Volume 32 — Number 1 — 2017 — ISSN 2190-6858

JLCL
Journal for Language Technology
and Computational Linguistics

NLP for Perso-Arabic Alphabets

Herausgegeben von Edited by
Adrien Barbaresi, Lothar Lemnitzer, Kais Haddar

GSCL Gesellschaft für Sprachtechnologie & Computerlinguistik

Contents

Impressum

Herausgeber	Gesellschaft für Sprachtechnologie und Computerlinguistik (GSCL)
Aktuelle Ausgabe	Band 32 – 2017 – Heft 1
Gastherausgeber	Adrien Barbaresi, Lothar Lemnitzer, Kais Haddar
Anschrift der Redaktion	Adrien Barbaresi Berlin-Brandenburgische Akademie der Wissenschaften Jägerstr. 22/23 10117 Berlin barbaresi@bbaw.de
ISSN	2190-6858
Erscheinungsweise	2 Hefte im Jahr, Publikation nur elektronisch
Online-Präsenz	www.jlcl.org

In order to widen the scope of the JLCL, it was decided to publish a special issue of the journal with the title NLP für Perso-Arabic alphabets. After an open call for papers, we received eight contributions. After the usual reviewing process (at least two reviews per contribution) we could accept three papers for publication. As a result, this issue mostly deals with topics related to natural processing and automatic annotation (articles 1 and 2), while contribution 3 presents an empirical study on a particular type of phrase structure. The contributions focus on Arabic and its variants, a field that still provides a number of challenges for computational methods and tools.

We would like to thank the GSCL-board for their support. We also would like to thank the reviewers who took the time to help us meet the quality standards of the journal.

Last but not least, we wish our readers a pleasant and informative lecture.

The editors, Adrien Barbaresi, Lothar Lemnitzer and Kais Haddar.

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[o]	-	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. [o] 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 يُؤْمِنُ أَحَدُكُمْ حَتَّى يَكُونَ هَوَاهُ تَبَعًا لِمَا جِئْتُ بِهِ *lā yuʷminu 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. ° 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 ثنا *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 أَيُّهَا *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 جِئتُ *ǧiʾtu*	Voweled Word جِئتُ *ǧiʾtu*	Prefix #	Stem جء *ǧʾt*	Type فعل ماض مبني للعلوم *fʿl māḍ mbny llmʿlwm*	Pattern فلت *filtu*	Root جيء *ǧyʾ*	POS Tags ثلاثي مجرد مسند إلى المتكلم أنا ولازم متعد *tlāty mǧrd msnd ʾlā ālmtklm ʾnā mtʿd wlāzm*	Suffix تاء المتكلم *tāʾ ālmtklm*
ji}otu	ji}otu	#	j}t	Active perfect verb	1i2o3u	jyʾ	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.

Osama Hamed and Torsten Zesch

A Survey and Comparative Study of Arabic Diacritization Tools

Abstract

Modern Standard Arabic, as well as other languages based on the Arabic script, are usually written without diacritics, which complicates many language processing tasks. Although many different approaches for automatic diacritization of Arabic have been proposed, it is still unclear what performance level can be expected in a practical setting. For that purpose, we first survey the Arabic diacritization tools in the literature and group the results by the corpus used for testing. We then conduct a comparative study between the available tools for diacritization (Farasa and Madamira) as well as two baselines. We evaluate the error rates for these systems using a set of publicly available, fully-diacritized corpora in two different evaluation modes. With the help of human annotators, we conduct an additional experiment examining error categories. We find that Farasa is outperforming Madamira and the baselines in both modes.

1 Introduction

Automatic diacritization is the task of restoring missing diacritics in languages that are usually written without diacritics like Arabic; or in languages that have diacritically marked characters in their orthography like Dutch, German, Hungarian, Lithuanian, or Slovene (Acs and Halmi, 2016). The challenge is that many words have different meanings depending on their diacritization, which can only be resolved by the context and proper knowledge of the grammar (Rashwan et al., 2011).

Restoring diacritics is an important task, as diacritized texts are crucial for many Natural Language Processing (NLP) applications, including automatic speech recognition (Zitouni et al., 2006; Ananthakrishnan et al., 2005), statistical machine translation (Diab et al., 2007a), text-to-speech (Shaalan et al., 2009), text analysis, information retrieval (Azmi and Almajed, 2015), and the normalization and analysis of social media texts (Čibej et al., 2016). Diacritized text is also important at the early stages of language learning and for second language (L2) learners.

Although there is a large body of research on the topic, only very few tools are freely available, and it is still unclear what performance level can be expected in a practical setting. We aim at a fully reproducible comparison and will thus only include tools that are freely available and can be integrated into our comparison pipeline. To the best of our knowledge, there are currently only two tools that fulfill these requirements:

Madamira (Pasha et al., 2014) and *Farasa* (Darwish and Mubarak, 2016). There exist some additional tools like Mishkal[1], which is only available as a web service, and ArabicDiacritizer[2], which only works in a Windows environment. Additionally, both tools limit the size of the input text and cannot be easily integrated in our Java-based comparison framework. For the same reasons of ensuring reproducibility, we only use training and test data that is publicly available without license fees.

In this paper, we conduct a comparative study between the available tools for diacritization using a reasonable amount and variety of test data in two evaluation modes: strict and relaxed. While the strict mode expects the diacritics to be exactly the same as in gold standard text, the relaxed mode normalizes the texts (output and gold standard) to hold a specific (smaller) ratio of diacritics. Thus, the relaxed mode does not punish a tool that only provides partial diacritization. In order to put the results into perspective, we implement two strong baselines: a dictionary lookup system and one based on character-based sequence labeling. The first baseline labels each word using the diacritized form that appears most often in the training set. The second baseline treats diacritization as a sequence classification problem using conditional random fields (CRF). We report the error rates for the baselines and state-of-the-art systems using diacritized text from Classical Arabic (Quran and Tashkeela corpora) and contemporary writing (RDI corpus) in both evaluation modes.

2 Linguistic Background

Languages based on the Arabic script usually represent only consonants in their writing and do not mark the short vowels (Belinkov and Glass, 2015). The Arabic script (الخط العربي) is written from right to left and contains two classes of symbols for writing words: letters and diacritics (Habash and Rambow, 2007; Habash, 2010). Figure 1 shows the non-diacritized and the diacritized versions of the sentence "*The Arabic script*".

without diacritics	الخط العربي
with diacritics	اَلْخَطّ اَلْعَرَبِيُّ

Figure 1: Example of an Arabic sentence without and with diacritics (eng: *The Arabic script*).

[1] http://tahadz.com/mishkal
[2] https://sourceforge.net/projects/arabicdiacritizer/

Letters The Arabic alphabet has 29 letters, those include three long vowels (Alif (ا), Waw (و) and Yeh (ي)), 25 consonants, and the Hamza (glottal stop).

Diacritics The diacritics are optional. If present they appear as small strokes that are placed above or below the letter, such as ـَ ـُ ـِ.

Diab et al. (2007a) group these diacritical marks into three categories: vowel, nunation, and Shadda (gemination). The vowel diacritics refer to the three short vowels (Fatha (ـَ) /a/[3], Damma (ـُ) /u/, and Kasra (ـِ) /i/) and a diacritic indicating the absence of any vowel (Sukun) (Bouamor et al., 2015). Nunation diacritics indicate a short vowel followed by a non-written sound of the Arabic letter (ن) /n/. The Nunation diacritics look like a doubled version of their corresponding short vowels (Habash, 2010). They are named in Arabic as such: Fathatan, Dammatan, Kasratan.[4] For example, (دٌ) is pronounced /dun/ and transliterated as "duN".[5] The gemination mark (Shadda) is a consonant-doubling diacritical (دّ) "d∼". Shadda can be combined with diacritics from the other two categories, which results in a total of thirteen diacritical marks. For instance, (دُّ) "d∼u" and (دٌّ) "d∼uN".

There are general rules for diacritizating Arabic text. For example, Shaddah and Sukun cannot follow a word-initial letter, whereas Tanween appears only at word-final position (Elshafei et al., 2006). Table 1 exemplifies the shapes of diacritics in conjunction with the Arabic letter (د) /d/.

Some of the diacritics vary depending on syntactic conditions (case-related), and some vary to indicate semantic differences. Functionally, diacritics fall into two types: lexical and inflectional diacritics (Diab et al., 2007a). The lexical diacritics distinguish between two lexemes; for example, "kAtib" (كَاتِب), meaning "writer," and "kAtab" (كَاتَب), meaning "to correspond". The inflectional diacritics distinguish different inflected forms of the same lexeme. For example, the final (last-letter) diacritic in "kitaAbu" (كِتَابُ), meaning "book," is *Damma* to indicate the nominative case (verb subject) and the final diacritic in "kitaAba" (كِتَابَ) is *Fatha* to indicate the accusative case (verb object) of the same word.

In unicode, the diacritics are presented as additional characters, so the diacritized word is longer than the non-diacritized word. For example, the diacritized word "Eal∼ama" (عَلَّمَ) has seven unicode characters, whereas the bare form "Elm" (علم) has only three.

[3] International Phonetic Alphabet (IPA)
[4] Dual feminine nouns that indicate two Fathas, two Dammas and two Kasras respectively.
[5] Buckwalter encoding (Buckwalter, 2004) is used exclusively in the paper.

Type	Diacritic Mark	Name	Transl.	IPA	Word Position
Short vowels	ـَ	Fatha	a	/a/	Any
	ـُ	Damma	u	/u/	Any
	ـِ	Kasra	i	/i/	Any
	ـْ	Sukun	o	Ø	Any
Nunation	ـًا	Tanween Fath	F	/an/	End
	ـٌ	Tanween Damm	N	/un/	End
	ـٍ	Tanween Kasr	K	/in/	End
Gemination	ـّ	Shadda	∼	:	Any

Table 1: Types of Arabic diacritics

2.1 Diacritization Levels

The level of diacritics refers to the number of diacritical marks presented on a word to avoid text ambiguity for human readers. Even in non-diacritized newswire text, 1.6% of all words have at least one diacritic indicated by their author to guide the reader with disambiguation (Habash, 2010). Ahmed and Elaraby (2000) grouped the diacritization levels into three levels (full, half, and partial):

Full All the letters are given appropriate diacritics. This applies to classical Arabic (CA), as in religion-related books, and at early stages of language learning, such as in children's books.

Half Only the morphological-independent letters are diacritized. In other words, all the letters of a word, except those that depend on the syntactic analysis of the word, are diacritized. For example, the word "wldh" (ولده), meaning "his son" consists of two clitics "wld+h" = (ه) + (ولد), i.e. the stem "wld" and the possessive pronoun "h" as suffix. With the half diacritization, it would be written like (وَلَده) instead of (وَلَدُهُ). This means that the diacritic was dropped from the pronoun "h" (ه) (morphology-dependent) and from the stem last letter "d" (د) (syntactic).

Partial Any other setting where one letter or a subset of letters is diacritized. While studying the impact of diacritization on statistical machine translation, Diab et al. (2007a) proposed to divide this level into four sub-levels for use with inflectional and lexical diacritics. A special case of partial diacritization is to drop the short

Type	Bare Form	Diacritized	Gloss / Transliteration
POS	علم	عِلم	Science / Eilom
		عَلَم	Flag / Ealam
		عَلِمَ	He knew / Ealima
		عُلِم	It was known / Eulim
		عَلَّمَ	He taught / Eal∼ama
Syntactic	مدير البنك الجديد	مُديرَ البَنكِ الجَديدِ	the manager of the new bank / mudyra Albanki Aljadydi
		مُديرَ البَنكِ الجَديدَ	the new bank manager / mudyra Albanki Aljadyda
Structure	ولي	وَلِي	and for me / waliy
		وَلِيّ	a pious person favored by God / waliy∼

Table 2: Types of ambiguity caused by missing diacritics

vowels and Sukun. For example, the short vowel is dropped from the letter that precedes a long vowel with similar sound like when *Fatha* is dropped from a letter if followed by an *Alef* (ا). Additionally, the Arabic definite article ال has only two diacritization possibilities depending on the preceding letter. The *Alef* is always diacritized with *Fatha*, and the *Lam* (ل) either has Sukun or has no diacritics.

2.2 Ambiguity

Writing Arabic without diacritics introduces three types of ambiguity (Azmi and Almajed, 2015). The first is part-of-speech (POS) tagging ambiguity (Maamouri et al., 2006). This is the case with the words that have the same spelling and POS tag but a different lexical sense, or words that have the same spelling but different POS tags and lexical senses (homograph ambiguity) (Farghaly and Shaalan, 2009). Second, there is ambiguity on the grammatical level (syntactic ambiguity). Sentences and phrases can be interpreted in more than one way, and diacritics are the only means to resolve ambiguity (Maamouri et al., 2006). The third is internal word structure ambiguity, such as when Arabic words are segmented in different ways. The agglutination property of Arabic might produce a problem that can only be resolved using diacritics. Table 2 summarizes the aforementioned types of ambiguity with excerpted examples from (Metwally et al., 2016; Farghaly and Shaalan, 2009).

Corpus	Description	Availability	# of tokens
Quran	Religious	Free	78 000
RDI	Religious/Modern	Free	20 000 000
Tashkeela	Religious	Free	60 000 000
ATB	News	Commercial	1 000 000
WikiNews	News	Free	18 300

Table 3: Overview of diacritized corpora.

3 State-of-the-Art Arabic Diacritization

In this section, we present the diacritized datasets usually used for evaluation and then give an overview of the results on different corpora that have so far been obtained using the standard evaluation metrics.

3.1 Datasets

Generally, the currently available diacritized corpora are limited to classical texts (usually religious or Arabic poetry), such as the Holy Quran, RDI, and Tashkeela on the one side, and newswire corpora, such as the Arabic Penn Treebank (ATB) from the Linguistic Data Consortium (LDC) on the other side, as shown in Table 3.

Quran The small diacritized Quranic corpus is part of Tanzil[6] project. It contains more than 78 thousand tokens that comes in a UTF-8 encoded text file. The file has no Arabic punctuation marks, and every Quranic verse appears in a separate line.

RDI The corpus was collected by the RDI[7] company for use in the field of automatic diacritization. It is composed of diacritized texts, which are mainly gathered from classical Arabic books with a small percentage from contemporary Arabic writing (modern books). Overall, it contains 20 million tokens. Our experiments are based on the subset of modern books, a collection of 12 books from the late 1990's.

Tashkeela The corpus contains more than 60 million diacritized tokens (Zerrouki and Balla, 2017). It is a collection of 84 Islamic religious heritage books. The books are provided in HTML format, encoded in CP1256 Windows Arabic. It can be downloaded under GPL license.[8]

[6] http://tanzil.net/download/
[7] http://www.rdi-eg.com/RDI/TrainingData/
[8] https://sourceforge.net/projects/tashkeela/

ATB Much of the previous work on diacritization relied on using the ATB. LDC's Arabic Penn Tree Bank (ATB) consists of distinct newswire stories collected from different news agencies and newspapers, including the Agence France-Presse (AFP), Al-Hayat, and An-Nahar newspapers (Maamouri et al., 2004, 2006, 2009). It contains about 1 million tokens. Though ATB is invaluable for many tasks, such as POS tagging and parsing, it is sub-optimal for diacritization (Darwish et al., 2017).

WikiNews Darwish et al. (2017) used a new test set composed of 70 WikiNews articles (the majority are from 2013 and 2014) that cover a variety of themes, namely: politics, economics, health, science and technology, sports, arts, and culture. The articles are evenly distributed among the different themes (10 per theme). The corpus contains 18,300 words.

3.2 Evaluation Metrics

In the literature, two standard evaluation metrics are used almost exclusively to measure systems performance (Rashwan et al., 2011; Said et al., 2013). It can either be expressed in terms of error rates on the character or on the word level. The smaller the error rate, the better the performance.

DER Diacritization Error Rate (DER) is the proportion of letters which are incorrectly labeled with diacritics. The following assumptions are made: (i) each letter or digit in a word is a potential host for a set of diacritics, and (ii) all diacritics on a single letter are counted as a single binary choice. The DER can be calculated as follows:

$$DER = (1 - \frac{|T_S|}{|T_G|}) \cdot 100 \quad (1)$$

where $|T_S|$ is the number of letters assigned correctly by the system, and T_G is the number of diacritized letters in the gold standard text.

WER Word Error Rate (WER) is the percentage of incorrectly diacritized white-space delimited words. In order to be counted as incorrect, at least one letter in a word must have a diacritization error. All words are counted, including numbers and punctuation.

While the diacritization techniques work relatively well on lexical diacritics (located on word stems), they are much less effective for inflectional diacritics (typically at

Diacritization Tool	Word Letters					
	A	**l**	**E**	**r**	**b**	**y**
Gold	a	o	a	a	i	∼u
Tool 1	-	o	a	a	i	∼u
Tool 2	-	o	a	a	-	∼u
Tool 3	-	-	-	a	-	∼u
Tool 4	-	-	a	a	-	∼u
in relaxed evaluation?	No	No	No	Yes	No	Yes

Table 4: The normalization of diacritics for comparison in relaxed evaluation mode.

word-final position) (Habash et al., 2007). In most cases, the last letter indicates the case ending. However, in some cases as with plural masculine nouns (جمع المذكر السالم) and dual masculine and feminine nouns (المثنى) the suffixes substitute the diacritics. The suffixes are added to the word to indicate case and number. For example, the suffixes (ون) or (ان) are added to the word to indicate plural masculines and dual masculine or feminine nouns in accusative case respectively. However, the suffix (ين) is added to the word to indicate plural masculines and dual masculine or feminine nouns in nominative and dative cases. Assigning the correct case can often only be decided using a wider context, thus diacritization tools usually perform worse on the last letter compared to the other positions in the word (Habash et al., 2007). It is thus usual to also report a variant of the above two mentioned metrics that ignore the last letter (assumed to have no syntactic diacritics), denoted as **DER-1** and **WER-1**.

3.3 Evaluation Modes

When comparing multiple tools, we distinguish two different evaluation modes:

Strict Mode Whenever a letter has a set of diacritics in the gold standard text, a diacritization tool is expected to predict this set exactly. This evaluation mode is most often used and gives an advantage to tools providing full diacritization.

Relaxed Mode This evaluation mode gives an advantage to tools that only output diacritics when being confident about the results. This might be useful for half or partial diacritization settings, e.g. the tools that drop the default diacritics. This is not so useful for other settings, e.g. full diacritization in children books.

In order to provide a fair comparison between multiple tools, the relaxed evaluation

Test Corpus	Size (10^3)	Approach	All Diacritics		Ignore Last	
			DER	WER	DER-1	WER-1
ATB (Parts 1–3)	144	(Nelken and Shieber, 2005)	12.8	23.6	6.5	7.3
	52	(Zitouni et al., 2006)	5.5	18.0	2.5	7.9
	52	(Habash and Rambow, 2007)	4.8	14.9	2.2	5.5
	613	(Schlippe et al., 2008)	4.3	19.9	1.7	6.8
	116	(Schlippe et al., 2008)	4.7	21.9	1.9	8.4
	16	(Alghamdi et al., 2010)	13.8	46.8	9.3	26.0
	52	(Rashwan et al., 2011)	3.8	12.5	1.2	3.1
	37	(Abandah et al., 2015)	2.7	9.1	1.4	4.3
	52	(Metwally et al., 2016)	-	13.7	-	-
Quran	1	(Elshafei et al., 2006)	4.1	-	-	-
	76	(Abandah et al., 2015)	3.0	8.7	2.0	5.8
Tashkeela	1902	(Hifny, 2012)	-	8.9	-	3.4
	272	(Abandah et al., 2015)	2.1	5.8	1.3	3.5
Tashkeela+RDI	199	(Bebah et al., 2014)	7.4	21.1	3.8	7.4
WikiNews	18	(Pasha et al., 2014)	5.4	19.0	1.9	6.7
	18	(Rashwan et al., 2015)	4.3	16.0	1.0	3.0
	18	(Belinkov and Glass, 2015)	7.9	30.5	3.9	14.9
	18	(Darwish et al., 2017)	3.5	12.8	1.1	3.3

Table 5: Performance of Arabic diacritization systems grouped by test corpus

mode only takes into account cases where all tools under consideration return a diacritic for a given letter. Table 4 gives an example.

3.4 Overview of Diacritization Results

The work on Arabic diacritization goes back quite a long time (El-Sadany and Hashish, 1989) and many different approaches have been proposed including hidden Markov model (Elshafei et al., 2006), n-gram language models (Hifny, 2012; Alghamdi et al., 2010), statistical machine translation (Schlippe et al., 2008), finite state transducers (Nelken and Shieber, 2005), maximum entropy (Zitouni et al., 2006), and deep learning (Rashwan et al., 2015; Abandah et al., 2015; Belinkov and Glass, 2015).

Additionally, many researchers have proposed to improve classification with morphological analysis (Habash and Rambow, 2005; Rashwan et al., 2011; Bebah et al., 2014; Metwally et al., 2016) and the standard n-gram language model. A recent approach by Darwish et al. (2017) employed a Viterbi decoder and SVM-rank to properly guess words diacritization.

ID	Corpus	# words (10^3)	∅ chars per word	Words / sentence
Q	Quran	78	4.25	12.6
T	Tashkeela	100	4.11	14.7
R	RDI	100	4.47	34.1

Table 6: Statistics of corpora sub-datasets used in this study.

Comparison Table 5 gives an overview of the reported results from the literature. The results are grouped by the corpus that was used for testing in order to allow for a fair comparison. There is a major drawback with these reported results: they do not follow a well-established framework for testing. For example, most numbers are still not directly comparable because they were obtained using different test sets. Moreover, some works used a fixed test set without performing any cross-validation, which further limits the weight that should be put on those numbers. The only exception to this is the last block of results, where Darwish et al. (2017) compared their system with other systems using the *WikiNews* test set. Under this controlled setting, their system outperforms all other systems regarding DER and WER. If we ignore the case-endings, the Rashwan et al. (2015) system performs best.

As most of the systems from the literature are not freely available, we have no way of directly comparing them. In this paper, we establish a comparative study that only includes the systems and corpora that are freely available in a controlled settings.

4 Experimental Setup

In this section, we present our experimental setup: used data, baselines, diacritization tools, and evaluation metrics.

The experiments were carried out using DKPro TC, the open-source UIMA-based framework for supervised text classification (Daxenberger et al., 2014). The baseline experiments were conducted as ten-fold cross-validation, reporting the average over the ten folds.

4.1 Datasets

Table 6 shows the statistics for the experimental sub-datasets (punctuation marks are not counted). All the experiments use a general setup for test sample-size: 78K, 100K, and 100K drawn from the Quran, RDI, and Tashkeela respectively.

Data Preprocessing The Quran text requires no special preprocessing. However, the files from Tashkeela and RDI contain Quranic symbols like the Dagger Alif (a small Alif quite common in Quranic Arabic (Dukes and Habash, 2010)) or English letters. In order to prepare those corpora for training and testing purposes, the following preprocessing steps are performed: (i) convert them from HTML to plain text files that have one sentence per line, (ii) clean the files by removing the Quranic symbols and words written in non-Arabic letters, and (iii) normalize the Arabic text by removing extra white spaces and Tatweel symbols.[9] For example, "qAl" (قـال), meaning "he said" has Tatweel, whereas قال has no Tatweel.

4.2 Baselines

We implemented two baselines: a simple dictionary lookup approach and a sequence labeling approach.

Dictionary Lookup This baseline labels each word with the diacritized form that appears most often in the training corpus. Words that are not found in the dictionary are not diacritized.

Sequence Labeling We treat diacritization as a sequence labeling problem and propose a baseline solution using conditional random fields (Lafferty et al., 2001). Given a sentence (set of non-diacritized words) separated using white-space delimiters, each word in the sentence is a sequence of characters, and we want to label each letter with its corresponding labels from the diacritics set $D = (d_1, ..., d_N)$. We represent each word as an input sequence $X = (x_1, ..., x_N)$ where we need to label each consonant in X with the diacritics that follow this consonant. Note that an Arabic letter has a maximum of two diacritics, and if it has two, then one of them is always Shadda. Shadda might accompany all diacritics except Sukun, so in total we have 14 labeling possibilities (including the 'no diacritic' option). Thus, in order to diacritize sequence X, we must find its labeling sequence Y (usually of word length) derived from D.
A word might have more than one valid labeling. The word "ktAb" (كتاب) represented as (k, t, A, b), can be labeled with $Y_1 = (i, a, o, u)$ or $Y_2 = (i, a, o, a)$ resulting in the diacritized words "kitaAobu" and "kitaAoba" respectively.

Our features are character n-grams language models (LMs) in sequence labeling approach. The features extractor selects the character-level features relevant to diacritics

[9] Tatweel are used to stretch words to indicate prominence or simply to force vertical justification (Habash, 2010).

from annotated corpora. It collects the diacritics on previous, current and following character and up to the 6th character.

Note that the out-of-vocabulary (OoV) rate of this approach is zero as it is able to provide a sequence of diacritics for arbitrary unknown words.

4.3 Diacritization Tools

To the best of our knowledge, the only tools that can be tested on large corpora and are easily integrated with Java frameworks are *Madamira* and *Farasa*.

Madamira Madamira (Pasha et al., 2014) improves upon its two ancestors MADA (Habash et al., 2009) and AMIRA (Diab et al., 2007b) with a Java implementation that is more robust, portable, extensible, and faster. Arabic processing with Madamira includes automatic diacritization, lemmatization, morphological analysis and disambiguation, part-of-speech tagging, stemming, glossing, tokenization, base-phrase chunking, and named-entity recognition. Madamira makes use of fast, linear SVMs implemented using *Liblinear* (Fan et al., 2008).

Madamira was trained on the training portion of ATB (parts 1, 2 and 3). There are two varieties of Madamira. The first integrates the public version of Arabic morphological analyzer (AraMorph).[10] The second integrates the Standard Arabic Morphological Analyzer (SAMA) and its recommended database (Graff et al., 2009).[11]

Our experiments are carried out using the SAMA enabled version of Madamira *v2.1*. Madamira was used to diacritize the test sequences from the three corpora. As the resulting diacritized text is encoded using Buckwalter transliteration, it is necessary to decode it into Arabic text. We compare the mapped Arabic text with a gold standard sequence and then calculate the different metrics.

Farasa Farasa (Darwish and Mubarak, 2016) is an open-source tool, written entirely in native Java. Farasa consists of a segmentation/tokenization module, POS-tagger, Arabic text diacritizer, and dependency parser. Its approach is based on SVM-ranking using linear kernels. Farasa matches or outperforms state-of-the-art Arabic segmenters (Darwish and Mubarak, 2016) and diacritizers.

Corpus	Approach	OoV	All Diacritics		Ignore Last	
		rate	DER	WER	DER-1	WER-1
Quran	Dict. Lookup	11.8	19.7	27.5	16.1	16.8
	Sequence Labeling	0.0	21.4	28.3	9.0	19.9
	Madamira	3.4	21.1	36.7	15.4	20.9
	Farasa	0.3	**12.2**	**19.0**	**8.9**	**9.5**
RDI	Dict. Lookup	13.3	26.6	31.8	19.7	22.5
	Sequence Labeling	0.0	24.9	37.0	15.4	22.4
	Madamira	2.1	17.8	28.4	13.1	14.2
	Farasa	0.1	**10.5**	**15.7**	**6.7**	**7.6**
Tashkeela	Dict. Lookup	13.4	26.9	32.2	19.9	22.7
	Sequence Labeling	0.0	24.9	37.0	15.7	22.3
	Madamira	2.2	17.9	28.6	13.1	14.2
	Farasa	0.1	**10.6**	**15.9**	**6.8**	**7.7**

Table 7: Error rates in strict evaluation mode. The "OoV" rate refers to the ratio of tokens that were not diacritized by the system.

5 Results

We now report the results of our diacritization experiments using first 'strict' and then 'relaxed' evaluation.

5.1 Strict Evaluation

Table 7 gives an overview of our evaluation results in *strict* mode. The results are grouped by the corpus that was used for testing. Note that the OoV column refers to the ratio of tokens that got "No Analysis" and thus no diacritization by the system.

In general, the error rates are rather high. With keeping in mind that the reported results are non-comparable, none of the methods (including the two well-known state-of-the-art systems) comes even close to the numbers in Table 5. It is likely that many approaches do not use strict evaluation mode when reporting results, even if it is the most comparable setup. When indirectly competing with other published results, the numbers obtained in that way are just not competitive.

Looking at individual results, Farasa outperforms all other methods under all metrics. For the remaining three approaches, there is no clear trend, but it should be noted that the baselines perform surprisingly well even if they make no real attempt at resolving ambiguity. Sequence labeling doesn't take context into account and the dictionary

[10] `http://www.nongnu.org/aramorph/`
[11] Catalog number LDC2009E73

	Diacritics per letter			Diacritized letters per word		
Approach	**Quran**	**RDI**	**Tashkeela**	**Quran**	**RDI**	**Tashkeela**
Gold	.84	.83	.83	.78	.77	.77
Dict. Lookup BL	.84	.84	.82	.78	.77	.77
Seq. Labeling	.82	.78	.78	.77	.74	.74
Madamira	.55	.59	.61	.51	.54	.56
Farasa	.58	.58	.61	.55	.54	.58

Table 8: Average number of diacritics per letter and average number of diacritized letters per word

lookup makes a majority class decision for each ambiguous token. We suspect that many tokens within a domain are not ambiguous and the repetitious nature of the religious texts increases the effect.

Table 7 also shows the out-of-vocabulary rate for each approach. As expected, the dictionary lookup baseline has a rather high rate and sequence labeling has no out-of-vocabulary tokens at all, because it always returns one of the possible diacritization patterns. For all corpora, Farasa has a lower OoV rate than Madamira.

When looking into individual OoV examples, we find that in some cases the tools do not return any analysis. However, in some cases they change the input token instead of just adding diacritics. For example in one case in Madamira, the verb "rawaAhu" (رواه), meaning "narrated by" is changed into "ruwaAp" (رواة), meaning "narrators". Another example is the passive verb "yusotavonaY" (يسْتثنى), meaning "to be excluded" that is changed into the present tense verb "yasotavoniy" (يسْتثني), meaning "excludes". In both examples, the last letter is changed into a very similar, but different form. We see a similar behavior in Farasa, where in some examples a word containing two adjacent *Lam* (ل) letters (with Shadda on the second *Lam*), where the first *Lam* is a preposition. In this case, there is an additional Alif letter introduced between the two Lam letters. For example, the word (للّه) "lil∼ah" (l + Allah) is transformed into (لاله) "liAlhi" – i.e. (l + Alh).

In Table 8, we show the average number of diacritics per letter as well for the gold standard and all systems used in our experiments. It shows that Madamira and Farasa both assign about the same amount of diacritics on average, but substantially fewer than the gold standard. This means that both tools are especially punished by the strict evaluation. These findings motivate us to repeat the evaluation using the *relaxed mode*.

Corpus	Approach	All Diacritics		Ignore Last	
		DER	WER	DER-1	WER-1
Quran	Dict. Lookup	**7.3**	24.0	**3.2**	15.6
	Seq. Labeling	15.1	22.0	7.6	13.5
	Madamira	14.5	26.4	10.2	15.6
	Farasa	7.8	**14.0**	5.0	**6.8**
RDI	Dict. Lookup	10.1	27.9	**3.4**	16.7
	Seq. Labeling	16.7	28.0	12.0	13.6
	Madamira	12.5	20.4	8.6	10.2
	Farasa	**8.3**	**13.8**	5.0	**5.1**
Tashkeela	Dict. Lookup	10.1	28.1	**3.3**	16.7
	Seq. Labeling	24.0	35.4	15.0	22.0
	Madamira	12.4	20.3	8.5	10.1
	Farasa	**8.3**	**13.9**	5.0	**5.1**

Table 9: Error rates in relaxed evaluation mode

5.2 Relaxed Evaluation

Table 9 shows the results in relaxed mode, where we only take into account cases where all tools under consideration return a diacritic for a given letter. As expected, the error rates drop substantially, but not evenly for all approaches. In order to better show the improvement (decrease in error rates) obtained by switching from strict to relaxed evaluation mode, we report the relative change between both modes in Table 10. It can be clearly seen that this switching improves the tools performance in general. Sometimes, a tool is making a dramatical change, such as the dictionary lookup baseline under the DER and DER-1 metrics.

Looking again at the error rates in Table 9, relaxed evaluation mode reveals that Farasa is still performing better than Madamira in all cases, but for the DER and DER-1 metrics the dictionary lookup baseline is close or even better. The big difference between DER and WER performance for the dictionary lookup approach is most likely explained by errors in the inflectional diacritics that are impossible to resolve without looking at the context. However, that such a simple approach performs so well is surprising and shows that there is still a lot room for improvement in the area of automatic diacritization.

Corpus	Approach	All Diacritics		Ignore Last	
		DER	WER	DER-1	WER-1
Quran	Dict. Lookup	63	13	80	7
	Seq. Labeling	29	22	16	32
	Madamira	31	28	34	25
	Farasa	36	26	44	28
RDI	Dict. Lookup	62	12	83	26
	Seq. Labeling	33	24	22	39
	Madamira	30	28	34	28
	Farasa	21	12	25	33
Tashkeela	Dict. Lookup	62	13	83	26
	Seq. Labeling	4	4	4	1
	Madamira	31	29	35	29
	Farasa	22	13	26	34

Table 10: The relative change (in %) between the strict and relaxed evaluation modes

6 Qualitative Analysis

As most of the systems from the literature are not freely available, we have no way of directly comparing our results with those approaches unless they have the same settings. There is still a gap between our experimental results in relaxed mode and some of the reported published results in Table 5. Part of the gap can certainly be attributed to differences in the corpora. To see how the systems are performing, we also conducted a small diacritization experiment that only involves the best baseline (dictionary lookup), Madamira, and Farasa. We conduct a simple experiment using a blind MSA test set, a sample with 94 non-diacritized words (crawled from the internet). It was then diacritized using dictionary lookup (which was trained with RDI), Madamira (SAMA-enabled), and Farasa. We gave the resulting diacritized text to two Arabic teachers with appropriate experience to conduct the evaluation.

To look at the kinds of errors we were getting, the annotators were asked to identify the incorrectly diacritized words using word error rates (WERs) metrics because it is easy to manage for the volunteer teachers. Additionally, they were asked to state the reason if a diacritization produced by Madamira or Farasa was incorrect. For that purpose, we are using a error classification scheme developed for Arabic learner corpora (Abuhakema et al., 2008).

Error Category	Error Subcategoy	Annotator 1		Annotator 2	
		Madamira	Farasa	Madamira	Farasa
Form/Spelling	Shadda	2	3	2	3
	Tanween	6	6	6	6
Morphology	Partial-Inflection	1	1	0	1
	Full-Inflection	2	0	2	0
Grammar	Active-Passive Voice	2	2	2	2
Diacritization	Missing Short Vowel	6	0	5	0
	Confused Short Vowel	1	5	1	4
Overall		20	17	18	16

Table 11: The annotated WERs subcategories.

Form/Spelling Errors caused by Shadda (consonant doubling), or Tanween (nunation).

Morphology Correct lexical item, but wrong case ending, e.g. Kasra instead of Fatha.

Grammar Errors caused by changes in grammatical role, e.g. active or passive voice (المبني للمعلوم و المجهول).

Diacritization Errors caused by incorrect, missing or redundant short vowels (i.e. lexical diacritics).

Table 11 shows the distribution of error categories as reported by the annotators. The inter-evaluator agreement for the annotated WER (using Cohen's kappa) is almost perfect with values of .93 and .96 for Madamira and Farasa respectively. The majority of the mistakes are due to form/spelling and diacritization errors. In the form/spelling category, both tools make a lot of Tanween errors. This is to be expected, as it has been reported that the diacritization tools work relatively well on lexical diacritics, but that they are much less effective for case-ending diacritics (Habash et al., 2007). In the 'Diacritization' category, we observe a quite different behavior. Madamira has more missing vowels, i.e. it seems to rather not return a diacritic than to get it wrong. Farasa is on the opposite side of the trade-off with no missing short vowels, but almost as many confused short vowels.

7 Conclusion

The performance numbers reported in the literature on automatic diacritization are inconclusive, as the experimental settings are not comparable in most cases. In this

paper we establish a framework to compare the state-of-the-art publicly available Arabic diacritizers. The test data was drawn from the Quran, Tashkeela, and RDI corpora. Under controlled settings, we compared two strong baselines and two well-known systems: Madamira and Farasa. The error rates are reported in strict and relaxed evaluation modes to ensure fair comparison. We find that Farasa is outperforming Madamira in both evaluation modes, but that in relaxed mode the simple dictionary lookup baseline is surprisingly strong. In general, our error rates are much higher than the ones reported in the literature and we currently have no satisfying explanation for the difference. We are making our evaluation framework publicly available in order to foster additional research in this area and to allow for more approaches to be tested under reproducible conditions.

References

Abandah, G., Graves, A., Al-Shagoor, B., Arabiyat, A., Jamour, F., and Al-Taee, M. (2015). Automatic diacritization of arabic text using recurrent neural networks. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(2):183–197.

Abuhakema, G., Faraj, R., Feldman, A., and Fitzpatrick, E. (2008). Annotating an arabic learner corpus for error. In *LREC*.

Acs, J. and Halmi, J. (2016). Hunaccent: Small footprint diacritic restoration for social media. In *Normalisation and Analysis of Social Media Texts (NormSoMe) Workshop Programme*, page 1.

Ahmed, A. and Elaraby, M. (2000). *A large-scale computational processor of the arabic morphology, and applications*. PhD thesis, Faculty of Engineering, Cairo University Giza, Egypt.

Alghamdi, M., Muzaffar, Z., and Alhakami, H. (2010). Automatic restoration of arabic diacritics: a simple, purely statistical approach. *Arabian Journal for Science and Engineering*, 35(2):125.

Ananthakrishnan, S., Narayanan, S., and Bangalore, S. (2005). Automatic diacritization of arabic transcripts for automatic speech recognition. In *Proceedings of the 4th International Conference on Natural Language Processing*, pages 47–54.

Azmi, A. and Almajed, R. (2015). A survey of automatic arabic diacritization techniques. *Natural Language Engineering*, 21(03):477–495.

Bebah, M., Amine, C., Azzeddine, M., and Abdelhak, L. (2014). Hybrid approaches for automatic vowelization of arabic texts. *arXiv preprint arXiv:1410.2646*.

Belinkov, Y. and Glass, J. (2015). Arabic diacritization with recurrent neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2281–2285.

Bouamor, H., Zaghouani, W., Diab, M., Obeid, O., Oflazer, K., Ghoneim, M., and Hawwari, A. (2015). A pilot study on arabic multi-genre corpus diacritization annotation. In *ANLP Workshop 2015*, page 80.

Buckwalter, T. (2004). Buckwalter arabic morphological analyzer version 2.0. linguistic data consortium, university of pennsylvania, 2002. ldc cat alog no.: Ldc2004l02. Technical report, ISBN 1-58563-324-0.

Čibej, J., Fišer, D., and Erjavec, T. (2016). Normalisation, tokenisation and sentence segmentation of slovene tweets. In *Normalisation and Analysis of Social Media Texts (NormSoMe) Workshop Programme*, page 5.

Darwish, K. and Mubarak, H. (2016). Farasa: A new fast and accurate arabic word segmenter. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Darwish, K., Mubarak, H., and Abdelali, A. (2017). Arabic diacritization: Stats, rules, and hacks. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 9–17.

Daxenberger, J., Ferschke, O., Gurevych, I., Zesch, T., et al. (2014). DKPro TC: A Java-based Framework for Supervised Learning Experiments on Textual Data. In *ACL (System Demonstrations)*, pages 61–66.

Diab, M., Ghoneim, M., and Habash, N. (2007a). Arabic diacritization in the context of statistical machine translation. In *Proceedings of MT-Summit*.

Diab, M., Hacioglu, K., and Jurafsky, D. (2007b). Automated methods for processing arabic text: From tokenization to base phrase chunking. *Arabic Computational Morphology: Knowledge-based and Empirical Methods. Kluwer/Springer*.

Dukes, K. and Habash, N. (2010). Morphological annotation of quranic arabic. In *LREC*.

El-Sadany, T. and Hashish, M. (1989). An arabic morphological system. *IBM Systems Journal*, 28(4):600–612.

Elshafei, M., Al-Muhtaseb, H., and Alghamdi, M. (2006). Statistical methods for automatic diacritization of arabic text. In *The Saudi 18th National Computer Conference. Riyadh*, volume 18, pages 301–306.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874.

Farghaly, A. and Shaalan, K. (2009). Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(4):14.

Graff, D., Maamouri, M., Bouziri, B., Krouna, S., Kulick, S., and Buckwalter, T. (2009). Standard arabic morphological analyzer (sama) version 3.1. *Linguistic Data Consortium LDC2009E73*.

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

Habash, N., Gabbard, R., Rambow, O., Kulick, S., and Marcus, M. P. (2007). Determining case in arabic: Learning complex linguistic behavior requires complex linguistic features. In *EMNLP-CoNLL*, pages 1084–1092.

Habash, N. and Rambow, O. (2005). Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 573–580. Association for Computational Linguistics.

Habash, N. and Rambow, O. (2007). Arabic diacritization through full morphological tagging. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 53–56. Association for Computational Linguistics.

Habash, N., Rambow, O., and Roth, R. (2009). MADA + TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In *Proceedings of the 2nd international conference on Arabic language resources and tools (MEDAR), Cairo, Egypt*, pages 102–109.

Hifny, Y. (2012). Smoothing techniques for arabic diacritics restoration. In *12th Conference on Language Engineering*, pages 6–12.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

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.

Maamouri, M., Bies, A., and Kulick, S. (2006). Diacritization: A challenge to arabic treebank annotation and parsing. In *Proceedings of the Conference of the Machine Translation SIG of the British Computer Society*. Citeseer.

Maamouri, M., Bies, A., and Kulick, S. (2009). Creating a methodology for large-scale correction of treebank annotation: The case of the arabic treebank. In *Proceedings of MEDAR International Conference on Arabic Language Resources and Tools, Cairo, Egypt*.

Metwally, A. S., Rashwan, M. A., and Atiya, A. F. (2016). A multi-layered approach for arabic text diacritization. In *Cloud Computing and Big Data Analysis (ICCCBDA), 2016 IEEE International Conference on*, pages 389–393. IEEE.

Nelken, R. and Shieber, S. (2005). Arabic diacritization using weighted finite-state transducers. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 79–86. Association for Computational Linguistics.

Pasha, A., Al-Badrashiny, M., Diab, M., 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*, pages 1094–1101.

Rashwan, M., Al-Badrashiny, M., Attia, M., Abdou, S., and Rafea, A. (2011). A stochastic arabic diacritizer based on a hybrid of factorized and unfactorized textual features. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(1):166–175.

Rashwan, M. A., Al Sallab, A. A., Raafat, H. M., and Rafea, A. (2015). Deep learning framework with confused sub-set resolution architecture for automatic arabic diacritization. *IEEE Transactions on Audio, Speech, and Language Processing*, 23(3):505–516.

Said, A., El-Sharqwi, M., Chalabi, A., and Kamal, E. (2013). A hybrid approach for arabic diacritization. In *International Conference on Application of Natural Language to Information Systems*, pages 53–64. Springer.

Schlippe, T., Nguyen, T., and Vogel, S. (2008). Diacritization as a machine translation problem and as a sequence labeling problem. In *8th AMTA conference, Hawaii*, pages 21–25.

Shaalan, K., Abo Bakr, H., and Ziedan, I. (2009). A hybrid approach for building arabic diacritizer. In *Proceedings of the EACL 2009 workshop on computational approaches to semitic languages*, pages 27–35. Association for Computational Linguistics.

Zerrouki, T. and Balla, A. (2017). Tashkeela: Novel corpus of Arabic vocalized texts, data for auto-diacritization systems. *Data in Brief*, 11:147–151.

Zitouni, I., Sorensen, J., and Sarikaya, R. (2006). Maximum entropy based restoration of arabic diacritics. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 577–584. Association for Computational Linguistics.

Zainab Al-Zaghir

Relativisation across varieties: A corpus analysis of Arabic texts

Abstract

Relative clauses are among the main structures that are used frequently in written texts and everyday conversations. Different studies have been conducted to investigate how relative clauses are used and distributed in corpora. Some studies support the claim that accessibility to relativisation, represented by the Noun Phrase Accessibility Hierarchy (NPAH) which is proposed by KEENAN and COMRIE (1977), predict the distribution of relative clauses in corpora. Other studies found out that discourse functions of relative clauses have an important role in distributing relative clauses in corpora (FOX, 1987). However, little focus has been given to the role of the variety in which relative clauses are written in the distribution of relative clauses in written texts. This study investigates relativisation in Arabic written texts in three varieties: Classical Arabic, Modern Standard Arabic and Iraqi Arabic. A statistical analysis of the results shows that relativisation patterns differ significantly across varieties of the Arabic language and cannot be predicted by one accessibility hierarchy.

1 Introduction

Different studies have been conducted to investigate how relative clauses are used in written and spoken corpora. One of the significant cross-linguistic studies that investigate relativisation in different languages is KEENAN and COMRIE (1977), described as "one of the most influential works in the language universals literature" (Fox, 1987, p. 856). Based on the data of around fifty languages, KEENAN and COMRIE (1977, 1979) state that some grammatical positions are more accessible to relativisation than others, and that accessibility to relativisation follows an implicational hierarchy, the Noun Phrase Accessibility Hierarchy (NPAH[1]), which is as follows:

SU > DO> IO> OBL> GEN> OCOMP, where (>) indicates more accessible.

The following English examples are provided to demonstrate the grammatical positions of the NPAH:

SU (subject relative clauses), e.g. the man who bought the book...
DO (direct object relative clause), e.g. the man whom I met...
IO (indirect object relative clause), e.g. the man whom I gave the book to...
OBL (oblique relative clause), e.g. the house which I live in...
GEN (genitive relative clause), e.g. the man whose car is red...
OCOMP (object of comparison), e.g. the man whom I am taller than...

KEENAN and COMRIE presented the NPAH first in 1972, and then, in 1977, they presented the NPAH with a full account of methodological problems and counter examples. According to the NPAH, if a language can relativise only one grammatical position, then that position must be the SU position because it is the most accessible position on the NPAH.

Moving down the NPAH, the difficulty of relativisation increases: IO is more difficult to relativise than DO, and OBL is more difficult to relativise than IO and so on. The OCOMP is the most difficult position to relativise. For example, according to the NPAH, to identify a man, a speaker would rather use SU relative clause "the man who bought the book" than IO relative clause "the man whom I gave the book to" or the OCOMP relative clause "the man whom I am taller than".

KEENAN (1975) conducted a study to verify the validity of the NPAH by analysing data from written English texts. Two major findings were suggested by that study: First, the NPAH, which is founded on a cross-linguistic basis, "can determine performance constraints within languages" (KEENAN, 1975, p. 147). Second, the SU position is used more frequently in written texts because they are psychologically more accessible to relativisation than the other grammatical positions on the NPAH; accordingly, SU relative clauses are more accessible to comprehension, acquisition and production than other relative clauses. However, the second conclusion has been challenged by FOX (1987), who proposed the Absolutive Hypothesis (AH) instead.

FOX (1987), on the basis of the data from English discourse, suggested that accessibility to relativisation in discourse is not determined by the grammatical position of the head noun phrase (henceforth head NP) in the relative clause, rather it is determined by the functions of relative clauses in the text. Instead of what she called 'the Subject Primacy hypothesis' (henceforth SPH), which means the subject is more accessible than other grammatical positions to relativisation, she proposed the AH. The AH states that absolutive relative clauses: Intransitive Subjects (henceforth ISU, e.g. the man who looks handsome) and DOs (e.g. the man whom I met), are more accessible to relativisation than Transitive Subject (henceforth TSU, e.g. the man who bought the book) relative clauses, because of the discourse functions of the relative clauses "rather than a special cognitive status" (FOX, 1987, p. 869). FOX explains that "Relative clauses serve to situate the referent that is being introduced as a relevant part of ongoing discourse; in a sense they justify the introduction of the referent in the first place" (FOX, 1987, p. 861).

Situating a referent in discourse can be achieved through two strategies: first, providing a static description of the referent; second, linking or anchoring[2] the referent into discourse through another referent which is well known to the addressee. The function of ISU relative clauses is to provide a description or characterization of the referent, for example, "she is married to this guy who is really quiet" (FOX, 1987, p. 859). On the other hand, the function of TSU and DO relative clauses is to anchor the referent into the text, as in "I know somebody who has her now" and "This man who I have for linguistics is really too much", respectively (FOX, 1987, p. 859). The DO relative clause links or anchors the head of the relative clause into the context using the SU in the relative clause, as is shown in the example above where the anchor is 'I' (in bold type). The TSU relative clause, on the other hand, links the head into the context using the DO of the relative clause, which is 'her' in FOX's example. Noun phrases in the SU position mostly carry given information and tend to be pronominal, so they perform the anchoring function better than noun phrases in the DO position, which usually carry new information.

A number of studies were conducted on the role of the NPAH and the discourse functions of relative clauses in predicting accessibility to relativisation such as JENSEN (1999), GORDON and HENDRICK (2005) and HOGBIN and SONG (2007) . However, these studies did not yield similar results. While GORDON and HENDRICK (2005) supported the NPAH,

HOGBIN and SONG (2007) supported the AH and JENSEN (1999) showed that the genre in which the text is written plays a significant role in supporting the NPAH or the AH. Moreover, previous studies were based on the standard varieties of languages, ignoring the differences that might exist between the standard variety and other dialects of a language. In fact, data collected from the standard variety of a language have been found "fairly unrepresentative if compared to the overall picture" (FLEISCHER, 2004, p. 236). This is found to be true in German (FLEISCHER, 2004) and English (KORTMANN, HERRMANN, PIETSCH, & WAGNER, 2005).

Studying the NPAH across varieties of the same language is particularly important in the Arabic language due to the diglossic nature of Arabic. The word diglossia was used by FERGUSON to refer to

"..a relatively stable language situation in which, in addition to the primary dialects of the language (which may include standard or regional standards), there is a very divergent, highly codified (often grammatically more complex) superposed variety" (FERGUSON, 1959, p. 336).

The coexistence of different varieties, standard and colloquial, of the same language in a community is not enough to result in a diglossic situation; there should be a great gap between formal/ written and colloquial/ spoken (HAMAD, 1992). This gap is found in Arabic.

This study investigates relativisation in three varieties of the Arabic language, Classical Arabic (CA), Modern Standard Arabic (MSA) and Iraqi Arabic (IA), and it compares the data of these varieties to the predictions of the NPAH and the AH. This study aims at answering the following research questions:

1. Which hypothesis, the NPAH or the AH, better predicts the distribution of the relative clauses in Arabic texts?
2. Does the distribution of relative clauses differ from one variety of Arabic into another for the three studied varieties (CA, MSA, IA)?

The rest of this article is organised as follows: in 1.1 an introduction about relativisation in Arabic is presented. Then, the method used in this study is described in section 2. In section 3, results and analysis are produced, which is followed by the discussion and conclusions in sections 4 and 5, respectively.

1.1 Relativisation in Arabic

The relative clause, in Arabic, is a post-nominal clause that is used to modify an item in a way structurally similar to an attributive adjective. Relative markers are used to introduce relative clauses that modify definite heads only (BASHIR, 1982; RYDING, 2005). Hence, when the modified noun is indefinite, no relative marker is used (ABDELGHANY, 2010; SUAIEH, 1980). Relative markers usually shows gender and number agreement with the head of the relative clause as is shown in Table 1.

Table 1: The Relative Markers in Arabic

Number	NOM	ACC	NOM/FEM	ACC/FEM
Singular	allaḏī	allaḏī	allatī	allatī
Dual	allaḏān	allaḏain	allatān	allatain
Plural	allaḏīn	allaḏīn	allawatī	allawātī

	or allātī	or allātī

As can be observed in Table 1, relative markers are inflected for gender and number. A distinction between nominative case and accusative case only appears with the dual relative marker. However, the agreement indicated by the relative markers in Arabic is different from that in English relative clauses as the relative marker in Arabic agrees with the head's grammatical function in the main clause and not with its grammatical function in the relative clause[3] .

The relative clauses in the Arabic dialects are similar to those in the standard varieties, MSA and CA, in being post-nominal. However, Arabic dialects differ from CA and MSA in terms of the relative markers they use in relativisation. Relative markers in Arabic dialects are not inflected for gender and number (HOLES, 2004, p. 284). All varieties use the invariable relative pronoun illi, or its variants such as (halli or yalli for Syrian Arabic or sometimes the short form ill in the Iraqi dialects[4]) for relativisation in all positions (ALTOMA, 1969; BRUSTAD, 2000; HOLES, 1990, 2004).

2 Methods

Relative clauses are collected from fifteen books, which are written in three different varieties. These books are listed in Table 2. Six CA books are included in this study; the selection of these texts has been be done by referring to books that discuss Arabic literary texts such as (JAYYUSI, 2010; ALLEN 1998) in which these texts are discussed as classical works. The second variety from which the other group of texts is collected is MSA or as it is referred to as the "contemporary variant" of CA (CUVALAY-HAAK 1997). Six MSA books are included in this study; these texts are from dates more recent than the CA (1996-2008).

The third variety is IA. Data on IA is collected from three books; all of these books belong to the twentieth century (1972-1988). The reason that only 3 books are included for this variety is that Iraqi Arabic is considered as a spoken variety; therefore, up to the researcher's knowledge, there are no other books that are written in Iraqi Arabic. Furthermore, in these texts, only the conversations between the characters are written in the IA dialect, while the rest is found in MSA, so relative clauses from conversations only are included in this study. That might result in a significantly fewer number of relative clauses in comparison with the other two varieties, yet the statistical method that is used in this study helps in avoiding the consequences of such a difference.

After finishing the data collection, the data are analysed statistically using multi-level Poisson regression analysis. This method of data analysis has been proven to be a good way of analysing textual frequencies (BAAYEN, 2008). By using this method, the effect that differences among texts might have on the results is controlled since texts are considered as a random factor.

Table 2: Texts Included in the Data Collection for this Study

Variety	texts
CA	1. Alf laila wa laila 2. Hayy Ibn Yaqḏān 3. Maqāmāt Al-ḥarīrī 4. Tārīḫ Al-ṭibarī 5. Tārīḫ Ibn Al-aṯīr 6. Tārīḫ Ibn khaldūn
MSA	1. Al-ṭarīq ʾilā tall al-muṭrān 2. Taġrīdat al-baǧaʿah 3. Al-manbūḏ 4. Tārīḫ Al-ʿarab wa haḍāratihum fi al-Andalus 5. Al-saif wa al-sīyāsah fi al-Islam. 6. Tārīḫ Al-ʿAarab al-muʿāsir
IA	1. Al-raǧʿ al-baʿīd 2. Al-naḫlah wa al-ǧīrān 3. Rubaʿīyāt Abu Gāṭiʿ

It has been found that CA books, in particular non-fiction books, are quite longer than books written in other varieties. Therefore, to maintain consistency among books in different varieties, only 200 pages are included from each book. This number has been chosen because preliminary results showed that using a lower number of pages did not provide accurate results, where the order of relative clauses is changed dramatically from 20 into 50 pages.

Because the CA is an older variety than the other two varieties, there is a diachronic dimension in the study. However, this study does not focus on the development of the language over time. This study considers CA and MSA as two varieties, as has been done by other linguists such as RYDING (2005), PASHOVA (2002) and VERSTEEGH (2001).

2.1 Relative clauses

The semantic definition that is used by KEENAN and COMRIE (1977) to identify relative clauses will also be used as a basis in this studyecause. This definition is as follows:

"We consider any syntactic object to be an RC if it specifies a set of objects (perhaps a one-member set) in two steps: a larger set is specified, called the domain of relativization, and then restricted to some subset of which a certain sentence, the restricting sentence, is true. The domain of relativization is expressed in surface structure by the head NP, and the restricting sentence by the restricting clause, which may look more or less like a surface sentence depending on the language" (KEENAN and COMRIE 1977).

The data of this study will include restrictive relative clauses only. Furthermore, structures of relative clauses that are included in this study should have at least one of the following characteristics: first, a relative clause should contain a relative marker, this is only true if the relative clause is definite (see section 1.1); second, a relative clause should contain a verb. Although the first criteria cannot be used to detect indefinite relative clauses, the second one can as is shown in example 1. In the mentioned example, *yukabbiluna* 'tie-us-up', is considered a relative clause *yukabbil* for two reasons: first, the clause identifies the noun phrase *hilman* 'dream'; second, it has a verb 'tie'.

Counting[5] relative clauses is the principal method used in this study. Relative clauses in the sample texts are counted and then classified according to the positions of the NPAH[6]. Then the percentage of relative clauses formed on each position is worked out depending on the number of relative clauses found in the texts. This method is chosen because it has been proven to be effective in previous studies, including the two major ones (FOX, 1987; KEENAN, 1975) where the frequency of relative clauses in each position is implemented as a measure of the accessibility of that position to relativisation.

The original hierarchy proposed in KEENAN and COMRIE (1977) is as follows:

1. SU > DO > IO > OBL > GEN > OCOMP

According to KEENAN and COMRIE (1977), the SU position is the most accessible position followed by the DO and the other positions going down the hierarchy, IO, OBL, GEN, OCOMP.

FOX (1987), on the other hand, claims that intransitive subject and direct object are more accessible to relativisation than the transitive subject (refer to section 1). Therefore, the assumption that the SU position is the most accessible position cannot be taken for granted especially in light of other studies that agree with Fox's claims (e.g. GORDON & HENDRICK, 2005; HOGBIN & SONG, 2007; ROLAND, DICK, & ELMAN, 2007). Thus, in this study, in order to test both the SPH and the AH, SU relative clauses will be further classified into transitive subject relative clauses (TSU) and intransitive subject relative clauses (ISU). Accordingly, hierarchy (1) will be tested as follows:

2. ISU + TSU > DO > IO > OBL > GEN > OCOMP

According to the AH presented by Fox (1987), the predicted hierarchy is:

3. ISU + DO > TSU > IO > OBL > GEN > OCOMP

In this study, both hierarchies (2) and (3) will be considered and the data of this study will show which of these hierarchies is reflected in the distribution of relative clauses in Arabic texts. Examples of relative clauses formed on grammatical positions in hierarchies (2) and (3) are presented in examples (1-6):

TSU

1. la nurīdu ḥilm-an yukabbilu-nā
   not we.want dream-ACC tie up-us
   We do not want a dream that ties us up. [MSA, 2: 249][7]

ISU

2. šifi-t al-youm wāḥid chān yištuġul wiy-yay bi-l-bank
   saw-I the-day one was work with-me in-the-bank
   I saw today one who was working with me in the bank. [IA, 1: 80]

DO

3. rafaḍ siğāra-tī allatī qaddam-tu- hā ilai-hi
   refused cigarette-my REL(3.SG.FEM) presented-I-it to-him
   He refused my cigarette that I have presented to him. [MSA, 2: 10]

IO

4. kān awal wāli faraḍ la-hu raʿiatu-hu nafaqāt-hu
   was first governor assigned to-him people-his salary-his
   He was the first governor to whom his citizens assigned a salary. [CA, 5: 306]

OBL

5. fa-ḫarağ-tu anā min al-makān allaḏī
   then-went out-I I from the-place REL(3.SG.MAS)
   kun-tu fī-hi sirra
   was-I in-it secretly
   Then I went out secretly from the place in which I was. [MSA, 1: 301]

GEN

6. tawaqaf-tu amām al-manzil allaḏī
   stopped-I in front of the-house REL(3.SG.MAS)
   ašʿala-t Ilene al-nūr fī sālat- i-hi
   turned on- FEM Ilene the-light in lounge-GEN-its
   I stopped in front of the house whose lounge Ilene turned on the light in.
   [MSA, 1: 246]

OCOMP

7. qad nazala bi-nā qawm lam narā
   already came down in-us people not we.see
   qaum qaṭ aḥsana min-hum

people never better than-them
People who we have never seen people better than came to our house...[CA. 4: 103]

4 Results and analysis

The Arabic texts included in this study yielded 2785 relative clauses. ISU (789), DO (749), TSU (601), OBL (475), GEN (161), OCOMP (8), IO (2). TSU+ ISU (1390) are significantly higher than relative clauses in other grammatical positions in the NPAH, which follows the NPAH's predictions, and supports the SPH. However, the IO position is the least frequent position (as is shown in Figure 1); this is counter to the NPAH's order, which is as follows: SU >DO> IO> OBL> GEN> OCOMP.

Thus, the results suggest that there is a gap shown in the IO position since it occurs only twice, although it is the third position in the NPAH. HOGBIN and SONG (2007) revealed similar results and offered two explanations among which the following is found true in the case of Arabic. IO is infrequently used as head NPs in the main clauses in discourse. The infrequent use of indirect object in main clauses in discourse might lead us to the expectation that IO relative clauses would occur infrequently if at all, and this is reflected in the results of this study. The number of relative clauses which have IO heads is only 6 out of 2785. This can be attributed to the fact that indirect object is restricted to the role of beneficiary or recipient and it is also connected with human or animate referents (PALMER, 1994). Because of the small numbers of IO relative clauses (2) and OCOMP (8), these two positions are excluded from the statistical models in this study, as is shown in Table 3.

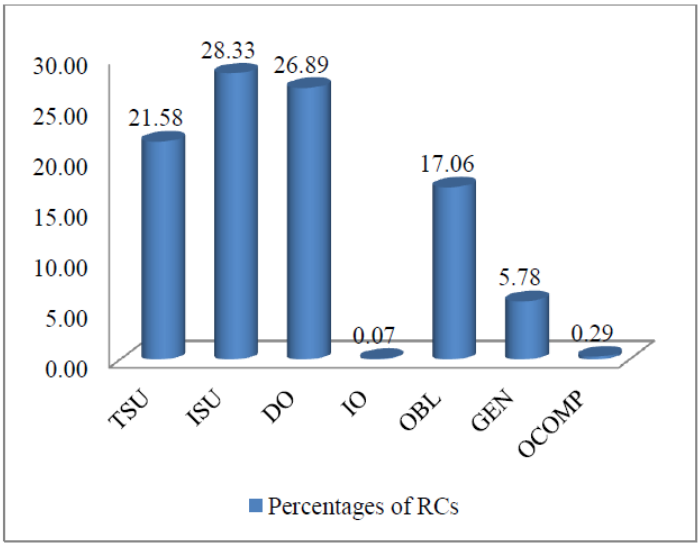

Figure 1: The distribution of relative clauses in Arabic written texts

Table 3: Model 1: Mixed Effect Poisson Regression for the Distribution of Relative Clauses in Arabic Texts Estimate

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.266	0.133	9.500	<0.001 *
TSU	-0.362	0.185	-1.953	0.050 .
DO	-0.036	0.131	-0.279	0.780
OBL	-0.547	0.127	-4.300	< 0.001 *
GEN	-1.573	0.164	-9.577	< 0.001 *
IA	-1.273	0.223	-5.698	< 0.001 *
MSA	-0.279	0.127	-2.206	0.027 *

Model 1 tests two among other predictors: (1) relative clause types with the SU position split into ISU and TSU, (2) variety; these predictors appear in the first column of the table. The dependent variable is the count of relative clauses of the relevant category. Rows 2-5 of Table 3 show a comparison between ISU and other relative clauses on the NPAH. Levels of each predictor are coded by alphabetical order; for example, in the case of the models in this study, all relative clauses on the grammatical positions of the NPAH would be compared to the DO relative clauses since DO comes first in alphabetical order. In this and the following models, alphabetic characters (a, b, c, etc.) are joined to the names of the grammatical positions for ordering purposes. For example, ISU becomes a. ISU, and TSU becomes b. TSU and so on down the NPAH to make the results appear in the order of the positions in the NPAH, (as is shown in Figure 2), which makes the analysis of the results easier. The asterisk (*) in the table indicates that the value is significant, while the dot (.) indicates that the value is approaching significance. Therefore, Model 1 compares relative clauses in all grammatical positions to ISU, as is shown in Figure 2.

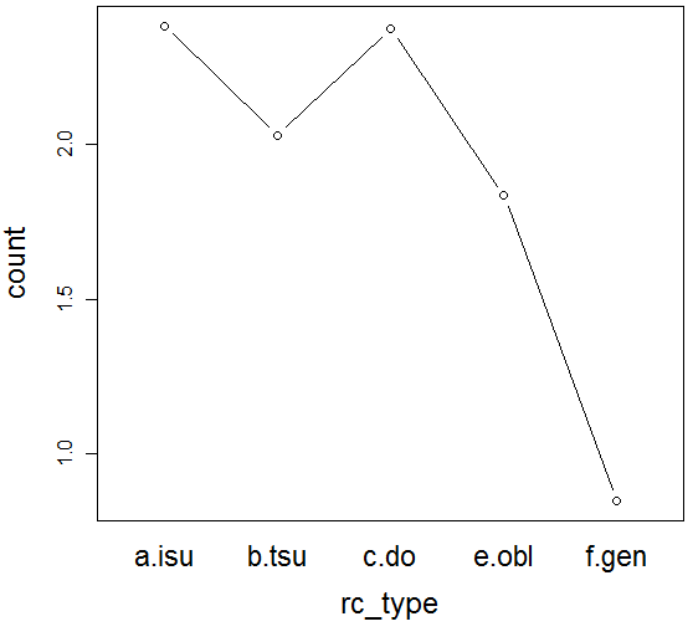

Figure 2: The distribution of relative clauses in Arabic texts

Figure 2 has five points that show the grammatical positions tested in Model 1, ISU, TSU, DO, OBL, GEN. The count axis shows how frequently the relative clauses are used on different positions within the regression model. As is shown in Figure 2, the DO position is close to the ISU position, which is the most frequently used. On the other hand, the DO position is higher than the TSU position with a significant difference (Wald's z= 3.117, $p<0.001$) [8], and there is no statistical difference between TSU and OBL (Wald's z= 1.128, p=0.259), the difference between the OBL and GEN is significant (Wald's z= -7.562, $p<0.001$).

The overall results of this study indicate that ABS relative clauses (ISU+DO) (55.22%) are used more frequently than other relative clauses on the NPAH (TSU, IO, OBL, GEN, and OCOMP). At this stage the results of this study conform to both the SPH and the AH. Therefore, to determine which of these hypotheses the results support more, a comparison is made between the SU category, which includes TSU + ISU and the absolutive category, which includes ISU +DO. A model is created in which SU and absolutive are treated as two different categories. The results suggest that the number of ABS relative clauses is significantly higher than the number of relative clauses in the SU category (Wald's z=2.805, p=0.005). This result gives some support to Fox's assertion that "it seems to be the category

ABSOLUTIVE, rather than SUBJECT, which occupies the leftmost position on the accessibility hierarchy" (FOX, 1987, p. 869) .

4.1 Relativisation across varieties

Three varieties of Arabic are included in this study, CA, MSA, IA. The CA texts reveal 1166 relative clauses, which makes up to 41.86% of all relative clauses in the Arabic corpus, the MSA texts yielded 1451 relative clauses, which make up to 52% of all relative clauses found in Arabic texts, and the IA counts yielded 168 relative clauses, which make up only 6.03% of the data of this study.

To study the influence of variety on the distribution of relative clauses in the text, a model is created to test the interaction between variety and the distribution of relative clauses, as is shown in Table 4. Model 2 tests the interaction between types of relative clauses and variety. In relation to varieties, there is a significant difference between IA and CA (Wald's z=-5.279, <0.001). On the other hand, the difference between MSA and CA does not appear to be significant. The final eight rows show the results of the interaction between variety and relative clauses. There are significant interactions between TSU and IA (Wald's z=2.369, p=0.018), and TSU and MSA (Wald's z=4.183, <0.001). There are also significant interactions between DO and IA (Wald's z=2.387, p=0.017), DO and MSA (Wald's z=2.588, p=0.010). The results suggest that the variety in which relative clauses are written plays an important role in deciding the order of the frequency of relative clauses in the upper grammatical positions (ISU, TSU, DO). The interactions are better shown in Figure 3.

Table 4: Model 2: Mixed Effect Poisson Regression for the Interaction between Relative Clauses and Varieties

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.197	0.141	8.501	<0.001 *
TSU	-0.959	0.188	-5.103	<0.001 *
DO	-0.395	0.159	-2.489	0.013 *
OBL	-0.628	0.192	-3.263	0.001*
GEN	-1.678	0.244	-6.885	<0.001 *
IA	-1.364	0.258	-5.279	<0.001 *
MSA	-0.085	0.165	-0.512	0.609
TSU:IA	0.894	0.377	2.369	0.018 *
DO:IA	0.784	0.328	2.387	0.017 *
OBL:IA	0.077	0.411	0.187	0.852
GEN:IA	0.559	0.502	1.114	0.265
TSU:MSA	1.080	0.258	4.183	<0.001*
DO:MSA	0.574	0.222	2.588	0.010 *
OBL:MSA	0.148	0.272	0.545	0.586

GEN:MSA	0.148	0.345	0.141	0.888

There are three graphs in Figure 3. Each line represents a variety; for each line, there are five points which represent the five grammatical positions tested in this model (ISU, TSU, DO, OBL, GEN). The count axis shows the frequency of relative clauses in each variety. IA (blue) relative clauses are less than the CA (black) and MSA (red). CA and MSA are very close in the ISU, OBL and GEN positions and only differ significantly in the TSU and DO positions. Similarly, IA differs significantly from CA in the three upper positions, ISU, TSU and DO, but there is a slight difference between IA and MSA in the TSU position.

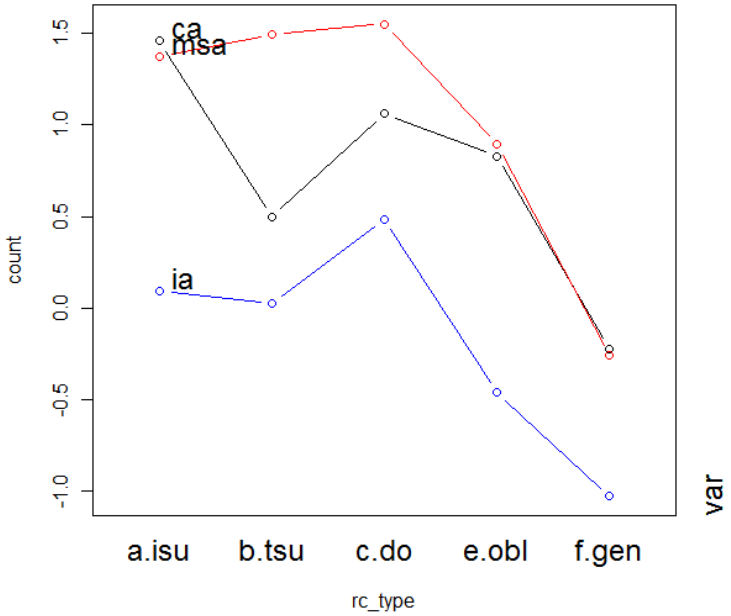

Figure 3: The interaction between relative clauses and varieties

As is shown in Figure 3, the CA line (in black) has the highest point in the ISU position followed by DO, OBL, where there is a slight non-significant difference between the two positions, and then comes the TSU position, which is followed by GEN. To test the difference between SU relative clauses (TSU+ISU) and ABS relative clauses (DO+ISU), a model was created in which TSU and ISU relative clauses were put under SU, and ABS was in-

cluded as a type of relative clauses. The results show that the ABS is significantly more frequent than the SU (Wald's z= 3.180, p= 0.002) in CA texts.

As is shown in Figure 3, the MSA line (in red) shows that the DO position appears at the highest point which indicates that it has the highest frequency. There are slight differences between ISU, TSU and DO. The OBL and GEN positions are significantly lower than the three upper positions. The difference between ABS and SU relative clauses in MSA is not significant.

Relative clauses in IA texts are found in the following descending order, DO, ISU, TSU, OBL, GEN, as is shown in Figure 3. The differences among relative clauses is found significant only between ISU and GEN (Wald's z=-2.998, p=0.003). The ABS relative clauses are used more than SU relative causes, yet the difference between these two categories is not significant. The differences among the three varieties are better shown in Figure 4, where (>>[9]) indicates that the difference is significant, (>) indicates that the difference is approaching significance, and (,) indicates that the difference is not significant.

The NPAH is not reflected in any of the three varieties considered separately, especially in CA where the frequency of OBL relative clauses is significantly higher than TSU relative clauses. The frequency of SU relative clauses has not been found higher than ABS relative clauses in any of the three varieties. Therefore, the SPH is not supported in the three varieties. For this reason and the infrequent use of IO relative clauses in the three varieties, which is considered as a violation to the hierarchical order of the NPAH, the results do not confirm to KEENAN's (1975) claim that the frequency distribution of relative clauses in texts follows the order of the NPAH.

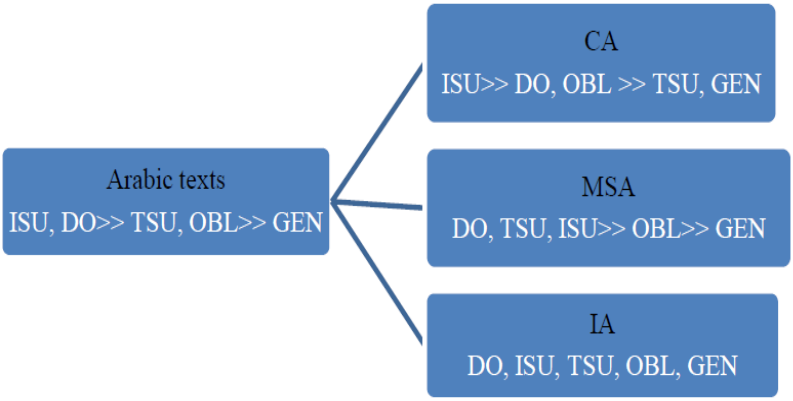

Figure 4: The distribution of relative clauses in the three varieties of Arabic

The results show that there are significant differences among the three varieties of Arabic. ABS relative clauses are used significantly more than SU relative clauses only in CA. Thus, the AH is manifested in the results of CA, which confirms FOX's (1987) claim that it is "the category ABSOLUTIVE, rather than SUBJECT, which occupies the left-most position on the accessibility hierarchy"(p. 869). The difference between CA and the other two varieties might be attributed to the stylistic changes and the linguistic structures that are used in CA but not in MSA and IA. These stylistic changes take place due to the differences between the chronological periods of CA on one hand and the other two varieties on the other. An example of these linguistics structures is the use of yuqāl li- 'said to-', which is always found in the passive form, to give the meaning of 'called' as in example (8). The use of this verb in passive contributes to the high frequency of ISU in CA.

8.	kān	fī	masǧid	yuqāl	la-hu	masǧid	ṣāliḥ
	was	in	mosque	say(PASS)	to-it	mosque	Salih

He was in a mosque which is called Salih's mosque. [CA, 4: 80]

The distribution of relative clauses differs from one variety to another as is shown in Figure 4. Whereas CA relative clauses appear in the following descending order ISU>> DO, OBL>> TSU, GEN, relative clauses used in MSA have the following descending order DO, TSU, ISU>> OBL>> GEN; and IA relative clauses appear in the following descending order DO, ISU, TSU, OBL, GEN. That is, the order of relative clauses in either CA, MSA or IA is different from the overall order of the relative clauses in the data of this study, which is ISU, DO>> TSU, OBL>> GEN.

CA is the only variety of Arabic that KEENAN and COMRIE (1977) included in their study to represent the Arabic language (p. 76). However, the results of this study show that neither of the varieties can represent the Arabic language because the overall distribution of relative clauses in the data of this study with the distributions of relative clauses in each variety does not reveal similar results. Therefore, CA does not sufficiently represent the Arabic language, which conforms with FLEISCHER's claim that the standard variety is "fairly unrepresentative if compared to the overall picture" (2004, p. 236). Thus, the variety in which relative clauses are written might not contribute to whether relative clauses follow the NPAH or the AH, yet it is an important factor that influences the general distribution of relative clauses in the texts.

5 Conclusions

The overall distribution of relative clauses in Arabic texts conforms with the AH more than the NPAH's predictions. However, looking at varieties of Arabic individually, I have found that each variety of Arabic has revealed a different pattern of relativisation. The distribution of MSA is closer to IA than to CA. IA is different from MSA in two positions only, and both of these varieties differ from CA in four positions. The overall distribution of the whole number of relative clauses in Arabic written texts does not match with any of the distributions revealed by the varieties. Moreover, the AH is reflected in the overall distribution of relative clauses as well as in CA, but not in MSA and IA.

These results seem to suggest two conclusions; first, patterns of relativisation are influenced by the variety in which it occurs. Second, the Arabic language cannot be represented by any single variety; that is, a sample of CA relative clauses is not enough for studying relativisation in Arabic. In general, therefore, these results suggest that it is important to consider different varieties of the same language in deciding accessibility to relativisation in that language. Results also bring up a question of what accessibility in a diglossic situation is. In other words, whether an individual who speaks the three varieties of Arabic has different accessibility hierarchies in his mind, which he uses according to the variety he speaks with. This question can be investigated in future research.

---

[1] Other abbreviations used in previous studies to refer to the Accessibility Hierarchy is the AH in Song (2001) and NP accessibility hierarchy in Croft (2003)

[2] Fox adapts the term 'anchor' from Prince (1981); "A discourse entity [= 'referent' in Fox's terminology] is anchored if the NP representing it is LINKED, by means of another NP, or 'anchor', properly contained in it, to some other discourse entity"

[3] There are two clauses in the relative clause sentence: the main clause and the dependent clause, which is the relative clause. For example, the sentence 'the girls I gave the books to are my friends" consists of the two clauses: "the girls are my friends" and "I gave the books to the girls". As a result, the head noun phrase 'the girls' has two functions, it is the SU of the main clause, "the girls are my friends", and at the same time it is the IO of the restrictive clause or dependent clause "I gave the books to the girls".

[4] There is more than one dialect in Iraq as the spoken dialect in Baghdad is different from the one spoken in the south of Iraq. However, relative clauses in all dialects of Iraq have the same structure. Therefore, no attempt is made in this study to distinguish among Iraqi dialects.

[5] Counting is done manually because finding relative clauses in online corpora depends on putting the exact word in the search engine; this can be done with definite relative clauses by putting the relative marker, for example 'allaḏī'. However, this is not possible in the case of indefinite relative clauses. Therefore, finding indefinite relative clauses requires reading the whole text.

[6] In this paper, the way Arabic relative clauses are identified and classified according to the grammatical positions of the NPAH is adopted from (Al- Zaghir, 2014).

[7] Reference to any of the texts is made using the variety abbreviation (e.g. MSA) and the number of the book assigned in Table 2. For example, the reference to Taġrīdat al-baġaʿah is to be made by using the symbol MSA, 2, this is followed by the page number such as MSA, 2:249

[8] Since Model 1 compares the grammatical positions to ISU, other models have been created to test whether the difference between other grammatical positions is significant.

[9] These symbols are used in this figure for the purpose of illustrating the differences among the values as far as the statistical significance, and should not be confused with (>), which is used in the original NPAH.

References

ABDELGHANY, H. (2010). Prosodic phrasing and modifier attachment in standard Arabic sentence processing. Unpublished doctoral dissertation, City University of New York.

ALLEN, R. (1998). The Arabic literary heritage: the development of its genres and criticism. Cambridge: Cambridge University Press Cambridge.

AL-ZAGHIR, Z. M. (2014). Relativisation and Accessibility: A Corpus Analysis of Relative Clauses in Arabic Written Texts. Doctoral dissertation, University of Otago, Dunedin, New Zealand.

ALTOMA, S. (1969). The problem of diglossia in Arabic: A comparative study of classical and Iraqi Arabic. Cambridge: Harvard University Press.

BAAYEN, R. H. (2008). Analyzing linguistic data: A practical introduction to statistics using R. Cambridge: Cambridge University Press.

BASHIR, S. M. (1982). A functional approach to Arabic relative clauses with implications for foreign language teaching. Unpublished Ed.D dissertation, Columbia University Teacher College, New York.

BRUSTAD, K. (2000). The syntax of spoken Arabic: A comparative study of Moroccan, Egyptian, Syrian, and Kuwaiti dialects. Washington D.C.: Georgetown University Press.

CROFT, W. (2003). Typology and universals (2nd ed.). Cambridge: Cambridge University Press.

EVIATAR, Z., & IBRAHIM, R. (2012). Multilingualism among Israeli Arabs, and the neuropsychology of reading in different languages. In M. LEIKIN, M. SCHWARTZ & Y. TOBIN (Eds.), Current Issues in Bilingualism: Cognitive and socio-linguistic perspectives (pp. 57-74). Dordrecht: Springer.

FERGUSON, C. (1959). Diglossia. Word- Journal of the International Linguistic Association, 15(2), 325-340.

FLEISCHER, J. (2004). A typology of relative clauses in German dialects. In B. Kortmann (Ed.), Dialectology meets typology: Dialect grammar from a cross-linguistic perspective (pp. 211–243). Berlin: Mouton de Gruyter.

FOX, B. A. (1987). The noun phrase accessibility hierarchy reinterpreted: Subject primacy or the absolutive hypothesis? Language, 63(4), 856-870.

GAS, S. (1979). Language transfer and universal grammatical relations. Language Learning, 29(2), 327-344.

GORDON, P., & HENDRICK, R. (2005). Relativization, ergativity, and corpus frequency. Linguistic Inquiry, 36(3), 456-463.

HAMAD, A. (1992). Diglossia in Arabic: the beginning and the end. Islamic Studies, 339-353.
HAWKINS, J. A. (1994). A performance theory of order and constituency. Cambridge: Cambridge University Press.
HOGBIN, E., & SONG, J. (2007). The accessibility hierarchy in relativisation: The case of eighteenth- and twentieth-century written English narrative. SKY Journal of Linguistics, 20, 203-233.
HOLES, C. (1990). Gulf Arabic. London: Routledge.
HOLES, C. (2004). Modern Arabic: Structures, functions, and varieties. Washington, D.C.: Georgetown University Press.
HYLTENSTAM, K. (1984). The use of typological markedness conditions as predictors in second language acquisition: The case of pronominal copies in relative clauses. In R. ANDERSEN (Ed.), Second languages: A cross-linguistic perspective (pp. 39-58). Rowley, Mass: Newbury House.
JAYYUSI, S. K. (2010). Classical Arabic stories: An anthology. New York: Columbia University Press.
JENSEN, H. B. (1999). Reduced relative clauses used in eighteenth-century diaries and letters: A sociohistorical perspective. Unpublished doctoral dissertation, Indiana University, Bloomington.
JOHNSON, D. E. (1977). On Keenan's definition of "subject of". Linguistic Inquiry, 8(4), 673-692.
KEENAN, E. (1975). Variation in universal grammar. In R. W. Fasold & R. W. Shuy (Eds.), Analysing variation in language (pp. 136-148). Washington DC: Georgetown University Press.
KEENAN, E., & COMRIE, B. (1977). Noun phrase accessibility and universal grammar. Linguistic Inquiry, 8(1), 63-99.
KEENAN, E., & COMRIE, B. (1979). Data on the noun phrase accessibility hierarchy. Language, 55(2), 333-351.
KORTMANN, B., HERRMANN, T., PIETSCH, L., & WAGNER, S. (2005). Agreement, gender, relative clauses. Berlin: De Gruyter Mouton.
MAXWELL, D. N. (1979). Strategies of relativization and NP accessibility. Language, 55(2), 352-371.
PALMER, F. R. (1994). Grammatical roles and relations. Cambridge: Cambridge University Press.
PAVESI, M. (1986). Markedness, discoursal modes, and relative clause formation in a formal and an informal context. Studies in Second Language Acquisition, 8(01), 38-55.
PRINCE, E. F. (1981). Toward a taxonomy of given-new information. In P. Cole (Ed.), Radical Pragmatics. New York: Academic Press.
ROLAND, D., DICK, F., & ELMAN, J. L. (2007). Frequency of basic English grammatical structures: A corpus analysis. Journal of Memory and Language, 57(3), 348-379.

RYDING, K. (2005). A reference grammar of modern standard Arabic. Cambridge: Cambridge University Press.
SAIEGH-HADDAD, E. (2012). Literacy reflexes of Arabic diglossia. In M. Leikin, M. SCHWARTZ & Y. TOBIN (Eds.), Current issues in bilingualism: Cognitive and socio-linguistic perspectives (pp. 43-55). Dordrecht: Springer.
SONG, J. J. (2001). Linguistic typology: Morphology and syntax. Harlow: Pearson Education
SUAIEH, S. I. (1980). Aspect of Arabic relative clauses: A study of the structure of relative clauses in modern written Arabic. Doctoral dissertation, Indiana University, Bloomington.
ZUGHOUL, M. R. (1980). Diglossia in Arabic: investigating solutions. Anthropological Linguistics, 201-217.

List of the texts used for the data collection:

Alf laila wa laila 'Arabian nights'. (1740). Beirut: Al-maktabah Al-sha3biah.
AL-HARIRI, A. A. (1873). Maqaamaat Al-hariri. Beirut: Matba3at Al-ma3aarif
AL-SAMARRAA'I, K., TAAHA, A.-W., & MATLOUB, N. (2000). Tariikh al-Aarab wa hadharatihum fi Al-andalus 'History of Arabs and their civilaizations in Andalus'. Beirut: Dar Al-kitab Al-jadiid.
AL-SHAYBANI, I. (1231). Al-kamil fi al-tariikh: Tariikh Ibn Al-athiir ' The complete history: Ibn Al-athiir's history': International Ideas Home.
AL-SHAYKH, R. (1996). Tariikh al-Aarab al-mu'asir 'Contemporary Arab history'. Cairo: Ein for Human and Social Studies.
AL-TAKARLI, F. (1980). Al-raj' al-ba'iid 'The far echo'. Beirut: Dar Al-mada Lilthaqafah wa Al-finuun.
AL-TIBARI, M. (838-923). Tariikh al-umam wa al-muluuk tariikh Al-tibari 'History of nations and kings Al-ṭibari's history': International Ideas Home.
AL-WARDAANI, S. (1999). Al-saif wa al-siyaseh fi al-Islam: al-siraa' bain al-Islam al-Nabawi wa al-Islam al-Amawi 'The sword and politics in Islam: The conflict between the Prophet's Islam and the Amawi's Islam' Beirut: Dar Al-Ra'i.
AL-YASIRI, S. (1972). Ruba'iyaat Abu Gaati' 'Abu Gaaṭi"s Quadruplet'. Baghdad: Maṭba3at Al-sha3b.
BADR, A. (2005). Al-tariiq ila tall al-mutṭraan 'The road to tal al-mutran'. Beirut: Riad Al-Rayyes Books.
FURMAAN, G. (1988). Al-nakhlah wa al-jiraan 'The palm tree and neighbours'. Baghdad: Dar Al-faraabi/ Dar Babil.
SA'IID, M. (2008). Taghriidat al-baja'ah 'The tweet of the swan'. Beirut: Dar Al-adaab
TUFAIL, A. (1900). Hayy Ibn Yaqdhaan Algeria: Fontanah wa Shraka'eh.
ZAAYID, A. (2006). Al-manbuuð 'The castaway'. Beirut: Al-daar Al-Aarabia lil'uluum-Nashiruun.

Author Index

Zainab Al-Zaghir
Department of English Language and Linguistics
Otago University, Dunedin, New Zealand
z.m.alzaghir@gmail.com

Abdulrahman Alosaimy
School of Computing
University of Leeds, Leeds, UK
scama@leeds.ac.uki

Eric Atwell
School of Computing
University of Leeds, Leeds, UK
E.S.Atwell@leeds.ac.uk

Osama Hamed
Language Technology Lab
University Duisburg-Essen, Duisburg, Germany
osama.hamed@uni-due.de

Torsten Zesch
Language Technology Lab
University Duisburg-Essen, Duisburg, Germany
torsten.zesch@uni-due.de