSIS#=1/3
BP      9-0     ji}     jA’_1   VERB_PERFECT    -       ANALSIS#=1/3
BP      9-1     tu      -       PVSUFF_SUBJ:1S Number=S|Aspect=PERF|Voice=ACT|Person=1 ANALSIS#=1/3
EX      9       ji}tu   jaA’    VP      Gender=M|Number=S|Aspect=PERF|Voice=ACT|Person=1  ANALSIS#=1/4
KH      9       ji}otu  jaA’a   fEl mAD mbny llmElwm    Person=1        ANALSIS#=3/3
MS      9-0     ji}otu  jaA’a   FE3L_MADI_MOGARRAD      Aspect=PERF     ANALSIS#=1/1
MS      9-1     -       -       SUFFIX_TA2_FA3EL_MOTAKALLEM  Person=1       ANALSIS#=1/1
QT      9       UNK-WORD

AL      10-0    bihi    bi_1    prep    -       ANALSIS#=1/1
AL      10-1    -       -       3ms_pron        -       ANALSIS#=1/1
AR      10-0    b_      -       prep    -       ANALSIS#=1/1
AR      10-1    _h      -       objcon  Gender=M|Number=S|Person=3      ANALSIS#=1/1
BP      10-0    bi      -       PREP    -       ANALSIS#=1/1
BP      10-1    hi      bi-_1   PRON_3MS        -       ANALSIS#=1/1
EX      10-0    bi      bi      P-      -       ANALSIS#=1/1
EX      10-1    hi      huwa    SP      Gender=M|Number=S|Case=ACC|Person=3     ANALSIS#=1/1
KH      10      bihi    bihi    jAr wmjrwr      -       ANALSIS#=8/17
MS      10-0    bihi    bi      HARF_GARR       -       ANALSIS#=1/1
MS      10-1    -       -       SUFFIX_HA2_MODAF_GHA2EB_MOTHAKKAR       Gender=M|Person=3       ANALSIS#=1/1
QT      10      UNK-WORD
```

A.2 POS taggers

```
AM      1       lA      -       RP      -       ANALSIS#=1/1
FA      1       lA      -       PART    -       ANALSIS#=1/1
MA      1       lA      lA_1    part_neg        -       ANALSIS#=1/1
MD      1       lA      lA_1    part_neg        -       ANALSIS#=1/1
MR      1       lA      -       RP      -       ANALSIS#=1/1
```

```
ST      1       lA      -       RP      -       ANALSIS#=1/1
WP      1       lA      -       part_neg        -       ANALSIS#=1/1

AM      2       y&mn    -       VBP     Aspect=IMPF|Voice=ACT|Person=2 ANALSIS#=1/1
FA      2       y&mn    -       V       -       ANALSIS#=1/1
MA      2       yu&omin |man_1 verb     Gender=M|Number=S|Aspect=IMPF|Voice=ACT|Person=3        ANALSIS#=1/1
MD      2       yu&omin |man_1 verb     Gender=M|Number=S|Aspect=IMPF|Voice=ACT|Person=3        ANALSIS#=1/1
MR      2       ymn     -       VBP     -       ANALSIS#=1/1
ST      2       y&mn    -       VBP     Aspect=IMPF|Voice=ACT ANALSIS#=1/1
WP      2       yu’minu -       verb    -       ANALSIS#=1/1

AM      3-0     >Hd     -       NN      -       ANALSIS#=1/1
AM      3-1     km      -       PRP     Person=2        ANALSIS#=1/1
FA      3-0     >Hd     -       NOUN    Person=1        ANALSIS#=1/1
FA      3-1     km      -       PRON    -       ANALSIS#=1/1
MA      3-0     >aHadakum       >aHad_1 noun    Gender=M|Number=S|Case=ACC      ANALSIS#=1/1
MA      3-1     -       -       2mp_poss        -       ANALSIS#=1/1
MD      3-0     >aHadkum        >aHad_1 noun    Gender=M|Number=S|Case=-        ANALSIS#=1/1
MD      3-1     -       -       2mp_poss        -       ANALSIS#=1/1
MR      3-0     AHd     -       NN      -       ANALSIS#=1/1
MR      3-1     +km     -       PRP$    -       ANALSIS#=1/1
ST      3-0     AHd     -       NN      Number=S        ANALSIS#=1/1
ST      3-1     km      -       PRP$    -       ANALSIS#=1/1
WP      3       AHadukum        -       noun    -       ANALSIS#=1/1

AM      4       HtY     -       CJP     -       ANALSIS#=1/1
FA      4       HtY     -       PREP    -       ANALSIS#=1/1
MA      4       Hat~aY  Hat~aY_1        prep    -       ANALSIS#=1/1
MD      4       Hat~aY  Hat~aY_1        prep    -       ANALSIS#=1/1
MR      4       Hty     -       AN      -       ANALSIS#=1/1
ST      4       HtY     -       IN      -       ANALSIS#=1/1
WP      4       Hat~ay  -       noun    -       ANALSIS#=1/1

AM      5       ykwn    -       VBP     Aspect=IMPF|Voice=ACT|Person=2 ANALSIS#=1/1
FA      5       ykwn    -       V       -       ANALSIS#=1/1
MA      5       yakuwn  kAn_1   verb    Gender=M|Number=S|Aspect=IMPF|Voice=ACT|Person=3        ANALSIS#=1/1
MD      5       yakuwn  kAn_1   verb    Gender=M|Number=S|Aspect=IMPF|Voice=ACT|Person=3        ANALSIS#=1/1
MR      5       ykwn    -       VBP     -       ANALSIS#=1/1
ST      5       ykwn    -       VBP     Aspect=IMPF|Voice=ACT ANALSIS#=1/1
WP      5       yakwna  -       verb    -       ANALSIS#=1/1

AM      6-0     hwY     -       NN      -       ANALSIS#=1/1
AM      6-1     h       -       PRP     Person=2        ANALSIS#=1/1
FA      6-0     hwA     -       NOUN    Person=1        ANALSIS#=1/1
FA      6-1     h       -       PRON    -       ANALSIS#=1/1
MA      6-0     hawAh   hawaY_1 noun    Gender=M|Number=S|Case=-        ANALSIS#=1/1
MA      6-1     -       -       3ms_poss        -       ANALSIS#=1/1
MD      6-0     hawAh   hawaY_1 noun    Gender=M|Number=S|Case=-        ANALSIS#=1/1
MD      6-1     -       -       3ms_poss        -       ANALSIS#=1/1
MR      6-0     hwy     -       NN      -       ANALSIS#=1/1
MR      6-1     +h      -       PRP$    -       ANALSIS#=1/1
ST      6-0     hwA     -       NN      Number=S        ANALSIS#=1/1
ST      6-1     h       -       PRP$    -       ANALSIS#=1/1
WP      6       hawAhu  -       noun    -       ANALSIS#=1/1

AM      7       tbEA    -       NN      -       ANALSIS#=1/1
FA      7-0     tbE     -       NOUN    Person=1        ANALSIS#=1/1
FA      7-1     A       -       CASE    -       ANALSIS#=1/1
MA      7       tabaEAF tabaE_1 noun    Gender=M|Number=S|Case=ACC      ANALSIS#=1/1
MD      7       tabaEAF tabaE_1 noun    Gender=M|Number=S|Case=ACC      ANALSIS#=1/1
MR      7       tbEA    -       NN      -       ANALSIS#=1/1
ST      7       tbEA    -       NN      Number=S        ANALSIS#=1/1
WP      7       tabaEAF -       verb    -       ANALSIS#=1/1

AM      8-0     l       -       IN      -       ANALSIS#=1/1
AM      8-1     mA      -       WP      -       ANALSIS#=1/1
FA      8-0     l+      -       PREP    -       ANALSIS#=1/1
FA      8-1     mA      -       PART    -       ANALSIS#=1/1
MA      8-0     li      -       prep    -       ANALSIS#=1/1
MA      8-1     mA      mA_1    pron_rel        Gender=M|Number=S|Case=-        ANALSIS#=1/1
MD      8-0     li      -       prep    -       ANALSIS#=1/1
MD      8-1     mA      mA_1    pron_rel        Gender=M|Number=S|Case=-        ANALSIS#=1/1
MR      8-0     l#      -       IN      -       ANALSIS#=1/1
MR      8-1     mA      -       WP      -       ANALSIS#=1/1
ST      8-0     l       -       IN      -       ANALSIS#=1/1
ST      8-1     mA      -       WP      -       ANALSIS#=1/1
WP      8       limA    -       noun_prop       -       ANALSIS#=1/1

AM      9       j}t     -       VBD     Aspect=PERF|Voice=ACT|Person=2 ANALSIS#=1/1
FA      9-0     j}      -       V       -       ANALSIS#=1/1
FA      9-1     t       -       PRON    -       ANALSIS#=1/1
MA      9       ji}otu  jA’_1   verb    Gender=M|Number=S|Mood=IND|Aspect=PERF|Voice=ACT|Person=1 ANALSIS#=1/1
MD      9       ji}otu  jA’_1   verb    Gender=M|Number=S|Mood=IND|Aspect=PERF|Voice=ACT|Person=1 ANALSIS#=1/1
MR      9       jt      -       VBD     -       ANALSIS#=1/1
ST      9       j}t     -       VBD     Aspect=PERF|Voice=ACT ANALSIS#=1/1
WP      9       ji’tu   -       noun_prop       -       ANALSIS#=1/1

AM      10-0    b       -       IN      -       ANALSIS#=1/1
AM      10-1    h       -       PRP     Person=2        ANALSIS#=1/1
FA      10-0    b+      -       PREP    -       ANALSIS#=1/1
FA      10-1    h       -       PRON    -       ANALSIS#=1/1
MA      10-0    bihi    bi_1    prep    -       ANALSIS#=1/1
MA      10-1    -       -       3ms_pron        -       ANALSIS#=1/1
MD      10-0    bihi    bi_1    prep    -       ANALSIS#=1/1
MD      10-1    -       -       3ms_pron        -       ANALSIS#=1/1
MR      10-0    b#      -       IN      -       ANALSIS#=1/1
MR      10-1    +h      -       PRP     -       ANALSIS#=1/1
ST      10-0    b       -       IN      -       ANALSIS#=1/1
ST      10-1    h       -       PRP     -       ANALSIS#=1/1
WP      10      bihi    -       noun_prop       -       ANALSIS#=1/1
```

Appendix B Output Format Differences

```
SOLUTION #1
Lemma :       jA'
Vocalized as :  ji}tu
Morphology :
       prefix : Pref-0
       stem : PV_C
       suffix : PVSuff-t
Grammatical category :
       stem : ji}    VERB_PERFECT
       suffix : tu   PVSUFF_SUBJ:1S
Glossed as :
       stem : arrive/come/occur
       suffix : I <verb>

... 2 more solutions
```

(a) Java

```
INPUT STRING: j}t
LOOK-UP WORD: j}t
  SOLUTION 1: (ji}otu) [jA'_1
] ji}/VERB_PERFECT+tu/PVSUFF_SUBJ:1S
     (GLOSS): + arrive/come/occur + I <verb>
  SOLUTION 2: (ji}ota) [jA'_1
] ji}/VERB_PERFECT+ta/PVSUFF_SUBJ:2MS
     (GLOSS): + arrive/come/occur + you [masc.sg.] <verb>
  SOLUTION 3: (ji}oti) [jA'_1
] ji}/VERB_PERFECT+ti/PVSUFF_SUBJ:2FS
     (GLOSS): + arrive/come/occur + you [fem.sg.] <verb>
```

(b) Perl

Figure 1: A sample of the output of AraMorph in two versions Java and Perl. On Perl version, each solution has the vocalized word (in parenthesis), lemma (in square brackets), analyses of each segments where segments are separated by plus sign, and finally a helper glossary.

Input	Voweled Word	Prefix	Stem	Type	Pattern	Root	POS Tags	Suffix
جِئْتُ *ǧiʾtu*	جِئْتُ *ǧiʾtu*	#	جءت *ǧʾt*	فعل مَاض مبني لِّلمعلوم *fʿl māḍ mbny llmʿlwm*	فِلْتُ *filtu*	جيء *ǧyʾ*	ثِلَاثي مجرد مسنِد أَلِى المتكلم أنَا متعد ولَازم *ṯlāṯy mǧrd msnd ʾlā ālmtklm ʾnā mtʿd wlāzm*	تَاء المتكلم *tāʾ ālmtklm*
ji}otu	ji}otu	#	j}t	Active perfect verb	1i2o3u	jy´	VIII Unaugmented first-Person Transitive and Intransitive	t: t of first-person

Table 6: Alkhalil output of one analysis of the word "ji}otu" is on the first row. We added a new row for translating the output shown in the first row. It is clear that the POS tags the type of the word is not in a good reusable format.

```
j}t    +verb+past+activejA'+1pers@
j}t    +verb+past+activejA'+2pers+sg+masc@
j}t    +verb+past+activejA'+2pers+sg+fem@
```

Figure 2: A sample of the output of AraComLex.

```
:::: ji}otu

 ::: <^gi'tu>

  :: (792,1)   ["arrive","come","occur"]
               Verb [] [FIL] []      [I]
               ^gA'   "^g y '"       FAL    jaA'   jA'
   : <^gi'tu>  ji}tu  j}t
     VP-A-1MS-- ^gi'tu "^g y '"      FiL |<< "tu"   ji}tu  j}t
   : <^gi'tu>  ji}tu  j}t
     VP-A-1FS-- ^gi'tu "^g y '"      FiL |<< "tu"   ji}tu  j}t
   : <^gi'tu>  ji}tu  j}t
     VP-P-1MS-- ^gi'tu "^g y '"      FiL |<< "tu"   ji}tu  j}t
   : <^gi'tu>  ji}tu  j}t
     VP-P-1FS-- ^gi'tu "^g y '"      FiL |<< "tu"   ji}tu  j}t
```

Figure 3: A sample of the output of Elixir FM. Each analysis has seven columns(e.g. first column is an eight-slot string that represent the POS tag and morphological features).

```
<Word number_of_possibilities="2" original_string="limaA">
       <SurfaceFormMorphemes certainty="0.8125" voweled_form="limaA">
              <Proclitcs>
                     <Proclitc arabic_description="Hrf, Hrf jr, ZAhr" tag="p,p"/>
              </Proclitcs>
              <Cliticless arabic_description="Asm, m*kr >w m&nv, mfrd >w mvnY >w jmE, ?, mjrwr, Asm mwSwl
                    m$trk, mErfp, ZAhr" tag="n,mf,sdp,?,g,c,d"/>
              <Enclitics/>
       </SurfaceFormMorphemes>
       <SurfaceFormMorphemes certainty="0.5" voweled_form="limaA">
              <Proclitcs>
                     <Proclitc arabic_description="Hrf, Hrf jr, ZAhr" tag="p,p"/>
              </Proclitcs>
              <Cliticless arabic_description="Asm, ?, ?, ?, mjrwr, Asm $rT, nkrp, ZAhr" tag="n,?,?,?,g,h,i
                    "/>
              <Enclitics/>
       </SurfaceFormMorphemes>
</Word>
```

Figure 4: A sample of the XML output of Qutuf System.

```
;;WORD j}t
diac:ji}ota lex:jA’_1 bw:+ji}/PV++ta/PVSUFF_SUBJ:2MS gloss:arrive/come/occur pos:verb prc3:0 prc2:0 prc1:0
      prc0:0 per:2 asp:p vox:a mod:i gen:m num:s stt:na cas:na enc0:0 rat:na source:lex stem:ji} stemcat:
      PV_C
diac:ji}oti lex:jA’_1 bw:+ji}/PV++ti/PVSUFF_SUBJ:2FS gloss:arrive/come/occur pos:verb prc3:0 prc2:0 prc1:0
      prc0:0 per:2 asp:p vox:a mod:i gen:f num:s stt:na cas:na enc0:0 rat:na source:lex stem:ji} stemcat:
      PV_C
diac:ji}otu lex:jA’_1 bw:+ji}/PV++tu/PVSUFF_SUBJ:1S gloss:arrive/come/occur pos:verb prc3:0 prc2:0 prc1:0
      prc0:0 per:1 asp:p vox:a mod:i gen:m num:s stt:na cas:na enc0:0 rat:na source:lex stem:ji} stemcat:
      PV_C
```

Figure 5: A sample of the output of ALMORGEANA. The representation of the analysis is like

```
;;WORD j}t
;;SVM_PREDICTIONS: j}t asp:p cas:na enc0:0 gen:m mod:i num:s per:1 pos:verb prc0:0 prc1:0 prc2:0 prc3:0 stt:
      na vox:a
*1.000126 diac:ji}otu lex:jA’_1 bw:+ji}/PV++tu/PVSUFF_SUBJ:1S gloss:arrive/come/occur pos:verb prc3:0 prc2:0
       prc1:0 prc0:0 per:1 asp:p vox:a mod:i gen:m num:s stt:na cas:na enc0:0 rat:na source:lex stem:ji}
      stemcat:PV_C
_0.944387 diac:ji}ota lex:jA’_1 bw:+ji}/PV++ta/PVSUFF_SUBJ:2MS gloss:arrive/come/occur pos:verb prc3:0 prc2
      :0 prc1:0 prc0:0 per:2 asp:p vox:a mod:i gen:m num:s stt:na cas:na enc0:0 rat:na source:lex stem:ji}
      stemcat:PV_C
_0.910868 diac:ji}oti lex:jA’_1 bw:+ji}/PV++ti/PVSUFF_SUBJ:2FS gloss:arrive/come/occur pos:verb prc3:0 prc2
      :0 prc1:0 prc0:0 per:2 asp:p vox:a mod:i gen:f num:s stt:na cas:na enc0:0 rat:na source:lex stem:ji}
      stemcat:PV_C
```

Figure 6: A sample of the output of MADA. It is identical to ALMORAGRANA with ranked solution (first column). Starred solutions are the selected solution.

```
;;WORD j}t
;;LENGTH 3
;;OFFSET 37
;;SVM_PREDICTIONS: j}t diac:ji}otu lex:jA’ asp:p cas:na enc0:0 gen:m mod:i num:s per:1 pos:verb prc0:0 prc1
      :0 prc2:0 prc3:0 stt:na vox:a
*0.893935 diac:ji}otu lex:jA’_1 bw:ji}/PV+tu/PVSUFF_SUBJ:1S gloss:arrive/come/occur sufgloss:I_<verb> pos:
      verb prc3:0 prc2:0 prc1:0 prc0:0 per:1 asp:p vox:a mod:i gen:m num:s stt:na cas:na enc0:0 rat:na
      source:lex stem:ji} stemcat:PV_C
_0.856916 diac:ji}ota lex:jA’_1 bw:ji}/PV+ta/PVSUFF_SUBJ:2MS gloss:arrive/come/occur sufgloss:you_[masc.sg.]_
      <verb> pos:verb prc3:0 prc2:0 prc1:0 prc0:0 per:2 asp:p vox:a mod:i gen:m num:s stt:na cas:na enc0:0
      rat:na source:lex stem:ji} stemcat:PV_C
_0.830216 diac:ji}oti lex:jA’_1 bw:ji}/PV+ti/PVSUFF_SUBJ:2FS gloss:arrive/come/occur sufgloss:you_[fem.sg.]_<
      verb> pos:verb prc3:0 prc2:0 prc1:0 prc0:0 per:2 asp:p vox:a mod:i gen:f num:s stt:na cas:na enc0:0
      rat:na source:lex stem:ji} stemcat:PV_C
```

Figure 7: A sample of the output of MADAMIRA: Like MADA output with `sufgloss` (suffix gloss) feature.

```
1       lA      _       _       _       PRT|RP _        NEG|PART
2       ymn     _       _       _       VRB|VBP _       IV3MS|IV|IVSUFF|MOOD|I
3       AHd     _       _       _       NOM|NN _        NOUN|CASE|DEF|ACC
4       km      _       _       _       NOM|PRP$        _       POSS|PRON|2MP
5       Hty     _       _       _       PRT|AN _        SUB|CONJ
6       ykwn    _       _       _       VRB|VBP _       IV3MS|IV|IVSUFF|MOOD|S
7       hwy     _       _       _       NOM|NN _        NOUN
8       h       _       _       _       NOM|PRP$        _       POSS|PRON|3MS
9       tbEA    _       _       _       NOM|NN _        NOUN|CASE|INDEF|ACC
10      l       _       _       _       PRT|IN _        PREP
11      mA      _       _       _       NOM|WP _        REL|PRON
12      jt      _       _       _       VRB|VBD _       PV|PVSUFF|SUBJ|3FS
13      b       _       _       _       PRT|IN _        PREP
14      h       _       _       _       NOM|PRP _       PRON|3MS
```

Figure 8: A sample of the output of MarMoT.

```
# 0 0.554063
lA      part_neg+none+none+none part_neg+none+none+none/0.999943
y&mn    verb+none+none+none   verb+none+none+none/0.999972
>Hdkm   noun+none+none+none   noun+none+none+none/0.974859
HtY     prep+none+none+none   prep+none+none+none/0.682635
ykwn    verb+none+none+none   verb+none+none+none/0.950193
hwAh    noun+none+none+none   noun+none+none+none/0.969479
tbEA    noun+none+none+none   noun+none+none+none/0.979848
lmA     pron_rel+none+PREP+none pron_rel+none+PREP+none/0.922642
j}t     verb+none+none+none   verb+none+none+none/0.999986
bh      prep+none+PREP+none   prep+none+PREP+none/0.999839
```

Figure 9: A sample of the output of SAPA.

```
#ST
lA/RP y&mn/VBP_MS3 >Hd/NN +km/PRP_MP2 HtY/CJP ykwn/VBP_MS3 hwY/NN +h/PRP_MS3 tbEA/NN l#/IN mA/WP j}t/VBD_FS3
      b#/IN +h/PRP_MS3
#AM
lA/RP y&mn/VBP AHd/NN km/PRP$ HtY/IN ykwn/VBP hwA/NN h/PRP$ tbEA/NN l/IN mA/WP j}t/VBD b/IN h/PRP
#FA
S/S lA/PART y&mn/V >Hd/NOUN-MS +km/PRON HtY/PREP ykwn/V hwA/NOUN-MS +h/PRON tbE/NOUN-MS +A/CASE l+/PREP +mA/
     PART j}/V +t/PRON b+/PREP +h/PRON E/E
```

Figure 10: A sample of the output of Stanford POS Tagger, AMIRA, and Farasa. Standford does not mark segmented morphemes (e.g for regrouping later).

References

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

Al-Sulaiti, L. and Atwell, E. S. (2006). The design of a corpus of contemporary Arabic. *International Journal of Corpus Linguistics*, 11(2):135–171.

Alabbas, M. and Ramsay, A. (2012). Improved POS-Tagging for Arabic by Combining Diverse Taggers. In Iliadis, L., Maglogiannis, I., and Papadopoulos, H., editors, *8th International Conference on Artificial Intelligence Applications and Innovations (AIAI)*, volume AICT-381 of *Artificial Intelligence Applications and Innovations*, pages 107–116, Halkidiki, Greece. Springer.

Albared, M., Omar, N., and Ab Aziz, M. J. (2009). Arabic part of speech disambiguation: A survey. *International Review on Computers and Software*, 4(5):517–532.

Aliwy, A. H. (2015). Combining Pos Taggers in Master-Slaves Technique for Highly Inflected Languages As Arabic. In *2015 International Conference on Cognitive Computing and Information Processing(CCIP)*, pages 1–5.

Alosaimy, A. and Atwell, E. (2015). A review of morphosyntactic analysers and tag-sets for Arabic corpus linguistics. In *Eighth International Corpus Linguistics conference (CL2015)*, pages 16–19.

Alosaimy, A. and Atwell, E. (2016). Ensemble Morphosyntactic Analyser for Classical Arabic. In *2nd International Conference on Arabic Computational Linguistics*, Konya, Turkey.

Alrabiah, M., Al-Salman, A., Atwell, E. S., Alhelewh, N., Alrabia, M., Al-Salman, A., Atwell, E. S., and Alhelewh, N. (2014). KSUCCA: a key to exploring Arabic historical linguistics. *International Journal of Computational Linguistics (IJCL)*, 5(2):27–36.

Altabbaa, M., Al-zaraee, A., and Shukairy, M. A. (2010). *An Arabic Morphological Analyzer and Part-Of-Speech Tagger*. Master thesis, Arab International University, Damascus, Syria.

Attia, M., Pecina, P., Toral, A., and Van Genabith, J. (2014). A corpus-based finite-state morphological toolkit for contemporary arabic. *Journal of Logic and Computation*, 24(2):455–472.

Atwell, E., Al-Sulaiti, L., Al-osaimi, S., and Shawar, B. A. (2004). A Review of Arabic Corpus Analysis Tools. In *Proceedings of JEP-TALN Arabic language processing*, pages 229–234, Fez, Morocco.

Beesley, K. R. (1998). Arabic Morphology Using Only Finite-State Operations. *Proceedings of the Workshop on Computational Approaches to Semitic languages*, pages 50–57.

Benajiba, Y. and Zitouni, I. (2010). Arabic Word Segmentation for Better Unit of Analysis. *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, pages 1346–1352.

Bin-Muqbil, M. S. (2006). *Phonetic And Phonological Aspects Of Arabic Emphatics And Gutturals*. PhD thesis, University Of Wisconsin-madison.

Boudchiche, M., Mazroui, A., Bebah, M. O. A. O., Lakhouaja, A., and Boudlal, A. (2016). AlKhalil Morpho Sys 2: A robust Arabic morpho-syntactic analyzer. *Journal of King Saud University-Computer and Information Sciences*.

Buckwalter, T. (2002). Arabic Morphological Analyzer (AraMorph).

Diab, M. Second generation AMIRA tools for Arabic processing: Fast and robust tokenization, POS tagging, and base phrase chunking. *Conference on Arabic Language Resources and Tools*, pages 285–288.

Diab, M., Habash, N., Rambow, O., and Roth, R. (2013). LDC Arabic Treebanks and Associated Corpora: Data Divisions Manual. *ACL, Short Papers*.

Diab, M., Hacioglu, K., and Jurafsky, D. (2004). Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. *HLT-NAACL 2004: Short Papers*, pages 149–152.

Dmitriev, K. (2016). Open Arabic Project. https://github.com/OpenArabic/Annotation.

El-haj, M. and Koulali, R. (2013). KALIMAT a multipurpose Arabic Corpus. In *Second Workshop on Arabic Corpus Linguistics (WACL-2)*, pages 22–25.

Gahbiche-Braham, S., Bonneau-Maynard, H., Lavergne, T., and Yvon, F. (2012). Joint Segmentation and POS Tagging for Arabic Using a CRF-based Classifier. In Chair), N. C. C., Choukri, K., Declerck, T., Dogan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proc. of LREC'12*, pages 2107–2113, Istanbul, Turkey. European Language Resources Association (ELRA).

Habash, N. (2007). Arabic Morphological Representations for Machine Translation. In *Arabic Computational Morphology*, Text, Speech and Language Technology, pages 263–285. Springer Netherlands.

Habash, N., Rambow, O., and Roth, R. MADA+TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, pages 102–109. elda.org.

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

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*, pages 29–32. Association for Computational Linguistics.

Hulden, M. and Samih, Y. (2012). Conversion of Procedural Morphologies to Finite-State Morphologies: a Case Study of Arabic. In *10th International Workshop on Finite State Methods and Natural Language Processing*, page 70. Citeseer.

Jaafar, Y. and Bouzoubaa, K. (2014). Benchmark of Arabic morphological analyzers challenges and solutions. In *2014 9th International Conference on Intelligent Systems: Theories and Applications (SITA-14)*, pages 1–6. IEEE.

Kammoun, N., Belguith, L., and Hamadou, A. (2010). The MORPH2 new version: A robust morphological analyzer for Arabic texts. In Bolasco, S., Chiari, I., and Giuliano, L., editors, *JADT 2010: 10th International Conference on Statistical Analysis of Textual Data*, pages 1033–1044.

Khoja, S. (2001). APT: Arabic part-of-speech tagger. In *Proceedings of the Student Workshop at NAACL*, pages 20–25.

Kim, Y.-B., Snyder, B., and Sarikaya, R. (2015). Part-of-speech Taggers for Low-resource Languages using CCA Features. In *Empirical Methods in Natural Language Processing (EMNLP)*, number September, pages 1292–1302. ACL – Association for Computational Linguistics.

Maamouri, M., Graff, D., Bouziri, B., Krouna, S., Bies, A., and Kulick, S. (2010). Standard Arabic morphological analyzer (SAMA) version 3.1. *Linguistic Data Consortium, Catalog No.: LDC2010L01*.

Maegaard, B. (2004). NEMLAR-An Arabic Language Resources Project. *LREC*, pages 109–112.

Mohamed, E., Kübler, S., Sandra, K., and Hall, M. (2010). Is Arabic Part of Speech Tagging Feasible Without Word Segmentation? In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 705–708.

Mueller, T., Schmid, H., and Schütze, H. (2013). Efficient Higher-Order CRFs for Morphological Tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, number October, pages 322–332, Seattle, Washington, USA. Association for Computational Linguistics.

Paroubek, P. (2007). Evaluating Part-of-Speech Tagging and Parsing. *Evaluation of Text and Speech Systems*.

Pasha, A., Al-Badrashiny, M., Diab, M., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., and Roth, R. M. (2014). Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC), Reykjavik, Iceland*.

S. Rabiee, H. (2011). Adapting Standard Open-Source Resources To Tagging A Morphologically Rich Language: A Case Study With Arabic. In *Proceedings of the Second Student Research Workshop associated with RANLP 2011*, number September, pages 127–132, Hissar, Bulgaria. RANLP 2011 Organising Committee.

Sawalha, M. (2011). *Open-source resources and standards for Arabic word structure analysis: Fine grained morphological analysis of Arabic text corpora*. Phd thesis, University of Leeds.

Sawalha, M., Atwell, E., and Abushariah, M. a. M. (2013). SALMA: Standard arabic language morphological analysis. In *2013 1st International Conference on Communications, Signal Processing and Their Applications, ICCSPA 2013*.

Smrz, O. (2007). *Functional Arabic Morphology. Formal System and Implementation*. PhD thesis, Charles University in Prague.

Toutanova, K., Klein, D., and Manning, C. D. (2003). Feature-rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL '03),*, volume 1, pages 252–259.

Zhang, Y., Li, C., Barzilay, R., and Darwish, K. (2015). Randomized Greedy Inference for Joint Segmentation, POS Tagging and Dependency Parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 42–52.