Hindawi
Advances in Fuzzy Systems
Volume 2019, Article ID 6254649, 10 pages
https://doi.org/10.1155/2019/6254649

Research Article

Exploring the Performance of Tagging for the Classical and the Modern Standard Arabic

Dia AbuZeina[1] **and Taqieddin Mostafa Abdalbaset**[2]

[1] *College of Information Technology and Computer Engineering, Palestine Polytechnic University, Hebron, State of Palestine*
[2] *Palestine Technical University–Kadoorie, AL-Aroub Branch, Hebron, State of Palestine*

Correspondence should be addressed to Dia AbuZeina; abuzeina@ppu.edu

Received 7 August 2018; Accepted 23 October 2018; Published 23 January 2019

Guest Editor: Omar Abu Arqub

The part of speech (PoS) tagging is a core component in many natural language processing (NLP) applications. In fact, the PoS taggers contribute as a preprocessing step in various NLP tasks, such as syntactic parsing, information extraction, machine translation, and speech synthesis. In this paper, we examine the performance of a modern standard Arabic (MSA) based tagger for the classical (i.e., traditional or historical) Arabic. In this work, we employed the Stanford Arabic model tagger to evaluate the imperative verbs in the Holy Quran. In fact, the Stanford tagger contains 29 tags; however, this work experimentally evaluates just one that is the VB ≡ imperative verb. The testing set contains 741 imperative verbs, which appear in 1,848 positions in the Holy Quran. Despite the previously reported accuracy of the Arabic model of the Stanford tagger, which is 96.26% for all tags and 80.14% for unknown words, the experimental results show that this accuracy is only 7.28% for the imperative verbs. This result promotes the need for further research to expose why the tagging is severely inaccurate for classical Arabic. The performance decline might be an indication of the necessity to distinguish between training data for both classical and MSA Arabic for NLP tasks.

1. Introduction

The part of speech (PoS) tagging, also known as word-category disambiguation, is a process to determine the tag of each word in a given input text. The tagging process uses the context to label words using syntactic tags, such as noun, adjective, verb, or preposition that are also known as parts of speech, word-classes, grammatical categories, lexical class markers, or syntactic categories. Tagging is performed either manually by linguistic experts or automatically by machine learning algorithms; intuitively, this work considers the computational track. Word tags are mainly used to describe the words and their jobs according to the context for further processing. That is, each word has a particular role based on the position and the adjacent words in the sentence. The tagset is a predefined list that generally includes some symbols, such as nouns, pronouns, adjectives, verbs, adverbs, propositions, conjunctions, and the definite and indefinite articles (sometimes called "determiners"). Of course, the tagset is prepared by the language linguistic scholars to describe the language's membership or word family. The size of the tagset is variable and depends on the requirements or the capacity of developing applications. In any case, the tagset should best fit and efficiently serve the intended purposes. Hence, there is no predefined tagset for all languages and there is no standard (i.e., unique) tagset for a certain language. Rather, it is a debatable matter.

The PoS is increasingly becoming a vital factor in the related natural language processing (NLP) applications. In fact, creating knowledge base resources (e.g., tag relationships) is one objective of the PoS tagging that can be later used in other NLP tools. In fact, PoS tagging has many roles in the field of NLP as a basic prepossessing step. For instance, some of NLP PoS tagging based applications include syntactic parsing, information extraction, machine translation, speech synthesis, and named entity recognition (NER). This work is aimed at exploring the performance of the PoS for the classical Arabic using a modern standard Arabic (MSA) tagger that is the Stanford tagger [1]. Since it is difficult to evaluate the Stanford tagger for all tags (29 tags) as it requires a large annotated corpus, the Quranic imperative verbs were chosen in the evaluation process. The Stanford tagger uses the label VB to mark the imperative verbs. That is, this work is restricted to a testing dataset that contains a list of all imperative verbs in the Holy Quran that is obtained from [2]. This work is distinguished by presenting an experimental study of the classical Arabic performance using one of the freely available taggers and, therefore, making it clear for comparison purposes. This work also aims to demonstrate the tagging problems from different points of view, such as the Arabic PoS tagging benefits and challenges, tagsets capacities, tagging algorithms, and the recent studies in this field.

In spite of the importance of taggers' performance for both classical and MSA Arabic, few studies have explored the accuracy for the classical Arabic. On the other hand, most of the previous studies focused on the tagsets and the tagging approaches. For instance, one study [3] proposed an Arabic tagset with detailed hierarchical levels of the categories and their relationships (i.e., a tree of different levels). As indicated, this study focused on the imperative verbs in the Holy Quran. The reason for choosing the imperative is that it is easier to find such annotated testing collections due to the previous effort of Arabic scholars to serve the Quranic studies. In addition, the Arabic language is distinguished to have a stand-alone form of imperative verbs whereas it is mixed with the present verb as found in the English language. For instance, the English language has the verb "go" as an imperative and present verb, while the same verbs have a different form in the Arabic language as the imperative is "اذهب" and the present is "يذهب" which are completely different words in terms of transcription and tense.

Even though the documentation of the Stanford tagger [4] indicates that the accuracy of the Arabic model is 96.26% on an MSA test portion as described in [5] and 80.14% for unknown words, our measure shows extremely less accuracy. In this work, the Stanford tagger scored only 7.28 % accuracy for a collection of Arabic imperative verbs. It is worth indicating that the Stanford tagger works at word level (i.e., the tag is given to the whole word instead of its parts, such as prefixes, stems, and suffixes as some other taggers do). Despite diacritics playing an important role in the tagging process, nevertheless, they are discarded in this work since the Stanford tagger does not consider the diacritics of the input text. However, we do keep the Hamza (e.g., أ) and the Madd (آ) symbols in the corresponding characters. That is, the testing dataset is a nonvocalized Arabic text. The output of this work highlights the importance of reinvestigating the tagging problem for the Arabic language since many of previous studies report accuracies into the nineties percentile. Reinvestigation includes different aspects of training data as either classical or MSA, the tagsets, the corpora sizes, etc.

The rest of this paper is organized as follows. In the next section, we demonstrate the benefits of the tagging for various NLP applications. In Section 3, we present why tagging is a challenging task. We exhibit the literature review in Section 4 followed by the Stanford tagset in Section 5. The proposed method is described in Section 6 and the experimental results in Section 7. Finally, we conclude in Section 8.

2. The Benefits of Tagging

The PoS tagging is the core of many NLP algorithms due to the useful information it gives about a word and its neighbors. In fact, NLP applications employ the output of the PoS tagging for different purposes, such as checking the correctness of the syntactic structure around the word. For instance, regarding the Arabic language, adjectives are preceded by nouns while nouns are preceded by adjectives in the English language such as "A beautiful school⟷ مدرسة جميلة". Similarly, nouns are preceded by verbs in the English such as "He runs fast", while the Arabic allows both directions, such as "المعلم يكتب الدرس⟷ the teacher writes the lesson" and "يكتب المعلم الدرس ⟷ the teacher writes the lesson." Therefore, the Google translator gives the same translation for two different word order Arabic sentences. Hence, knowing the syntax of word order is extremely important for some NLP applications since it limits the output candidates and increases the probabilities of correct answers. The following are some of NLP applications that utilize the PoS tagging:

(i) **Capturing common syntactical rules**: [6] presents a data mining based method to extract the common syntactical rules in the Holy Quran. The study reported that the common relationships between the words' tags (i.e., the common rule) are tag1=RP tag2=NN tag3=WP 91 ⟹ tag4=VBD 90 accuracy (0.97912). For more information of the tags, the reader refers to Section 5 in this paper.

(ii) **Enhancing the performance in speech recognition**: [7] employs the PoS to generate new words based on the neighboring word tags. The study used compound nouns that are followed by adjectives and the preposition followed by any word. After recognition, the compound words were placed back to their original states (i.e., two parts). This method shows performance enhancement.

(iii) **Named Entity Recognition (NER)**: [8] employs a tagger for named entities recognition (NER). NER aims at extracting the names such as people, organizations, locations, cities, or companies. NER is beneficial for certain applications such as classifying content for news providers. This facilitates categorization and content discovery. NER also speeds up the search process in sizeable data that contains, for instance, millions of articles. Other applications include using powering content recommendations, customer support, and research papers.

(iv) **Syntactic Parsing:** [9] employs the PoS tagging for syntactic parsing. Syntactic parsing is a process to confirm that the input sentence follows the language's formal grammar. Figure 1 shows a parsing tree for a simple sentence. The parsing tree represents the

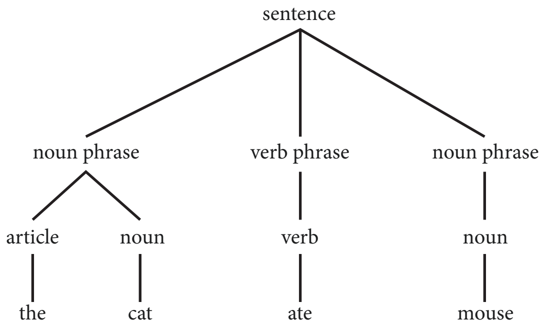


Figure 1: An example of a parsing tree.

syntactic structure of the text and is mainly used for analyzing the input sentence.

(v) **Other PoS tagging based applications include:** semantic role labeling [10], speech synthesis [11], speech recognition [12], information extraction [13], summarization [14], sentiment analysis also called opinion mining [15], diacritization [16], software engineering [17], question answering [18], translation [19], plagiarism detection [20], key phrases extraction [21], ontology [22], and extracting Arabic noun compound [23].

3. The Challenge of Tagging

That fact that a word can take different tags makes the PoS tagging a challenging task. That is, a word can be labeled by different tags based on the context. Therefore, the goal of the PoS tagging algorithms is to remove such ambiguity and label the words correctly. Table 1 shows some examples of words that take different tags based on the context. As shown in the table, the word "gold ⟷ ذهب" in sentence 1 is tagged as VBD (verb, past tense) while it is tagged as NN (noun, singular or mass) in sentence 2. Similarly, the word "Said ⟷ سعيد" in the first sentence is tagged as NNP (proper noun, singular) while it is tagged as JJ (adjective) in sentence 3. This shows how a particular word can have different labels, which is the challenge of the PoS tagging process. Hence, the problem of the PoS tagging is to resolve ambiguities by choosing the proper tag considering the surrounded words. Of course, the absence of diacritics in the Arabic formal writing system adds even more ambiguity. For instance, there is no ambiguity to know that the diacritized word "gold ⟷ ذَهَبْ" is a noun and the diacritized word "went ⟷ ذَهَبَ " is a verb. The figure also shows the tagging output for the translated sentences using the English model of the Stanford tagger.

4. Literature Review

Despite the importance of the PoS tagging for both MSA and classical Arabic, most of the previous tagging studies have mainly focused on the MSA. In addition, the literature shows there is an active research to consider suitable tagsets that truly reflect the linguistic items of Arabic as one of the morphologically rich languages. In this literature, we demonstrate the up-to-date Arabic tagging research which focused on the main aspects and components, such as the type of the training text (i.e., MSA, classical, tweets), tagsets, tagging algorithms, unknown words, stemming. In [30], the study indicated that stemming (i.e., removing prefixes and postfixes or suffixes) enhances the tagging performance. In [31], the study presented a method to tag tweets that is usually written out of the formal and proper spelling of the language. In [28], the study considered a method to handle the "unknown words", which are the words that did not appear in the training corpus. In [26], the study considered the problem which arises when estimating the transition probabilities in limited amounts of training data. The study proposed decision trees based method to handle this problem that generally occurs in the hidden Markov models (HMM) tagging technique. In [32], the study implemented the master-slave technique for the PoS tagging; they used HMM as a master tagger and maximum match (MM) and Brill taggers as slaves. There are many approaches to perform the PoS tagging, the most widely used is the statistical approach that is based on the HMM. Another approach is not-statistical, which is on a number of hand-crafted disambiguation rules to find the most appropriate tag for each word as in [33].

The recent studies of part of speech tagging include different aspects. For instance, [34] developed a part of speech tagger for the Arabic heritage. They scored an accuracy of 96.22%. They also reported that the most of the tagging errors are results of segmentation. Reference [35] employs part of speech tagging to enhance the performance of Arabic text classification. Reference [36] demonstrates part of speech tagging for the Arabic Gulf dialect. For the tagging process, they employ Support Vector Machine (SVM) classifier and bidirectional Long Short Term Memory (Bi-LSTM). Reference [37] presents a tagging based study regarding Arabic dialects identification. Reference [38] uses part of speech and semantic tagging to extract features for training Neural Machine Translation.

Table 2 presents some information regarding tagging systems, such as tagging algorithms, tagsets, corpora, and accuracies. We are aware that the accuracy is not a matter since each work has its own corpus; nevertheless, reporting these measures might give an indication of the overall accuracy of the Arabic PoS tagging. Similarity, even the tagset size is important; however, it is more important to have enough training set to cover the tags used; otherwise, zero values might be assigned to the HMM transition probabilities which raises a tagging problem.

5. Stanford Tagset

As indicated in the literature review, there are many tagsets that are used in the previous studies. Mainly, the tags are divided into two classes (i.e., categories), which are closed class and open class. The closed class has a fixed membership, such as prepositions while the open class can accept new

Table 1: Some Stanford tagged Arabic and English sentences.

Input sentences and the translation using Google translator		
Sentence 1	Sentence 2	Sentence 3
ذهب سعيد الى المدرسة Said went to school	حصل الفائز على قلادة من ذهب The winner got a gold necklace	يوم سعيد نتمناه لكم Happy day we wish you
Stanford tagger outputs (Arabic model)		
DTNN/المدرسة IN/الى NNP/سعيد VBD/ذهب	NN/ذهب IN/من NN/قلادة IN/على DTNN/الفائز VBD/حصل	VBG/لكم VBP/نتمناه JJ/سعيد NN/يوم
Stanford tagger outputs (English model)		
Said/NNP went/VBD to/TO school/NN	The/DT winner/NN got/VBD a/DT gold/JJ necklace/NN	Happy/JJ day/NN we/PRP wish/VBP you/PRP

The proposed algorithm

```
1. Obtain the text of the Holy Quran from [24] and remove the diacritics.
2. Install the full version of the Stanford Arabic model tagger from [25].
3. Have the text of the Holy Quran tagged.
4. Obtain a list of all imperative verbs in the Holy Quran from [2].
5. Find all words that have the tag VB ≡ imperative verb.
6. Compare the two lists; the one we obtained in step 5 and the list we obtained in step 4 to find the correctly tagged imperative verb.
7. Find the accuracy based on the information that is obtained in step 6.
```

Algorithm 1: Evaluating imperative verbs using the Stanford tagger.

Table 2: Some of the literature tagging research.

No	Ref.	Tagging Method	Tagset Size (tags)	Corpus Size (words)	Accuracy (%)
1	[26]	A decision tree based tagger	110	78K &500K	91.65 ⟶ 97.18
2	[27]	Support Vector Machines (SVM)	24	140K	95.49
3	[28]	Hidden Markov Models	24	29,300	95.0 ⟶ 97.1
4	[29]	SVM and a Neural network	21	6,844	91.0
5	[1]	Maximum Entropy based tagger	29	588,244	96.1

words especially in the technology fields as "to fax". Table 3 shows the 29 tags of the Arabic model of the Stanford tagger.

6. The Proposed Method

This section presents the steps that we follow to find the performance of the Stanford tagger against the Quranic imperative verbs. The first step is the tagging process that produces an annotated text file of the entire Quranic sentences. Then we used a number of Python programs to extract the correctly tagged imperative verbs as well as the wrongly tagged imperative verbs, etc. The textual version of the Holy Quran is obtained from the Quran Printing Complex, Saudi Arabia website [36]. Algorithm 1 summarizes the implemented steps.

The input testing set is the nondiacritized textual form of the Holy Quran. Figure 2 shows what the testing set looks like. The figure contains the first chapter or Surah of the Holy Quran (Sūrat al-Fātiḥah—The Opening) in addition to the first three sentence of the second chapter (Sūrat Al-Baqarah—The Cow). Figure 3 shows the output of the Stanford tagger for the Quranic sentences that appear in Figure 2. As it is observed, Figure 3 shows some correctly tagged words such as the following: { المستقيم/DTJJ, أنعمت /VBD, الذين/WP, يؤمنون/VBP}. The figure also shows some wrongly tagged words such as the following: { بسم/NNP, إياك /VBD, اهدنا/VBD}.

The tagger output that is shown in Figure 3 is the main content that can be used for further analysis to find the behavior of the tagger. Of course, the correctly tagged words are required (i.e., the correct labels of the testing words) in order to measure the accuracy which adds more difficulty in this kind of research. In other words, if we want to measure the accuracy for the "entire" Holy Quran, we have to prepare an annotated version of the Holy Quran which is a difficult task. This is why we chose a subset that contains only the imperative verbs.

7. The Experimental Results

For the evaluation, we used the full Stanford tagger (129 MB) that is freely available at the website of the Stanford natural language processing group through the link [37]. It is relatively simple to execute the tagger by running the command shown in Figure 4 in the Windows, Command Prompt program. That is, the tagger does not require special systems, as we run it on the Command prompt of the Windows 10 home operating system. The figure shows that 77,749 words are tagged in a very short time.

The experimental results are demonstrated in Table 4. The table exposes the information regarding the imperative verbs; however, this work can be expanded to measure the performance for different tags such as noun or verb. Similarly, it is possible to find the performance of the Stanford tagger regarding the prepositions in the Holy Quran, in which the same steps can be followed to get the accuracy for prepositions, or the overall accuracy of all tags. Finally, exploring the performance for the Stanford tagger as well as for the other taggers will lead to discover more weakness points to be avoided in future NLP systems.

8. Conclusions

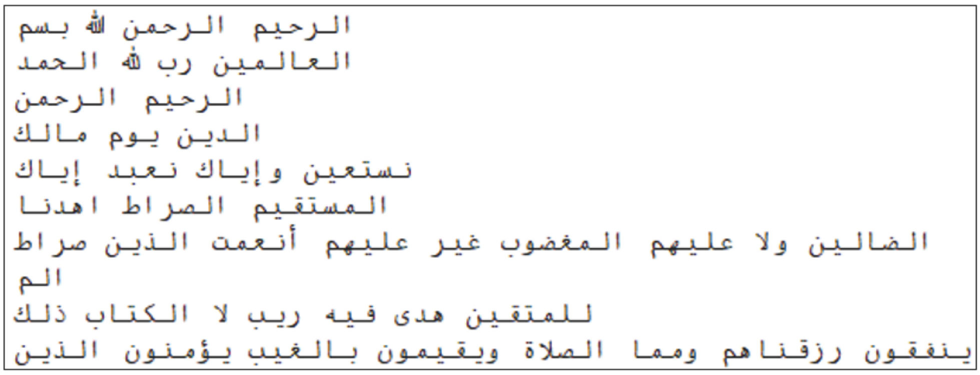
This work explored the performance of the Stanford tagger for the Arabic language. The experimental results show the importance of distinguishing between training data when preparing taggers. That is, the tagger that is prepared for poetry is different from the tagger that is prepared for prose.

Table 3: The tagset of the Stanford Arabic model tagger.

	Tag	Meaning with examples	#	Tag	Meaning with examples
1	DTJJ صفة معرفة	DT + Adjective النفطية، الجديد، الأبيض، العزيز	16	PRP ضمائر مفرد	Personal pronoun هي، هو، نحن
2	DTJJR صفة للمقارنة معرفة	DT + Adjective, comparative الكبرى، العليا، الوسطى	17	PRP$ ضمائر جمع	Possessive pronoun هم
3	DTNN اسم معرف	DT + Noun, singular or mass المنظمة، العاصمة، المال، العلم	18	RB ظرف زمان	Adverb هناك، حيث
4	DTNNP اسم علم معرف	DT + Proper noun, singular العراق، القاهرة، المسيح	19	RP حرف نفي	Particle لم، لا، لن
5	DTNNS اسم جمع	DT + Noun, plural السيارات، الولايات، الثمرات	20	VB فعل امر	Verb, the imperative form ادخلوا، ادع، قل، خذ
6	IN حرف جر	Preposition or subordinating conjunction (حرف جرامصدري)	21	VBD فعل ماض	Verb, past tense أعلن، قالت، كان
7	JJ صفة	Adjective جديدة، قيادية، سميع، شديد	22	VBG مصدر من فعل	Verb, gerund or present participle (نية، اعتبار، قول)
8	JJR صفة تتعلق بالمقارنة	Adjective, comparative أدنى، كبرى، أربى	23	VBN مبنى للمجهول	Verb, past participle يقام، يعد، تتلى
9	NN اسم نكرة	Noun, singular or mass إنتاج، نجم، أمة	24	VBP فعل مضارع	Verb, non3rd person singular present (يعمل)
10	NNP اسم علم	Proper noun, singular أوبك، لبنان، إبراهيم	25	VN اسم مفعول	Verb, 3rd person singular present (مسجلة، مدعومة)
11	NNS اسم جمع نكرة	Noun, plural توقعات، طلبات، درجات	26	WP اسم موصول	Wh pronoun الذي، اللذين
12	NOUN_QUANT	Noun, quantity الربع، ثلثي، كل، بعض	27	WRB ظرف مكان	Wh adverb حيث، كيف، كلما
13	CC حروف العطف	Coordinating conjunction ثم، و، كما، بل	28	ADJ_NUM المتعلق بالعدد	Adjective, Numeric السابع، الرابعة، ثالث
14	CD ارقام	Cardinal number مئة، ألفين، ثلاث، سبع	29	UH غامض	Interjection unusual kind of word (اللهم، كلا، نعم)
15	DT الـ ، أسماء الاشارة	Demonstrative pronouns هذه، ذلك، هذا			

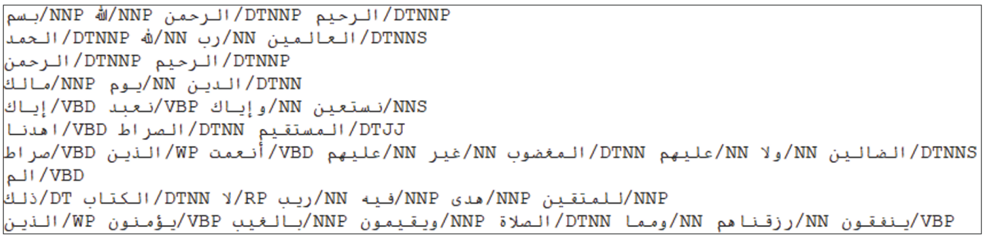
Table 4: The experimental results.

Measure	Total
The number of imperative verbs in the Holy Quran	1,848 verbs
The number of imperative verbs after removing duplicates	741 verbs
The number of words tagged as imperative verbs (i.e. VB)	282 words
The correctly tagged imperative verbs by comparing with the correct list. This leads to the correctly tagged list that includes: فاصبروا ، اقتلوا ، فاحذروا ، فادع ، اصبروا ، ارجعوا ، فافعلوا ، فانشزوا ، فارجعوا ، انشزوا ، فاستقيموا ، اهبطوا ، ارجعي } ، وصابروا ، اخرجوا ، قولوا ، خذ ، اغفر ، فأذن ، فادعوا ، اذكروا ، فأذنوا ، ادع ، قل ، فولوا ، فاعتبروا ، كلوا ، فآمنوا ، فادخلوا ، وقوموا ، اعدلوا ، اخسئوا ، انظر ، اجعلوا ، فأجمعوا ، اذهبوا ، اركعوا ، أخرجوا ، كونوا ، فابتغوا ، اخرج ، {فاعتزلوا ، فاعلموا ، قم ، فاستبقوا ، انفخوا ، فاصدع ، ادعوا ، اتبعوا ، فاعبدوا ، فاسجدوا ، صلوا ، انظروا ، ادخلوا	54 words
The rest is the wrongly classified verbs (i.e. classified out of VB tag) This list includes, for instance: {وأدخل، وأدخلنا، أدخلني، وأدخلهم، ادخل، ادخلا، وادخلوا، ادخلوها، فادخلوها، ادخلي، فادخلي، وادخلي} This list intentionally contains words that have the same root “دخل ≡ enter”. This gives an indication of the richness and the derivative nature of the Arabic language.	687
$\text{Accuracy} = \frac{\text{Correctly taggged imperative verbs}}{\text{All imperative verbs}}$	$\text{Accuracy} = \frac{54}{741} = 7.28\%$



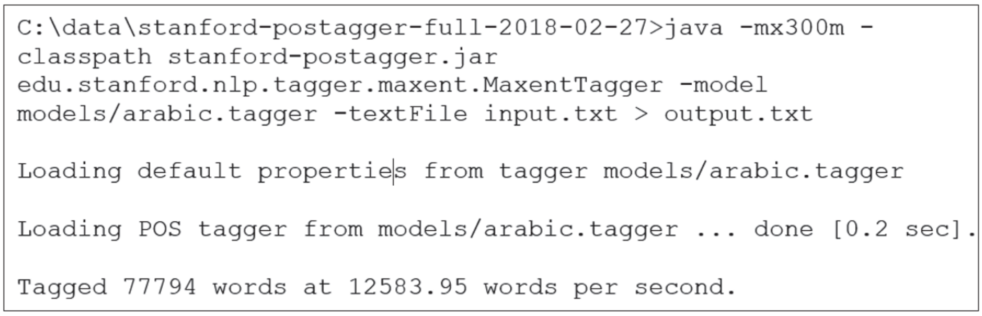
الرحيم الرحمن الله بسم
العالمين رب لله الحمد
الرحيم الرحمن
الدين يوم مالك
نستعين وإياك نعبد إياك
المستقيم الصراط اهدنا
الضالين ولا عليهم المغضوب غير عليهم أنعمت الذين صراط
الم
للمتقين هدى فيه ريب لا الكتاب ذلك
ينفقون رزقناهم ومما الصلاة ويقيمون بالغيب يؤمنون الذين

Figure 2: A part of Quranic testing set.



```
بسم/NNP الله/NNP الرحمن/DTNNP الرحيم/DTNNP
الحمد/DTNNP لله/NN رب/NN العالمين/DTNNS
الرحمن/DTNNP الرحيم/DTNNP
مالك/NNP يوم/NN الدين/DTNN
إياك/VBD نعبد/VBP وإياك/NN نستعين/NNS
اهدنا/VBD الصراط/DTNN المستقيم/DTJJ
صراط/VBD الذين/WP أنعمت/VBD عليهم/NN غير/NN المغضوب/DTNN عليهم/NN ولا/NN الضالين/DTNNS
الم/VBD
ذلك/DT الكتاب/DTNN لا/RP ريب/NN فيه/NNP هدى/NNP للمتقين/NNP
الذين/WP يؤمنون/VBP بالغيب/NNP ويقيمون/NNP الصلاة/DTNN ومما/NN رزقناهم/NN ينفقون/VBP
```

Figure 3: A part of the Stanford tagger output.



```
C:\data\stanford-postagger-full-2018-02-27>java -mx300m -
classpath stanford-postagger.jar
edu.stanford.nlp.tagger.maxent.MaxentTagger -model
models/arabic.tagger -textFile input.txt > output.txt

Loading default properties from tagger models/arabic.tagger

Loading POS tagger from models/arabic.tagger ... done [0.2 sec].

Tagged 77794 words at 12583.95 words per second.
```

Figure 4: The running command of the Stanford tagger.

Similarly, the tagger used in the old text is different than one that is prepared for MSA. The tweets are also different from MSA. This is the main observation of this study as the performance of the MSA based tagger sharply declines for the classical text. The study also shows the differences between the literature tagsets which promotes a better study and work for a standard tagset that thoroughly covers the language. However, preparing a comprehensive tagset requires an extensive double check of the transition probabilities between all tags since zero probabilities might give errors especially in HMM based taggers. As a future work, it might be good to merge between hand-crafted rules and statistical approaches for the PoS tagging. It is also important to consider word segmentation before tagging, as many Arabic words contain different tags, such as a preposition and a noun for example as in the word" بالمدرسة ≡ at school". Finally, the Arabic language is characterized by sizeable vocabulary as well as an extremely rich morphology that requires an endless effort towards optimal NLP systems. It is worth indicating [39, 40] as they have a thorough discussion of the Arabic challenges, as well as some recent Arabic NLP contribution such as stemming, corpora, and classifiers.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank the Palestine Polytechnic University (PPU) and the Palestine Technical University–Kadoorie for their support to conduct this research.

References

[1] K. Toutanova and C. D. Manning, "Enriching the knowledge sources used in a maximum entropy part-of-speech tagger," in *Proceedings of the the 2000 Joint SIGDAT conference*, pp. 63–70, Hong Kong, October 2000.

[2] "The Quran Imperative Verbs," http://jamharah.net/showthread.php?p=51814.

[3] I. Zeroual, A. Lakhouaja, and R. Belahbib, "Towards a standard Part of Speech tagset for the Arabic language," *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 2, pp. 171–178, 2017.

[4] "Stanford tagger," https://nlp.stanford.edu/software/tagger.shtml.

[5] D. Chiang, M. Diab, N. Habash, O. Rambow, and S. Shareef, "Parsing arabic dialects," in *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2006*, pp. 369–376, Italy, April 2006.

[6] D. E. M. A. Abuzeina and M. H. Alsaheb, "Capturing the Common Syntactical Rules for the Holy Quran: A Data Mining Approach," in *Proceedings of the Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences, NOORIC 2013*, pp. 670–680, Saudi Arabia, December 2013.

[7] D. AbuZeina, W. Al-Khatib, M. Elshafei, and H. Al-Muhtaseb, "Toward enhanced Arabic speech recognition using part of speech tagging," *International Journal of Speech Technology*, vol. 14, no. 4, pp. 419–426, 2011.

[8] B. Farber, D. Freitag, N. Habash, and O. Rambow, "Improving NER in Arabic using a morphological tagger," in *Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008*, pp. 2509–2514, Morocco, May 2008.

[9] A. Shahrour et al., "Camelparser: A system for arabic syntactic analysis and morphological disambiguation," in *Proceedings of the of COLING 2016, the 26th International Conference on Computational Linguistics*, System Demonstrations, 2016.

[10] D. Gildea and D. Jurafsky, "Automatic labeling of semantic roles," *Computational Linguistics*, vol. 28, no. 3, pp. 245–288, 2002.

[11] J. R. Bellegarda, *Combined statistical and rule-based part-of-speech tagging for text-to-speech synthesis*, vol. 719, 006. 6, 2014, U.S. Patent No. 8,719,006.

[12] R. Beutler, *Improving speech recognition through linguistic knowledge. Diss. ETH Zurich*, 2007.

[13] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld, "Open information extraction from the web," *Communications of the ACM*, vol. 51, no. 12, pp. 68–74, 2008.

[14] A. Z. Arifin, M. Z. Abdullah, A. W. Rosyadi, D. I. Ulumi, A. Wahib, and R. W. Sholikah, "Sentence Extraction Based on Sentence Distribution and Part of Speech Tagging for Multi-Document Summarization," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 16, no. 2, p. 843, 2018.

[15] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, "Sentiment Analysis Is a Big Suitcase," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 74–80, 2017.

[16] A. Shahrour, S. Khalifa, and N. Habash, "Improving Arabic diacritization through syntactic analysis," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pp. 1309–1315, Portugal, September 2015.

[17] N. Ibrahim and F. Khamayseh, *A Semi-Automated Generation of Activity Diagrams from Arabic User Requirements*, 2015.

[18] Q. Zhonghua and Y. Liu, "Sentence Dependency Tagging in Online Question Answering Forums," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 554–562, Jeju, Republic of Korea, 2012.

[19] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 48–54, Edmonton, Canada, May 2003.

[20] A. S. Hussein, "A plagiarism detection system for Arabic documents," *Advances in Intelligent Systems and Computing*, vol. 323, pp. 541–552, 2015.

[21] M. Nabil, A. F. Atiya, and M. Aly, "New approaches for extracting Arabic keyphrases," in *Proceedings of the 1st International Conference on Arabic Computational Linguistics, ACLing 2015*, pp. 133–137, Egypt, April 2015.

[22] A. Al-Arfaj and A. Al-Salman, "Arabic NLP tools for ontology construction from Arabic text: An overview," in *Proceedings of the 1st International Conference on Electrical and Information Technologies, ICEIT 2015*, pp. 246–251, Morocco, March 2015.

[23] M. Al-Mashhadani and N. Omar, "Extraction of arabic nested noun compounds based on a hybrid method of linguistic approach and statistical methods," *Journal of Theoretical and Applied Information Technology*, vol. 76, no. 3, pp. 408–416, 2015.

[24] "Quran Printing Complex," https://www.qurancomplex.org/.

[25] "The Stanford natural language processing group," https://nlp.stanford.edu/software/tagger.shtml.

[26] I. Zeroual and L. Abdelhak, "Adapting a decision tree based tagger for Arabic," in *Proceedings of the International Conference on Information Technology for Organizations Development, IT4OD 2016*, Morocco, April 2016.

[27] M. Diab, K. Hacioglu, and D. Jurafsky, "Automatic tagging of Arabic text," in *Proceedings of the HLT-NAACL 2004: Short Papers*, pp. 149–152, Boston, Massachusetts, May 2004.

[28] A. Mohammed et al., "Probabilistic arabic part of speech tagger with unknown words handling," *Journal of Theoretical & Applied Information Technology*, 2016.

[29] R. Alharbi12 et al., *Part-of-Speech Tagging for Arabic Gulf Dialect Using Bi-LSTM*, 2018.

[30] I. Zeroual, M. Boudchiche, A. Mazroui, and A. Lakhouaja, "Developing and performance evaluation of a new Arabic heavy/light stemmer," in *Proceedings of the the 2nd international Conference*, pp. 1–6, Tetouan, Morocco, March 2017.

[31] M. Abdulkareem and S. Tiun, "Comparative analysis of ML POS on Arabic tweets," *Journal of Theoretical and Applied Information Technology*, vol. 95, no. 2, pp. 403–411, 2017.

[32] A. H. Aliwy, "Combining POS taggers in master-slaves technique for highly inflected languages as Arabic," in *Proceedings of the 2015 1st International Conference on Cognitive Computing and Information Processing, CCIP 2015*, India, March 2015.

[33] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, Prentice-Hall, New Jersey, 2000.

[34] E. Mohamed, "Morphological Segmentation and Part-of-Speech Tagging for the Arabic Heritage," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 17, no. 3, pp. 1–13, 2018.

[35] A. Al-Thubaity, A. Alqarni, and A. Alnafessah, "Do Words with Certain Part of Speech Tags Improve the Performance of Arabic Text Classification?" in *Proceedings of the the 2nd International Conference*, pp. 155–161, Lakeland, FL, USA, April 2018.

[36] S. Ramakrishnan et al., *Part-of-Speech Tagging for Arabic Gulf Dialect Using Bi-LSTM*, 2012.

[37] M. Zampieri, S. Malmasi, N. Ljubešić et al., “Findings of the VarDial Evaluation Campaign 2017,” in *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pp. 1–15, Valencia, Spain, April 2017.

[38] Y. Belinkov et al., *Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks*, 2018, arXiv preprint arXiv:1801.07772.

[39] F. S. Al-Anzi and D. Abuzeina, “Stemming impact on Arabic text categorization performance: A survey,” in *Proceedings of the 5th International Conference on Information and Communication Technology and Accessibility, ICTA 2015*, Morocco, December 2015.

[40] F. S. Al-Anzi and D. AbuZeina, “Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing,” *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 2, pp. 189–195, 2017.

Applied Computational Intelligence and Soft Computing
The Scientific World Journal
Mathematical Problems in Engineering
Journal of Engineering
Advances in Multimedia
Modelling & Simulation in Engineering
Advances in Artificial Intelligence
International Journal of Reconfigurable Computing
Advances in Fuzzy Systems
Scientific Programming
Advances in Human-Computer Interaction
International Journal of Engineering Mathematics
Advances in Civil Engineering
Journal of Computer Networks and Communications
Journal of Robotics
International Journal of Computer Games Technology
International Journal of Biomedical Imaging
Journal of Electrical and Computer Engineering
Computational Intelligence and Neuroscience