Arabic Verb Translator

Muhammad Nassirudin - 13511044
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
m.nassirudin@students.itb.ac.id

***Abstract*—Arabic has a unique and unfamiliar grammar compared to common European language concepts. Arabic has more complicated system as a word can be formed by adding, eliminating, or substituting letter within stem word. Fortunately, Arabic grammar has some pattern over word form. This concept is called as maṣdar (مصدر) that literally means 'source', 'origin', or 'root'. This paper goes through this concept to make a translator of Arabic verb in any form. The idea is to make use of pattern matching over maṣdar to search for appropriate stem word, from which the translation being made.**

***Index Terms*—Arabic, pattern matching, translator, verb.**

I. Introduction

Arabic is widely used in Middle East countries. Moreover, Arabic also spoken by many more people in the world as it is the main language in Islam. Arabic, however, follows an unfamiliar grammar compared to most of language system we already know. Most people consider that Arabic grammar is really hard. Despite this, there is a standard pattern that rules the language. We only focus on verb since it is a good example as a first glance. Besides, Arabic verbs also store many other informations within it, such as tenses and number of person involved. Not to mention it, Arabic also has more complex system to state doer/actor/subject in a sentence.

Arabic uses concept of maṣdar 'root' as the source of pattern/model used in word formation. Verb may contain 3 to 6 letters, but most of them is in 3 letters. Such verb is called as trilateral verbs. For simplicity, we will only focus on trilateral verbs in this paper and every 'verb' mentioned in this paper refers to trilateral verb.

This paper makes use of maṣdar concept to make a program that has functionality to translate Arabic word into English with all information it stores at once—that is, the tense it possesses and number of person/actor involved as a subject. The idea is to find appropriate pattern, then extract the time and doer information stored in the verb. Together with the stem word, a translation with grammar is generated.

This idea is simple yet hard to implement. This is because of programming world represents non-ASCII character (that is, alphabets in English and some common marks in writing (?, !, $, etc.)) in a different way. As for non-ASCII character, like Arabic letters we use in the program, Unicode system is used. A common representation for Unicode characters is UTF-8 (UCS Transformation Format—8-bit). This is a Unicode standard and commonly used over the world.

The input of program will be an Arabic verb in Arabic letter, hijāiyyah (هجائية), whereas the output will be in English. The input will be read as a sequence of Unicode characters and then compared to list of defined maṣdar. The comparison will be done using regular expression as it is the most suitable approach for our need. On top of that, all maṣdar are stored in database also with Unicode system. We need also store some English verbs together with its 3 basic forms: infinitive, past tense, and past participle. Beside these 3 forms, there are also 3rd person singular form (adding suffix –s onto verb) and gerund (adding suffix –ing onto verb).

II. Arabic Reading/Writing System

Before we proceed further, we should understand how Arabic reading and writing system works. Unlike English, Arabic script is read from right to left. The orientation, however, is the same as English, i.e. horizontal.

Arabic has 28 letters, some of which have almost the same pronunciation and other have almost the same shape. The list of Arabic letters is provided in table 2-1. In the table, we use romanization/transliteration to make sound as similar with actual sound as possible. The standard we use for romanization is from Library of Congres/ALA.

Arabic alphabet, which is called hijāiyyah (هجائية), has 3 forms depending on where it appears, that is (a) beginning of word, (b) in the middle of word, and (c) at the end of word. However, Arabic does not have capital letter system.

Notice that, from table 2-1, most of letters are romanized with consonants and only 3 of them with vocal (a, i, and u). Actually that is not really what happens. Each Arabic letter, in fact, stands for a single consonant. Letters with romanization a, i, or u do state that they sound like so as extended vowel from preceeding consonant. Arabic vowels are not a part of alphabet, instead they are a mark written over or below a letter. These marks only stand for short vowels and are called ḥarakāt (حركات).

Table 2-1. Arabic alphabet (هجائية)

Romanization	Arabic	Romanization	Arabic
ḍ	ض	a/ā	ا
ṭ	ط	b	ب
ẓ	ظ	t	ت
ʻ	ع	th	ث
gh	غ	j	ج
f	ف	ḥ	ح
q	ق	kh	خ
k	ك	d	د
l	ل	dh	ذ
m	م	r	ر
n	ن	z	ز
h/t	ه/ة	s	س
w/ū	و	sh	ش
y/ī	ي	ṣ	ص

Arabic vowel system is slightly different from what we know. It gives a mark to a letter depending on how the letter should sound. Any letter may sound a, i, u, stay the same (be a consonant), an, in, or un. The last 3 sounds seem strange to us since Arabic letter basically has a letter pronounced like ‘n’. Indeed, this matter has something to do when we come to talk about grammatical aspect. We do not worry about this since verb won’t use these 3 vowel sounds. In the table 2-2, we provide the mark together with its position relative to the letter.

Table 2-2. Arabic Vowel Marks

Mark	Name	Sound	Position
◌َ	fatḥah	a	above
◌ِ	kasrah	i	below
◌ُ	ḍammah	u	above
◌ْ	sukūn	(no vowel)	above
◌ً	fatḥatayn	an	above
◌ٍ	kasratayn	in	below
◌ٌ	ḍammatayn	un	above

These marks are not always attached to alphabet. They will be omitted when the verb’s meaning is obvious. This is a common thing in Arab, so we should also consider an input without vowel mark. It is done by assuming the mark is the same with mark on maṣdar.

III. MAṢDAR

All pattern we will make use is provided in maṣdar. This is the most important aspect to build our program. So, we should know basic concept about maṣdar before we go through pattern matching.

As stated before, maṣdar literally means ‘root’ or ‘origin’ which later we call ‘model’ as its main use is to be a reference in forming a verb. What we compare with is not really the letter; instead, the number of letter and ḥarakāt for each letters are being compared (although, some letter comparison are unavoidably involved). Beside maṣdar, we need also Arabic dictionary with list of stem word for every verb. We make use of it when there is no ḥarakāt within input.

When no ḥarakāt within input, we will search for approriate maṣdar just from the pattern and the number of letter involved. After we find it, we will extract stem/root word of the verb and then look up on the Arabic database for the right meaning. As for input with ḥarakāt, we will use number of letter, ḥarakāt, and letter pattern as parameter of searching. Again, we will use Arabic database to find its meaning in English. There is one important fact that we have not mentioned in this paper. Arabic verb does not really depend on ḥarakāt to be distinguished one from another. A sequence (in our case, three) letters only and can only stand for exactly one meaning.

After all basics we have gone through, now we will take a look at actual maṣdar table. Generally, Arabic verbs are classified into 3 major kinds based on the time it takes place. They are past tense, present tense, and imperative tense. In fact, stem word of Arabic verbs take form of past tense. We will now go through these 3 tenses one by one.

A. Past Tense/alfi’lu almāḍī (الفعل الماضى)

Past tense shows the action/doing is done in the past, just like in English system. Special property of Arabic past tense is that it is the stem word of Arabic verbs. Literally, a stem word is subjected to a third male person. For instance, kataba (كتب) actually means ‘he wrote’. The translation, however, will see the context, so we will translate kataba (كتب) as ‘to write’. All verbs (in root form) in past tense must be falling into one of these 3 forms: fa’ala (فَعَلَ), fa’ila (فَعِلَ), and fa’ula (فَعُلَ).

Furthermore, just like what we stated before, the pattern store not only time it takes place, but also the number of person involved. For instance, in the example given above, stem words have singular third male person as doer. In fact, Arabic has strict rule to follow over classification of doer involved in a verb. Table 3-a-1 gives us a first glance about this rule. We only provide pattern for model fa’ala (فَعَلَ) since another two have the same pattern except ḥarakāt of second letter (one is with kasrah and another is with ḍammah).

Table 3-a-1. Maṣdar for Model fa’ala (فَعَلَ)

Role	Genitive	Plural	2 Persons	Singular
3rd person	Male	فَعَلُوْا	فَعَلَا	فَعَلَ
	Female	فَعَلْنَ	فَعَلَتَا	فَعَلَتْ
2nd person	Male	فَعَلْتُمْ	فَعَلْتُمَا	فَعَلْتَ
	Female	فَعَلْتُنَّ		فَعَلْتِ
1st person	Male/female	فَعَلْنَا		فَعَلْتُ

The last thing that should be noted that all we have discussed is in active voice, i.e. the verb states that the doer do something. The pattern for passive voice is given by these 2 steps:

(a) change ḥarakāt of first letter to be marked as ḍammah; and
(b) change ḥarakāt of second letter to be marked as kasrah.

So, if we want to say ‘to be written’, we say kutiba (كُتِبَ) instead of kataba (كَتَبَ). (Notice that kutiba (كُتِبَ) actually means ‘he was written/prescribed’.)

B. Present Tense/alfi’lu almuḍāri’ (الفعل المضارع)

When we talk about present tense verb in Arabic, it actually does not have the same meaning as we know in English. In Arabic, present tense means the actor is currently doing the action *or* the actor will do it in the future. It is safe for us to see them as the same since both hold almost the same functions in daily conversation.

Present tense verbs are formed from past tense verbs. The formation is done either by adding, eliminating, or substituting one or more letter. Table 3-b-1 provides formation of the model fa’ala (فَعَلَ).

Table 3-b-1. Present Tense for Model fa’ala (فَعَلَ)

Role	Genitive	Plural	2 Persons	Singular
3rd person	Male	يَفْعَلُوْنَ	يَفْعَلَانِ	يَفْعَلُ
	Female	يَفْعَلْنَ	ثَفْعَلَانِ	تَفْعَلُ
2nd person	Male	تَفْعَلُوْنَ	ثَفْعَلَانِ	تَفْعَلُ
	Female	تَفْعَلْنَ		تَفْعَلِيْنَ
1st person	Male/female	نَفْعَلُ		أَفْعَلُ

C. Imperative Tense/fi’lu al-amru (فعل الأمر)

Arabic has another tense called imperative tense that does not belong to 16 tenses in English. Imperative tense, as what stated by its name, is a command verb. This tense is used to give another a command to do something. Imperative tense is only for 2nd person male or female, singular, 2 persons, or plural. To form an imperative verb, we follow these 4 steps:

(a) take the present tense of the verb;
(b) omit the first letter;
(c) change ḥarakāt of last letter to be in sukūn; and
(d) add alif without hamzah in the beginning of word.

For instance, if we want to say ‘(you) write!’ we say uktub (اكْتُبْ). The formation is illustrated as follows.

يَكْتُبُ – كْتُبُ – كْتُبْ – اكْتُبْ

We may have noticed that the first letter in the final form is without ḥarakāt. This is so because that is the rule. So, how do we read it? The first letter follows the ḥarakāt of third letter. Since the third letter in the example above is ḍammah, we read it as uktub, nor iktub neither aktub. The example given above stands for 2nd singular male person. Other patterns are given in table 3-c-1.

Table 3-c-1. Imperative Tense Pattern for Model fa’ala (فَعَلَ)

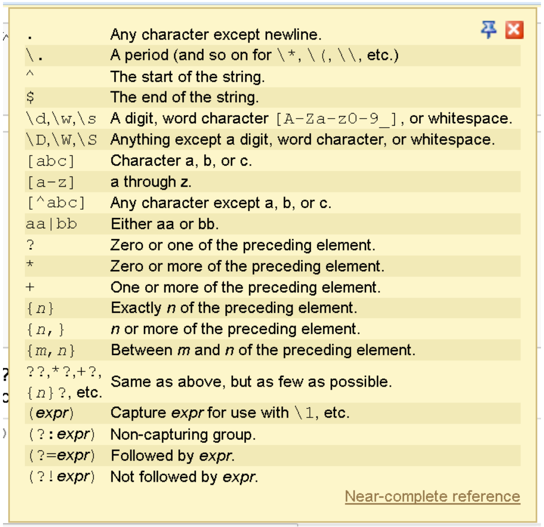
Role	Genitive	Plural	2 Persons	Singular
2nd person	Male	اُفْعُلُوْا	اُفْعُلَا	اُفْعُلْ
	Female	اُفْعُلْنَ		اُفْعُلِى

There are a lot more pattern in Arabic grammar system beside what we have revealed above, but we will not go any deeper since above pattern is sufficient to demonstrate our program. Another reason is to keep our paper simple. Therefore, what we should know about domain knowledge (i.e. Arabic grammar) has been completed. We can safely continue to pattern matching algorithm, that is our main program algorithm.

IV. Regular Expression

A regular expression (regex or regexp for short) is a special text string for describing a search pattern [5]. Regex defines a pattern that a string must follow. If it fits perfectly, the string is accepted and that is what we search for. If no string (or substring) matches the pattern, the word/pattern we search for is not found.

Regular expression is commonly used to search certain pattern within a sequence of string. There are some standard syntax that can be used to make a general pattern we want string (or substring) to follow. Picture 4-1 gives some standard regex syntax from http://regexpal.com/.



.	Any character except newline.
\\.	A period (and so on for *, \\(, \\\\, etc.)
^	The start of the string.
$	The end of the string.
\d,\w,\s	A digit, word character [A-Za-z0-9_], or whitespace.
\D,\W,\S	Anything except a digit, word character, or whitespace.
[abc]	Character a, b, or c.
[a-z]	a through z.
[^abc]	Any character except a, b, or c.
aa|bb	Either aa or bb.
?	Zero or one of the preceding element.
*	Zero or more of the preceding element.
+	One or more of the preceding element.
{*n*}	Exactly *n* of the preceding element.
{*n*,}	*n* or more of the preceding element.
{*m*,*n*}	Between *m* and *n* of the preceding element.
??,*?,+?, {*n*}?, etc.	Same as above, but as few as possible.
(*expr*)	Capture *expr* for use with \1, etc.
(?:*expr*)	Non-capturing group.
(?=*expr*)	Followed by *expr*.
(?!*expr*)	Not followed by *expr*.

Picture 4-1. Regex Standard Syntax

Regular Expression is of DFA (Deterministic Finite Automata). Therefore, regex supports looping and ordered by a pattern we define beforehand. Regex works by scanning the whole string and find all matched substrings. In our case, anyway, there is no need to search entire string/word because an input only consists of exactly one verb.

There are many other algorithms that can be used for pattern matching. Regular expression, however, is chosen as the property of pattern we have is already has some fixed character but there are also changeable characters. This makes other algorithm, like KMP and BM, lack of efficiency. In a word, we need the algorithm to find exact/matched pattern/maṣdar at once, not one by one.

V. ANALYSIS

For simplicity, we will only use model fa'ala (فَعَلَ) for analysis purpose. Any other model can simply follow the exact steps to reach the same goal, i.e. determine the meaning of an Arabic verb. First of all, let's reexamine ḥarakāt concept over the model.

Like what we already stated in earlier writing, the input is of 2 kinds: with ḥarakāt and without ḥarakāt. As for case one, input with ḥarakāt, the representation of input (which is encoded in UTF-8) follows below convention.

Let's take kataba (كتب) as example. The UTF-8 (hex) code for letter ك is 0xD9 and 0x83; letter ت is 0xD8 and 0xAA; and letter ب is 0xD8 and 0xA8. In the first case, there are also three ḥarakāt for each character. For our instance, there is only one ḥarakāt, that is fatḥah with encoding 0xD9 and 0x8E. In Unicode system, ḥarakāt will be written (in hexadecimal) right after corresponding letter.

As for second case, an input without ḥarakāt, we will simply search only the number of letter in the pattern and some fixed letter that cannot be omitted or substituted. All ḥarakāt in the root pattern will be ignored. It can be done like that because the uniqueness Arabic verb from only consonants it consists of.

The pattern matching will be done by using defined regular expression and the comparison will be done two by two characters/hexadecimals since a Unicode character is represented in two hexadecimals.

For above example, the output is 'he wrote'. Another input is yaktubu (يكتب). The output from the program is 'he is writing/he will write'.

VI. CONCLUSION

String matching with regular expression as base pattern can be useful to search for appropriate pattern over all available patterns. One of practice in the real world is to translate Arabic verb together with the tense and the doer(s) involved in the action.

It can be done because Arabic grammar follows some pattern in formation of a verb. There are so many such pattern that many persons consider Arabic is difficult because of this very aspect. Our program, however, make people way of learning Arabic easier.

REFERENCES

[1] Al-Atsary, Abu Hamzah Yusuf, *Belajar Mudah Bahasa Arab*. Bandung: toobagus publishing, 2010.

[2] Zakaria, A., *Ilmu Nahwu Praktis, Sistem Belajar 40 Jam*. Garut: ibnazka press, 2004.

[3] http://arabic.speak7.com/arabic_verbs.htm. Diakses pada jam 20.30 WIB, tanggal 19/12/2013.

[4] http://www.al-bab.com/arab/language/lang.htm. Diakses pada jam 21.00 WIB, tanggal 19/12/2013.

[5] http://www.regular-expressions.info/. Diakses pada jam 14.28 WIB, tanggal 20/12/2013.

[6] http://regexpal.com/. Diakses pada jam 14.49 WIB, tanggal 20/12/2013.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 29 April 2010

Ttd,

Muhammad Nassirudin
13511044