ISSN 2278-3091

Volume 8, No.1.1, 2019

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse3481.12019.pdf

https://doi.org/10.30534/ijatcse/2019/3481.12019

An Arabic Language-Processing Computer Program for Building Verb Formats Using a New Method That Incorporates Pronunciation

Ahmad Abdulqader Abuseeni [1], Mutaz Rasmi Abu Sara [*2], Rashad J. Rasras [*3]

[1]Department of Management Information Systems, Taibah
University, Medina, Saudi Arabi
aseeni@taibahu.edu.sa

[2]Department of Computer Science, Taibah
University, Medina, Saudi Arabia
mabusara@taibahu.edu.sa

[3]Department of Computer Engineering, Faculty of Engineering Technology
Al-Balq' Applied University, Amman, Jordan
rashad.rasras@bau.edu.jo

ABSTRACT

In the information retrieval process of linguistic and other Arabic language processing, when using dictation writing, ignoring diacritical marks and punctuation, results become inaccurate specially in research and classification processes, for this reason it became necessary to find an alternative process which can deal with punctuation and diacritic marks words based on Pronunciation.

Our program aims to produce Arabic verb derivatives without the adoption of previous traditional grammars; here, we use a new method based on the way we pronounce and write. This study suggests a developmental program for Arabic phonetic transcription rather than dictation writing. The program will accept Arabic script with diacritical marks, offering a modern way of avoiding errors when dealing with Arabic texts.

The interred root will be converted into separate letters based on the pronunciation of the word and diacritical marks, then the program will select one of fourteen existing formulas in the Arabic language. It will then select the desired verb type among five reaction types. If the user has not selected a specific formula or a certain type, the program will extract sixty-two derivatives of the input.

The main findings, we get as a result of suggesting our program was producing sixty-two derivatives of each verb root, that could be used in many ways for Arabic language processing and information retrieval by mapping each root in the Arabic language with its derivatives that will make it easy to retrieve a root when the text contains any of these derivatives. This will make Arabic information retrieval and language processing more accurate and easier. In addition, the program can be used to facilitate many tasks related to linguistic processes, such as Arabic text classification, learning, and teaching and other processes.

Key words: Pronunciation Forms; Verb Derivatives; Arabic Text Classification

1. INTRODUCTION

Communication through speech and language is possibly the most important function in human beings. It allows for a variety of commands and understandings through vocal speech patterns. Arabic is one of the world's oldest languages [9]. It is spoken across the Middle East, North Africa, and the Horn of Africa. A Central Semitic language, Arabic is closely related to Aramaic and Hebrew. Arabic is spoken by approximately 420 million speakers and is ranked as the sixth official language of the world (UN, 2013).

There are 10.5 million Arabic speakers with access to the Internet, compared to 287.5 million English speakers [2]. Unfortunately, efforts to improve Arabic information search and retrieval, in comparison to English or French, are limited and modest. The inadequacy of text-processing advancements for Arabic highlight the very complicated morphological structure of the Arabic language.

Our program will convert interred roots into separate letters and select a formula from among the fourteen existing in the Arabic language, as well as a desired verb type from among five types. If the user does not select a specific formula or a certain type, the program will extract sixty-two derivatives of the input.

In the following sections, we will discuss previous studies, a statement of the problem, the significance of the study, and in Section 2 the algorithm will be introduced in detail. The program is detailed in Section 3, followed by the conclusion, results, and future work.

The programs currently used to extract derivatives of Arabic verbs rely not on the method of word pronunciation, but on literal translation; they do not take into account the diacritics. The diacritics play an important role in many forms of language structure, especially in the cases of vowels, ablaut, and slurring. Unfortunately, several of the current Arabic classification systems do not recognize or apply these marks [10].

Consequently, many classification errors may occur, especially when the words have the same spelling but the meaning is different. For example, the word "فَرس", which means ‘mare,’ and " فُرس", which means ‘Persians’, are grouped as the same in existing classifiers [4].

We hope that our program will help in information retrieval and language processing that when the verb derivative appears in the text, it will be matched to the original root, this helps immensely in information retrieval and classification processing.

Restoring a word to its roots is one of the foundations upon which information retrieval and classification of Arab texts proceeds. Our program will help this process through mapping sixty-two derivatives to each variable; for example, the text in the word " يجلسون" would be linked with the verb "جلس", because in our program when we enter the word " جلس" , the word "يجلسون " will be one of the derivatives.

In addition to the above, this study will help individuals to use Arabic scripts with diacritical marks to convert text into Arabic speech symbols in a phonetic transcription form, subsequently, this will help identify and distinguish between words that have the same characters but different meanings.

Thus, the importance of this research arises because of the following reasons: (1) converting Arabic words into their equivalent phonetic transcription forms would help in the study of the Arabic language, including all its details; (2) it would solve many of the Arabic language problems arising from neglecting diacritical marks; (3) thus, converting Arabic text into its basic units (phonetic transcription forms) and mapping each root to its derivatives would pave the way for improving the current approaches to information retrieval and language processing.

2. OVERVIEW

[8] Notes that phonology is the theoretical study of how sounds are used in a language to encode meaning as governed by the rules of pronunciation. Pronunciation can be viewed as a skill and as the act of producing sounds of speech that enable an individual to use spoken words with the correct stress, rhythm, articulation and intonation patterns, with reference to a standard of correctness such as Received Pronunciation. This is in contrast to phonetics, which [5] refers to as the production of the sounds of speech within a language.

[1] Made a comparison between dictation writing and phonetic transcription and the results indicate outweigh the use of phonetic transcription.

[3] Say that due to the fact that Arabic word variants are formed by the usage of affixes (prefixes, infixes, and suffixes), the morphological language suffers from the unavailability of a standard Arabic translation algorithm. Other problems arise from the wide range of articles, conjunctions, prepositions, prefixes, and suffixes that can be attached to a single word. Furthermore, [6] argue that Arabic suffers from the common problem of irregularity of singular and plural nouns, which is not related to simple affixing. In Arabic, diacritical marks appear either above or below letters and play an essential role when it comes to semantically and phonetically distinguishing between two identical words with the same characters [7]. Consequently, one of the main reasons that the Arabic language suffers from processing errors is neglect of the diacritical marks.

3. THE ALGORITHM

The first step is to input the root; the program will then read the word and convert each letter to its equivalent phonetic transcription form based on the rules of [1], illustrated in table 1.

Table 1: Convert Each Letter to Its Equivalent Phonetic Transcription

If the letter has " ◌ً ", then it is written with " ◌ً – " after it in the phonetic transcription form.
If the letter has " ◌َ " under it, it is written with a " ◌َ " after it in the phonetic transcription form.
If the letter has " ◌ُ ", then it is written with a "-◌ُ " after it in the phonetic transcription form.
If the letter has " ◌َ ", then no diacritical mark is written after it.
If the letter has " ◌َ ", then the letter is duplicated in the phonetic transcription form. The " أ " and " ء" are to be dealt with in the same way as letters and written as " ء" in the phonetic transcription form.
The letter "ي" is represented by "◌َ -◌َ -" if it and the previous letter have no diacritical mark.
The letter "و" is represented by " ◌ُ - ◌ُ - " if it and the previous letter have no diacritical mark.
If the letter does not have any diacritical marks, then it is written as is.

In the second stage, the program will apply a set of rules to find the sixty-two derivatives for each verb; these rules will be initialized depending on the word pronunciation [2].

The derivatives will be initialized by determining the number of letters, then applying the following rules based on the rules of [2], illustrated in table 2.

Table 2: The Derivatives Initialization

If the word entered has three letters, then: If the first letter is "ي" or "أ" or "ت", then delete the first letter.
If the first letter is "ي" or "أ" or "ت" followed by (◌َ◌َ) or (◌َ◌ُ) or (◌َ), then delete the first letter and the diacritical mark after it.
Else, the derivative is left as it is.
If the word entered has four letters, then:
If the first letter is "ي" or "أ" or "ت", then delete the first letter.
If the first letter is "ي" or "أ" or "ت" followed by (◌َ◌َ) or (◌َ◌ُ) or (◌َ), then delete the first letter and the diacritical mark after it.
Else, the derivative is left as is.
If the word entered has five letters, then:
If the first letter is "أ" and after that is (◌َ◌َ) or (◌َ◌ُ) or (◌َ), and after that is the letter "ن"then delete the first letter "أ" and the diacritical mark after it and the letter "ن".
If the first letter is "ي" or "أ" or "ت", then delete the first letter.
If the first letter is "ي" or "أ" or "ت" followed by (◌َ◌َ) or (◌َ◌ُ) or (◌َ), then delete the first letter and the diacritical mark after it.
Else, the derivative is left as is.
If the word entered has six letters, then:
If the first letter is "ي" followed by "س" followed by "ت" and (◌َ◌َ), delete the first letter "ي", the letter "س", and the diacritical mark after it.
If the first letter is "أ" with (◌َ) followed by "س" followed by "ت" and (◌َ◌َ), delete the first letter "أ" and (◌َ), and the letter "س" and the letter "ت" and the diacritical mark after it.
If the first letter is "أ" followed by (◌َ◌َ) or (◌َ◌ُ) or (◌َ), and after that the letter "ن", then delete the first letter "أ" and the diacritical mark after it and the letter "ن".
If the first letter is "ي"or "أ" or "ت", then delete the first letter.
If the first letter is "ي" or "أ" or "ت" followed by (◌َ◌َ) or (◌َ◌ُ) or (◌َ), then delete the first letter and the diacritical mark after it.
Else, the derivative is left as is.

The overall algorithm is illustrated in Figure 1.

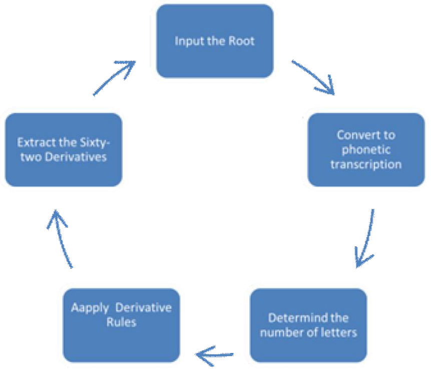

Input the Root
Convert to phonetic transcription
Determind the number of letters
Aapply Derivative Rules
Extract the Sixty-two Derivatives

Figure. 1: Algorithm methodology

4. THE PROGRAM

The application was built using C#, because it is a powerful and type-safe object-oriented language that supports the concepts of encapsulation, inheritance, and polymorphism. This enables developers to build a variety of robust and secure applications that can be run on the .NET Framework.

5. USER INTERFACE

The user interface was designed to be easy to use and attractive, so that the user can insert the root and then see the written pronunciation for the interred root. Then, he or she will be able to select the verb form and type to get the desired results. Also, the user can choose "All formulas" to get all sixty-two derivatives for the interred root. The user interface is illustrated in Figure 2.

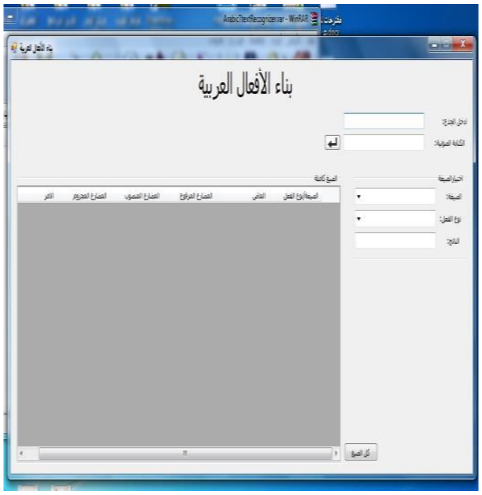


Figure. 2: User interface

Figure 3 shows the written pronunciation for the interred root.

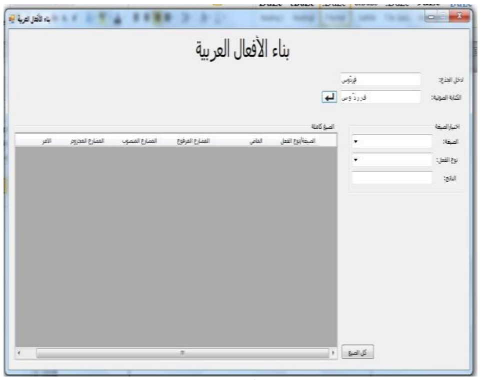


Figure 3: Written pronunciation of the interred root

Figure 4 shows the drop down menu that allows the user to select from the 14 formulas of verbs.

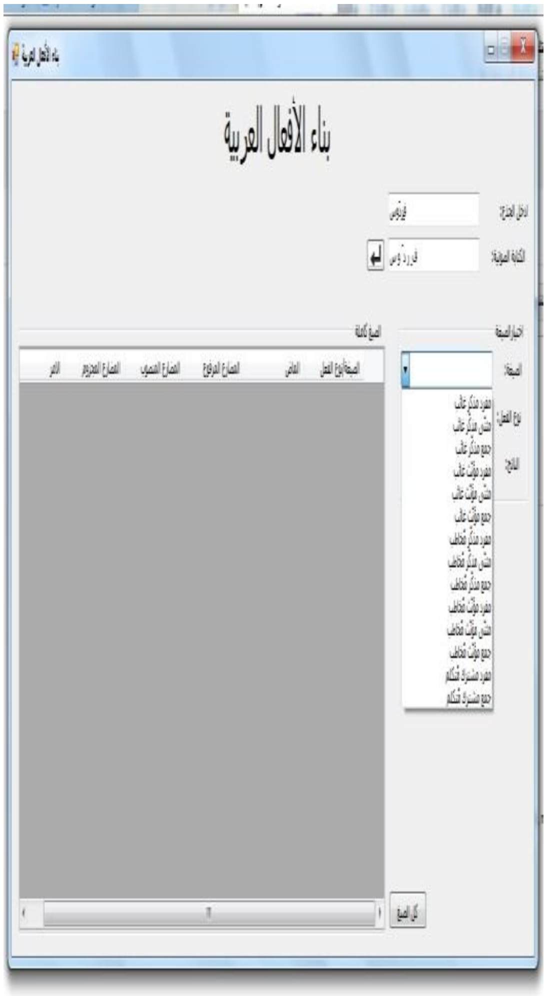


Figure 4:. Formulas of verbs

Figure 5 shows the dropdown menu to select from the 5 types of verbs.

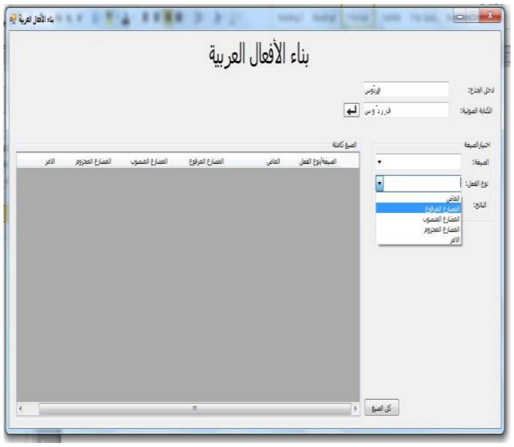
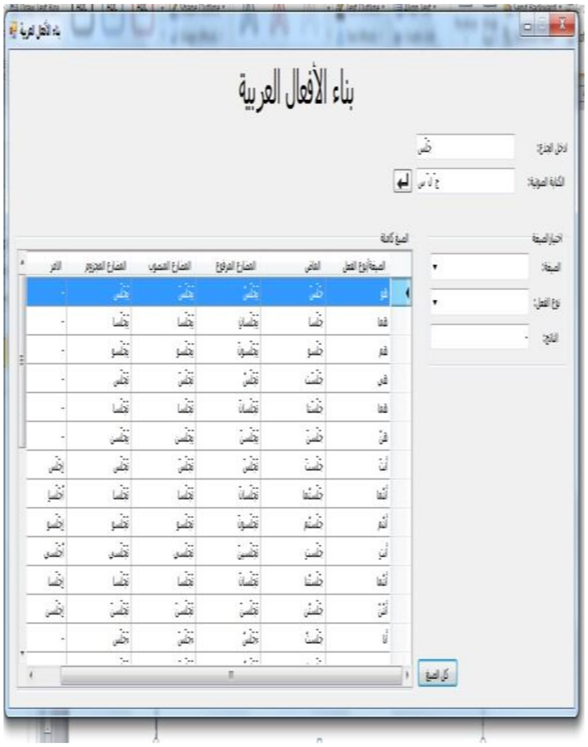


Figure 5: Formulas of verbs

Figure 6 shows all the derivatives of the selected root.



الامر	المضارع المجزوم	المضارع المنصوب	المضارع المرفوع	الماضي	الصيغة/نوع الفعل
-	يَخْلِسْ	يَخْلِسَ	يَخْلِسُ	خَلَسَ	هو
-	يَخْلِسَا	يَخْلِسَا	يَخْلِسَانِ	خَلَسَا	هما
-	يَخْلِسُوا	يَخْلِسُوا	يَخْلِسُونَ	خَلَسُوا	هم
-	تَخْلِسْ	تَخْلِسَ	تَخْلِسُ	خَلَسَتْ	هي
-	تَخْلِسَا	تَخْلِسَا	تَخْلِسَانِ	خَلَسَتَا	هما
-	يَخْلِسْنَ	يَخْلِسْنَ	يَخْلِسْنَ	خَلَسْنَ	هنّ
اِخْلِسْ	تَخْلِسْ	تَخْلِسَ	تَخْلِسُ	خَلَسْتَ	أنتَ
اِخْلِسَا	تَخْلِسَا	تَخْلِسَا	تَخْلِسَانِ	خَلَسْتُمَا	أنتما
اِخْلِسُوا	تَخْلِسُوا	تَخْلِسُوا	تَخْلِسُونَ	خَلَسْتُمْ	أنتم
اِخْلِسِي	تَخْلِسِي	تَخْلِسِي	تَخْلِسِينَ	خَلَسْتِ	أنتِ
اِخْلِسَا	تَخْلِسَا	تَخْلِسَا	تَخْلِسَانِ	خَلَسْتُمَا	أنتما
اِخْلِسْنَ	تَخْلِسْنَ	تَخْلِسْنَ	تَخْلِسْنَ	خَلَسْتُنَّ	أنتنّ
-	أَخْلِسْ	أَخْلِسَ	أَخْلِسُ	خَلَسْتُ	أنا

Figure 6: Formulas of verbs

6. CONCLUSION

Using Arabic text based on the Arabic phonetic transcription will help users to deal with Arabic scripts with diacritical marks, it is envisaged this will have a major role in resolving many Arabic information systems' problems arising from neglecting diacritical marks. It can also help Arabic language learners with reading and understanding the relationship between words and ineffective dictation writing, especially in dealing with vowels.

The proposed program produces Arabic verb derivatives using a new method that depends on pronunciation and writing, so it provides simplicity, realism and accuracy, to help native Arabic-speakers in daily life, it offers a fluent knowledge of Arabic verb pronunciation and writing.

Producing sixty-two derivatives of each verb root, help foreign Arabic language learners to understand the derivatives, moreover the new program can be used in many ways for Arabic language processing and information retrieval.

We recommend to use the proposed program in the process of mapping stage during classification approaches and research methodologies to relate the sixty-two derivatives with one root, we consider that will reduce unrelated retrieved information, as it will alleviate the difficulties faced by those who use diacritical marks and verbs formulas.

REFERENCES

[1] A. A. Abuseeni, "Computational method for accurate classification of Arabic texts based on Arabic phonetic transcription", *GLOBAL JOURNAL FOR RESEARCH ANALYSIS*. Vol. 4,issue -3, pp. 9- 16, March 2015.

[2] S. M. Abusini, " Morphological and phonetic reading in Arabic language structure", *Zarka J. Res. Stud.* 7:1, 2, 2005.

[3] A. A. Abuseeni, S. Abusini, " An approach for extracting Arabic word root based on phonetic transcription forms", An academic perspective 139, *14th International Business Information Management Association Conference*, Istanbul, Turkey, 2010.

[4] M. Alghamdi, "Voice Print: Voice Onset Time as a Model", *Arab Journal for Security Studies and Training.* 21. 42: pp. 89-118 , 2006.

[5] P. Carr, *English Phonetics and Phonology: An Introduction*, Oxford University Press, Oxford, ©2012, Wiley-Blackwell, August 2012.

[6] Larkey, L.S., Ballesteros, L., Connell, M.,. "Improving stemming for Arabic information retrieval. Light stemming and co-occurrence analysis", *Tampere, Finland.* pp. 275-282, 2002.

[7] M. Momani, J. Faraj," A novel algorithm to extract tri-literal Arabic root", *J. Amer. Soc. Inform. SCI.* Tech. 61, 583-591, 2010.

[8] D. Odden, *Introducing Phonology*, Cambridge University Press, Cambridge, 2008.

[9] (2014) website. [Online]. Available: http://en.wikipedia.org/wiki/UN_Arabic_Language_Day

[10] Wa'el, M., Eljinini, M., Mohammad, A., Ghatasheh, M., "Performance of NB and SVM classifiers in Arabic text data", *Business Transformation through Innovation and Knowledge*, 2010.