Chapter 22
Morphological Analysis of Ill-Formed Arabic Verbs for Second Language Learners

Khaled Shaalan
The British University in Dubai, UAE

Marwa Magdy
Cairo University, Egypt

Aly Fahmy
Cairo University, Egypt

ABSTRACT

Arabic is a language of rich and complex morphology. The nature and peculiarity of Arabic make its morphological and phonological rules confusing for second language learners (SLLs). The conjugation of Arabic verbs is central to the formulation of an Arabic sentence because of its richness of form and meaning. In this research, we address issues related to the morphological analysis of ill-formed Arabic verbs in order to identify the source of errors and provide an informative feedback to SLLs of Arabic. The edit distance and constraint relaxation techniques are used to demonstrate the capability of the proposed system in generating all possible analyses of erroneous Arabic verbs written by SLLs. Filtering mechanisms are applied to exclude the irrelevant constructions and determine the target stem which is used as the base for constructing the feedback to the learner. The proposed system has been developed and effectively evaluated using real test data. It achieved satisfactory results in terms of the recall rate.

INTRODUCTION

Language is a way of communicating ideas and feelings among people by the use of conventional symbols. People need to learn second languages to be able to communicate with other non-native speakers. Second language acquisition is a difficult task, especially for adults. There are various methods to acquire a new language and all of them require some form of feedback, which can be described as a reaction to what has been said or written. This feedback most often comes from other human beings with whom the language

DOI: 10.4018/978-1-60960-741-8.ch022

learner is interacting. There are, however, other means to receive automated feedback. One is the use of intelligent language tutoring system (ILTS) software. This software contains exercises for language learners. Their response to these exercises is analyzed by the system which provides some form of feedback that could identify the exact source of error a learner has made.

There are some types of exercises that are easy to be error diagnosed, such as multiple choice questions and gap filling exercises, because the number of possible answers is very limited. Simple methods can then be employed to provide a feedback to learners. Whenever the range of possible answers is large or even infinite, specialized intelligent tools are needed. For instance, in the case of exercises requiring learners to produce sentences in the language they are learning, Natural Language Processing (NLP) tools and techniques are necessary to analyze the learner's answer and produce intelligent feedback. In a morphological rich language such Arabic, an inflected verb can form a complete sentence (e.g. the verb سمعتك /samiEtuka/[1] [heard-I-you]) contains a complete syntactic structure in just a one-word sentence. In this case the NLP tools and techniques are also required to analyze the learner's answer and produce intelligent feedback.

The work presented in this chapter addresses issues related to the morphological analysis of ill-formed Arabic verbs written by beginner to intermediate SLLs. The proposed system is an integral part of an ILTS for Arabic. SLLs of Arabic, however, face a lot of morphological and syntactic difficulties during their language learning tasks, such as *word formation*, *word recognition*, *sentence construction*, and *disambiguation*. This complexity in learning Arabic makes addressing the diagnosis of *Arabic lexical errors* a challenge. This has motivated us to develop a tool that addresses the *word formation* problem that is usually faced by SLLs of Arabic. This is achieved by making the proposed tool analyzes the learner's answer which is used to provide learner with some form of feedback that identifies the exact source of the error s/he might made.

The edit distance and constraint relaxation techniques are used to generate all possible analyses of erroneous Arabic verbs. Filtering mechanisms are applied after the extraction of affixes and stems to exclude the irrelevant constructions and determine the target stem. For each case, a morphological gloss is incrementally formulated which is to be used as a base for constructing the feedback to the learner.

Many research, however, have attacked the problem of Arabic morphological analysis (Ahmed 2000; Beesley 2001; Buckwalter 2002; Darwish 2002; Al-Sughaiyer and Al-Kharashi 2004; Attia 2006). But to the best of our knowledge few research have addressed the problem of analysis of ill-formed Arabic words (e.g., Bowden and Kiraz 1995; Ahmed 2000; Buckwalter 2002). Bowden and Kiraz (1995) investigated the problem of correcting words in Semitic languages including Arabic language. Their approach integrated with morphological analysis using a multi-tape formalism. The model had two-level error rules that handle the following error types: vowel shift, deleted consonant, deleted long vowels, and substituted consonant. Moreover, Ahmed (2000) and Buckwalter (2002) applied some spelling relaxation rules (to deal with orthographic variations like the use of the final letter ه /h/ instead of the letter ة /p/) to get all possible analyses of an erroneous word. However, these systems only handle performance errors made by native speakers of the language. In contrast to the proposed system that handles competence errors made by nonnative speakers of Arabic. It does so by incorporating morphological knowledge and non-native intuitions into its algorithm. It does not depend on simple string matching between correct and erroneous words

The rest of this paper is structured as follows. Section 2 discusses lexical error analysis prob-

lem. Section 3 presents a background on Arabic morphology. Section 4 introduces an analysis of common Arabic lexical errors. Section 5 describes the proposed model. Section 6 discusses the results from an experiment. Finally, in Section 7 we give some concluding remarks.

LEXICAL ERROR ANALYSIS

Lexical (word) analysis is the first step for tools and applications that concerns recognizing the detailed structure of the inflected word. It is also necessary at this step to verify that the input word is linguistically correct (i.e. belongs to the respective language and conforms to its morphological rules). This is the basic level of checking and was included quite early in text processing software to ensure that the next levels of analysis are based on linguistically correct input words. Lexical errors can be classified into three classes (Tschichold 2003):

- **Errors in word formation.** These errors are related to the correct application of morphological and phonological rules. For example, it is incorrect to conjugate weak (irregular) verbs with regular verb morphological rules such as generating the imperfect form of نوصل /na-**wo**Sil/ instead of نصل /na-Sil/ (we arrive), which incorrectly leaves the weak letter (و) /w/ of the assimilated (first weak) verb in the imperfect form.
- **Errors in semantic or word choice.** These errors are to some extent related to ambiguity in word senses and phonetics. For example, it is incorrect to conjugate verbs that belong to the same root but differ in their patterns by mixing up one pattern with another such as incorrectly generating the perfect tense form of the verb ابتاع / {ibotAEa/ (purchased) according to the pattern افتعل /{ifotaEala/ and root ع-ي-ب /b-y-E/ instead of generating it as باع / bAEa/ (sold), which has the intended pattern فعل /faEala/ and root ع-ي-ب /b-y-E/.
- **Errors at the interface of lexical and grammar.** These errors are related to the morpho-syntactic features of words. For example, it is incorrect to negate a verb in its perfect form with the negative particle لم /Lam/ (*not*) unless this verb is in jussive imperfect form such as وجد لم *[2] /lam wajada/ (did-not find) instead of لم يجد /lam ya-jid/ (does-not find).

Existing studies of lexical error analysis fall into two main closely related systems: *Spelling Checkers* and *Intelligent Language Tutoring Systems*. The purpose of most spelling checkers is neither teaching nor learning as they are only designed for detecting spelling errors and suggesting possibly correct spelling (Hsieh et al., 2002). SLLs not only ask for correcting their errors, by just choosing the right word from a list of alternatives, but they also want to improve their language skills in order not to do same errors over and over again. Moreover, most of checkers are inappropriate for nonnative speakers because they are mainly designed for native speakers and as such they are not suitable for detecting and correcting competence errors made by nonnative speakers. For example, recent Microsoft office's Arabic spell checker© detects the word يقالن* / ya-qAlo-na/ as an error but it doesn't suggest the correct form يقلن /ya-qulo-na/ (they-(f) said). On the contrary, ILTS try to overcome these problems and be more useful to SLLs by making true diagnosis[3] of errors. Consequently, they point the learner to the right direction on how to correct their errors rather than providing the correct version directly (Faltin, 2003). However, most ILTS developed until now try to overcome shortcomings of general spell checkers. They do so by incorporating morphological knowledge and non-native intuitions into their algorithms in order to be able to handle competence errors made by nonnative

writers. Therefore, the basic step in any ILTS is to morphologically analyze learner's answer and uses this analysis to make a true diagnosis of the learner's answer.

THE ARABIC MORPHOLOGY SYSTEM

Arabic language is one of the Semitic languages that is defined as a *diacritized* language where the pronunciation of its words cannot be fully determined by their spelling characters only. It depends also on some special marks put above or below the spelling characters to determine the correct pronunciation; these marks are called diacritics, so-called "Tashkil" in Arabic.

Unfortunately, in nowadays Arabic writing, people do not explicitly mention diacritics. They depend on their knowledge of the language and the context to understand none or partial diacitized text. Due to the optional diacritization, two or more words in Arabic are homographic: they have the same orthographic form, though the pronunciation and meaning is totally different (Ahmed 2000; Attia 2006). Table 1 lists some homographic examples.

Arabic language has very rich derivational and inflectional morphology. Al-Sughaiyer et al. (2004) defined derivational morphology as the process of concatenating a set of morphemes to a given word that may affect the syntactic category of the word. While inflectional morphology is the process of creating the various forms of each word. It doesn't affect the word syntactic category such as verb, noun …etc. Features such as case, gender, number, tense, person, mood and voice are some examples that may be affected by the inflectional morphology. The next two subsections present Arabic derivational and inflectional morphology. The last subsection presents an introduction to Arabic verbal system.

Table 1. An Arabic word that is homographic

Word	Lemma	Different Interpretations
يعد / yEd/	أعاد />aEAd/	يعد /yuEid/ (bring back)
	عاد /EAd/	يعُد /yaEud/ (return)
	وعد /waEid/	يعد /yaEid/ (promise)
	عد /Ead~/	يَعُدّ /yaEud~/ (count)
	أعد />aEd~/	يُعِدّ /yuEid~/ (prepare)

Arabic Derivational Morphology

The Arabic derivational morphology has some challenging feature: morphotactic (rules governing which morphemes may come with each other). Whereas most languages construct words out of morphemes which are just concatenated one after another, as in *un-fail* + *ing* – *ly* (بدقة دائمة), Arabic words are derived using three concepts: root, pattern and form. Generally, each pattern carries a meaning which, when combined with the meaning inherent in the root, gives the goal meaning of the inflected form. Although Arabic roots and patterns carry one or more specific meaning, they cannot be an Arabic word on its own. Table 2 illustrates some examples of the derivation process of some Arabic words.

The derivation process in Arabic makes it has the richest vocabulary ever found among all important natural languages although it has a relatively small number of derivative *patterns* (Ahmed 2000). Each pattern is a string of two types of characters: *fixed*, possibly none, and three or four *generic* characters. For example, the pattern تفعيل /tafoEiyl/ has two fixed characters: (ت) /t/ and (ي) /y/ and three generic characters: ف- ع - ل /f-E-l/. Arabic has also a limited number of derivative *roots*. Each root is a set of three or four fixed characters.

Arabic Inflectional Morphology

The Arabic inflectional morphology changes the morpho-syntactic features of the word. It defines the number (singular, plural, and dual), gender

Table 2. The derivation process of some Arabic words

Pattern	Root	Form	Derived Word	Description
فاعل /fAEil/	ك-ت-ب /k-t-b/	The active participle noun (اسم الفاعل) "the doer of the action"	كاتب /kAtib/ (writer)	Both the word form and the root meaning (to write) form together a word that indicates the doer of the action of writing (writer).
مفعول /mafoEuwl/	و-ل-د /w-l-d/	The passive participle noun (اسم المفعول) "the object upon which the action is done"	مولود /mawoluwd/ (new born person)	Both the word form and the root meaning (to give birth) form together a word that indicates being born action (a new born person).
تفعيل /tafoEiyl/	د-ر-س /d-r-s/	The verbal noun (المصدر) "the action of doing something"	تدريس /tadoriys/ (teaching- instruction)	both the word form and the root meaning (to study) form together a word that indicates the action of teaching (teaching or instructing)

(feminine, masculine, and neutral), definiteness (definite, indefinite) and case (nominative, accusative, genitive) features for nouns. While it defines the following features for verbs: tense (prefect, imperfect, imperative), voice (active, passive), mood (indicative, subjunctive, jussive), subject and object (person, number, gender). It does so by adding more affixes to the stem[4] to form a well-formed Arabic word.

Arabic affixes can be prefixes such as ي /ya/ (imperfective subject 3rd person singular), suffixes such as تُ /tu/ (perfective subject 1st person singular) or circumfixes such as ت + + ان /t + + An/ (imperfective subject 2nd person dual). Multiple affixes can appear in a word. For example, the word وسيكتبونها /wasayaktubuwnahA/ (and-they-will-write-it) has two prefixes; one circumfix and one suffix (Habash and Rambow 2006):

- وسيكتبونها /wasayaktubuwnahA/
- و /wa/ + س /sa/ + ي /ya/ + بـتـك /ktub/ + ن و /uwna/ + اه /hA/
- And + will + 3rd person + write + masculine-plural + it

In general, the following can be inferred as a simple structure of the Arabic words (Darwish 2002; Ahmed 2000):

- The main part, a noun or verb, of the word occurs in the middle. Let us call this part as word *stem*.
- The stem may be prefixed by something like the definitive article, a preposition, a tense determiner… etc. or some combination of them. The prefix itself cannot be a standalone word. It may be absent and in this case, we can assume it as a null. When a prefix precedes a stem, it may modify its string and also be modified. For example, the deletion of the letter (و) /w/ from the stem وعد /waEada/ (promised) can be explained by taking the imperfect tense of this stem. The resulting word is يعد /yaEid/ (he promises). Also, the deletion of the letter (ا) /A/ from the prefix ال /Al/ (the) can be explained by adding the preposition ل /l/ (for) to this prefix. The resulting prefix will be لل /lial/ (for-the).
- The stem may be suffixed by something like a pronoun, a gender determiner… etc. or some combination of them. The suffix itself cannot be a standalone word. It may be absent and in this case, we can assume it as a null. When a suffix succeeds a stem, it may modify its string and also be modified. For example, the conversion of letter (ء) /'/ into (و) /w/ in the stem صحراء /SaHorA'/

(desert) can be explained by taking the dual form of this noun. The resulting word is صحراوان /SaHorAwAn/ (two deserts).

Introduction to Arabic Verbal System

One of the most puzzling problems in the study of Arabic is its verbal system which is very rich in forms and meaning (Soudi, Cavalli, and Jamari 2001). Arabic verbs can be conjugated from either trilateral or quadrilateral roots according to one of the traditionally recognized patterns (or forms). There are 15 trilateral forms[5] and 4 quadrilateral ones. Examples of all Arabic forms are shown in Table 3 (Bowden and Kiraz 1995; Wright 1967).

The conjugation of verbs in different tenses, voices and mood is achieved using well behaved morphological rules. The irregularities are due to the phonological constraints of certain root consonants. The important irregularity issues are related to Arabic weak verbs that include one or more weak letter. Weak letters can be deleted or substituted by other letters because of Arabic phonological constraints (El-Sadany and Hashish 1989). For example, the replacement of the letter (و) /w/ by (ا) in taking the past (perfect) tense of the trilateral root ق-و-ل/q-w-l/, using regular rules would generate قَوَلَ* /qa**w**ala/ but as it is a hollow (middle weak) verb it should be generated according to special weak rules and thus it is written as قَالَ /q**A**la/ (said).

To sum up, in this section, we demonstrated the difficulties that can make the process of learning Arabic verbs a difficult one. This has motivated us to address the challenges in developing a morphological analyzer that can handle ill-formed Arabic

Table 3. Arabic verbal forms

Form Number	Pattern	Active Voice Example
1	فعل /faEala/	شرب /$ariba/ (drank)
(a) Trilateral verbs		
2	فعّل /faE~ala/	كسّر /kas~ara/ (shattered)
3	فاعل /fAEala/	ضاعف /DAEafa/ (doubled)
4	أفعل />afEal /	أراح />arAHa/ (brought relief)
5	تفعّل /tafaE~ala/	تعلّم /taEal~ama/ (studied)
6	تفاعل /tafAEala/	تساقط /tasAqaTa/ (collapsed - fall piece by piece)
7	انفعل /{inofaEala/	انكسر /{inokasara/ (broke)
8	افتعل /{ifotaEala/	اقتفى /{iqotafaY/ (follow)
9	استفعل /{isotafaEala/	استغاث /{isotagAva/ (asked for help)
10	افعلّ /{ifoEal~a/	احمرّ /{iHomar~a/ (turned red)
11	افعالّ /{ifoEaAl~/	ازراقّ /{izoraAq~/ (became blue)
12	افعوعل /{ifoEawoEala/	اغرورق /{igoraworaqa/ (immersed)
13	افعوّل /{ifoEaw~la/	اجلوّز /{ijolaw~za/(lasted long)
14	افعنلل /{ifoEanolala/	احلنكك /{iHolanokaka/ (turned jet black)
15	افعنلى /{ifoEanolay/	احبنطى /{iHobanoTay/ (caused swollen)
(b) Quadrilateral verbs		
1	فعلل /faEolala/	دحرج /daHoraja/ (rolled)
2	تفعلل /tafaEolala/	تزلزل /tazalozala/ (quaked)
3	افعنلل /{ifoEanolala/	افرنقع /{iforanoqaEa/ (dismissed)
4	افعلّ /{ifoEalal~a/	اضمحلّ /{iDomaHal~a/ (faded away)

verbs which will be used as a tool in intelligent language tutoring framework to diagnose errors made by SLLs of Arabic.

ARABIC LEXICAL ERROR TYPOLOGY

An important step in the implementation of an error analysis system is to decide which type of errors to be analyzed. Realistically, not every imaginable error type can be analyzed within a single system (Faltin 2003). There are two main criteria to select errors. On the one hand, errors which are easy to implement given the linguistic resources at hand and the diagnosis techniques available. On the other hand, there are the needs of the end-user population which makes specific kinds of errors.

To decide on the set of errors handled by our system, the Arabic SLLs needs were investigated by examining a set of linguistic studies which indicates the most frequent types of errors made by SLLs (cf. Ali 1998; Abd Alghaniy 1998; Jassem 2000). Tables 4 through 5 provide details of possible errors which are commonly made by SLLs of Arabic. These errors are classified as *word formation errors* due to improper application of *morphology* and *phonology.*

However, the proposed system focuses on word formation errors. Other errors (i.e. semantic errors and errors at the interface of lexical and grammar) are out of this paper scope.

Table 4. Word formation errors due to morphology

Error Type	Source of Error
Connected pronouns Acronym: CP	Incorrect usage of pronouns with respect to verb tense **Example:** Wrong: **يجنت** *[6] /ya-ji}o-tu/ (I-he-came) Correct: **جنت** /ji}o-tu/ (I-came)
Verb conjugation Acronym: VC	Incorrect conjugation of Arabic weak verbs **Example:** wrong: ***نجو** /najaw/ correct: **نجا** /najA/ (he escaped)

THE PROPOSED MODEL

The proposed model takes into consideration a set of linguistic studies which follow error analysis approach in identifying most frequent errors made by SLLs. This approach, however, follows some steps to identify and classify errors: data collection, error identification, error classification, error description, error explanation and pedagogical application (Jassem 2000). These studies have acquired real materials written by SLLs in a typical teaching/learning environment; these learners have different backgrounds (i.e., differ in their first language). Consequently, the extracted errors are generally not aimed to a specific sort of learners. Therefore, the proposed model is general enough to be used by different sort of learners.

However, the proposed model generates *all* possible word analyses for each ill-formed learner answer. It uses *constraint relaxation* and *edit-distance* techniques to split each erroneous word into three segments: *prefix*+ *stem*+ *suffix*. In any language model, the partial structures can combine only if some constraints or conditions are met. When these constraints are relaxed, an

Table 5. Word formation errors due to phonology

Error Type	Source of Error
Consonant letters Acronym: CL	Incorrect usage of letters with a closely related pronunciation **Example:** Wrong: ***أستطيع**/>a-SotaTiyE/ Correct: **أستطيع** />a-sotaTiyE / (I-am-able).
Vowel letters Acronym: VL	Making short vowel a long one **Example:** Wrong: ***أصباحت**/>aSobAH-at/. Correct: **أصبحت** />aSobaH-at / (became)
	Making long vowel a short one **Example:** Wrong: ***تزرين** /ta-zuri-yna/. Correct: **تزورين** /ta-zwri-yna/ (you-visit)

attachment is allowed even if the constraint is not satisfied (Faltin 2003).

In Arabic, various constraints should be met to formulate a well-formed word such as *usage of certain connected pronouns with respect to a verb tense* and *the usage of certain affixes or clitics with conjugated verbs*. In the proposed model, these two example constraints can be relaxed to allow for error diagnosis.

In general, the proposed model takes every erroneous input word and proceeds with the following steps to perform its functionality:

1. Extract a list of all possible suffixes.
2. Filter the suffixes list.
3. Extract a list of all possible prefixes.
4. Filter the prefixes list.
5. Construct all possible correct stems.
6. Form groups of similar stems.
7. Get the base word forms[7] from stem strings.
8. Match the correct answer base word form with the learner answer base word forms and determine the analyses of the ill-formed input.

These steps are necessary as the conjugated verb might be made ill-*formed* due to either ill-formed generation of a stem as a result of applying an incorrect pattern to a root or ill-formed inflection of a stem with affixes. The following shows the application of these steps on Example 1.

Example 1. Write a Sentence Using the Following Arabic Roots.

- **ق-و-ل، ح-ق، د-و-م** /q-w-l, H-q, d-w-m/.

Assume the following two answers; where (a) includes a wrong conjugation of a *Hollow* (middle weak) verb, and (b) is the correct answer.

a. **قالتو*** الحق دائما /qAlo-tw AlHaq~ dA}imAF/ (I always told the-truth).

b. **أقول** الحق دائما />a-quwl AlHaq~ dA}imAF/ (I always tell the-truth).

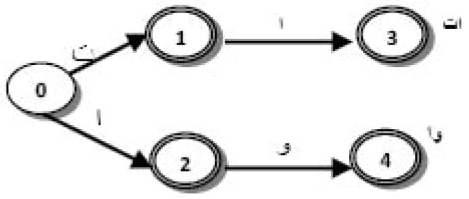
The proposed model first matches the correct answer with the learner answer and filters out the matched words. This leaves the correct answer with the word أقول />a-quwl/ (I-tell) while it leaves the learner answer with the word قالتو /qAlo-tw/ (told-I). Then the model applies all the previous steps on the word قالتو*.

Step 1: Extract a List of All Possible Suffixes

The model uses regular expressions for representing the list of affixes to be extracted. The regular expressions are implemented using the deterministic finite-state automata (FSA) approach. For more information about FSA, see (Jurafsky and Martin 2008). The suffix list is represented in the deterministic FSA in reverse order to facilitate left to right matches. Figure 1 illustrates a FSA representation of the suffixes: وا, ات, ا, ت/Waw-Alef, Alef-Teh, Alef, Teh/.

To extract the suffix list, the system matches the input word against the suffix automata. The match begins at the *end* of the word (Position 1) and works *backwards*. The system relaxes the *usage of certain affixes* constraint by using a *three-way-match* technique (Elmi and Evens 1998) to compare two strings: a suffix of the learner input with a legal suffix. This method assumes

Figure 1. A finite-state automata for four suffixes

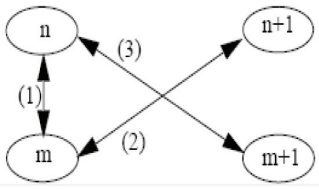


that when a character at location *n* of the first string does not match a character at location *m* of second string, there exist an error and two other comparisons are made (character at position *n* with character at position *m+1* and character at position *n+1* with character at position *m*); Initially, *n=1* to point to the last letter of the input string and *m=0* to point to a letter at the initial state in the FSA. The three-way-match comparison and the order of the comparisons are shown in Figure 2.

Given the FSA at Figure 1 and a learner response with the word وتلاق* /qAlo-tw/ (told-I), the system tries to: 1) match the last letter (*n=1*) و /w/ of the input word with the Arabic suffix ت (Teh) that occurs at the end (*m=1*) of Arabic verbs. The match fails. So, it tries to match again with the one but last letter (*n+1=2*) of the input word ت /t/ which succeeds. This process interprets the letter ت /t/ as a possible suffix and the letter و /w/ as an extra letter occurring at the end of the input word. Similarly, the match exhaustively proceeds with other Arabic suffixes yielding the following 10 possible solutions along with their error indications, respectively:

1. **[“”].** NULL suffix.

Figure 2. The three-way-match comparison and the order of the comparison



Results of Comparison	Comparison number		
	1	2	3
No error	**T**	---	---
Extra character	F	**T**	F
Missing character	F	F	**T**
Character substitution	F	F	F

2. **[“تا”].** Feminine plural noun suffix with extra Waw and missing Alef.
3. **[“ت”].** Third person singular feminine perfect verb suffix with extra Waw.
4. **[“ت”].** First person singular perfect verb suffix with extra Waw.
5. **[“ت”].** Second person singular feminine perfect verb suffix with extra Waw.
6. **[“ت”].** Second person singular masculine perfect verb suffix with extra Waw.
7. **[“و”].** Masculine plural noun suffix.
8. **[“او”].** Second person masculine plural imperative verb suffix with missing Alef.
9. **[“او”].** Masculine plural imperfect verb suffix with missing Alef.
10. **[“او”].** Third person masculine plural perfect verb suffix with missing Alef.

Practically, however, the use of constraint relaxation in analyzing Arabic verbs leads to over-generation. In order to resolve this issue, we introduced *heuristic rules* that eliminate highly implausible analyses made by Arabic SLLs. For example, SLLs of Arabic might find it difficult to choose among a vowel sign such as (قمضلا) /u/ and a genuine character, such as letter و /w/. So, one set of the heuristic rules restricts itself to handle the extra or missing weak letters. Another set of rules restricts itself to recognize a letter substituted by another letter that is similar in pronunciation. We categorized the closely related pronunciation letters into 7 groups: 1) [‘ض’, ‘د’, ‘ت’, ‘ط’], 2), [‘س’, ‘ص’, ‘ز’], 3) [‘ث’, ‘س’], 4) [‘ق’, ‘ج’], 5) [‘ق’, ‘ك’], 6) [‘ظ’, ‘ز’, ‘ذ’], and 7) [‘ح’, ‘ع’].

Step 2: Filter the Suffixes List

This step excludes some irrelevant suffixes according to: learner’s answer and the set of error categories handled by the proposed system. For example, the previous list of 10 suffixes could be minimized to **five** solutions: 1, 4, 8, 9, and 10. There are two explanations behind this filtering. The system does not handle errors related to Arabic

nouns which led to ignore the *second* and *seventh* solutions. Second, the other three eliminated solutions are discarded since their end case does not match the extra character و /w/.

Step 3: Extract a List of All Possible Prefixes

Extracting the prefixes list is the same as extracting the suffixes list except that the order of the match process begins at the *first* letter and proceeds *upwards*. Applying this step on Example 1 produces only null prefix solution:

1. **[“”].** NULL prefix.

Step 4: Filter the Prefixes List

As the prefixes list is null, the output of this step does not result in any filtered prefix.

Step 5: Construct All Possible Correct Stem Forms

To construct a possible correct stem, the system tries exhaustively to extract every possible stem (i.e., a substring that remains after removing prefixes and suffixes from the input word) such that either the compatibility conditions between affixes[8] is satisfied or the relaxed constraints are met. In Arabic, there are certain connected pronouns that can only be used with a certain verb tense. For example, the suffix pronoun ‘نا’ (Na) can only be used with the perfect tense while the prefix pronoun ‘نـ’ (Noon) can only be used with the imperfect tense. It is morphologically incorrect to use both pronouns at the same time (e.g. *‘نذهبنا’) as this will be considered as a severe contradiction in pronoun inflections which leads to a conflict in verb tense. Applying the constraint relaxation technique will split this erroneous word into: the prefix ‘ن,’ the stem ‘ذهب,’ and the suffix ‘نا’ even though the attachment constraint is not met.

The output of this step, using the results so far from Example 1, yields four solutions. Each solution consists of five elements constituting: *prefix*, *stem*, *suffix*, *feature structure (FS)*[9] that describes the analyzed word, and an initial *error indication*. The *error indication* is a list that denotes: the required operation (e.g., insert) to relax the affix, the actual character and the position where the operation should take place.

Solution (1):

- *Null affixes*[10]: Null Prefix + “قالتو” + Null suffix
  Error indication: [];

Solution (2):

- *Null prefix with first person singular perfect verb in active voice with extra Waw in the suffix*:
  Null Prefix “قال”+‘ت’
  Error indication: [insert(‘و,’5)];

Solution (3):

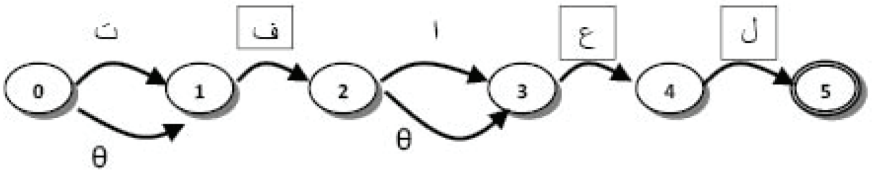
- *Null prefix with second person masculine plural imperative verb with deleted Alef in the suffix*: Null Prefix + “قالت + ‘وا’
  Error indication: [delete(‘ا,’6)];

Solution (4):

- *Null prefix with third person masculine plural perfect verb in active voice with deleted Alef in the suffix*: Null Prefix + 'وا' + "قالت"
  Error indication: [delete(‘ا,’6)]

Notice that, in this example, the solution *“Null prefix with masculine plural imperfect verb with deleted Alef in the suffix”* is discarded as the combination between a null prefix with the imperfect

Figure 3. A finite-state automata for four Arabic verb patterns. The letter inside a square is generic.



suffix 'وا' (Waw & Alef) (cf. the 9th suffix solution in *step 1*) is morphologically invalid (incompatible). This suffix can only be used together with either one of the following prefixes: 'ي' (Yeh) or 'ت' (Teh).

Step 6: Form Groups of Similar Stems

The current solution list may contain similar stem strings. So, in order to avoid redundancy, the list is organized into groups of lexicographically similar stems. The output of this step yields three groups in the solution stem list: {('قالتو', [1]), ('قال', [2]), ('قالت', [3, 4])}; where the number points to the corresponding five elements in the solution list.

Step 7: Get the Base Word Forms From Stem Strings

To get all possible base forms (normalized stems) from each string in the stem solution list, we need first to match with a list of Arabic verb patterns. These patterns are represented as deterministic FSA. Figure 3 illustrates a FSA of the relevant patterns تفاعل, تفعل, فاعل, فعل /tafAEala, tafoE~al, faEala, fAEala/. We differentiate between two types of characters in a pattern: *fixed* and *generic*. A generic character can represent any Arabic letter while a fixed character should represent an exact same character. For example, the pattern فعل / faEala/ has only three generic characters while the pattern تفعل /tafoE~al/ has one fixed character ت /t/ and three generic characters.

The system matches characters of the stem string against characters of the verb pattern using the *three-way-match* technique but to relax only the missing and substituted letters that are similar in pronunciations. If a match succeeds, the resultant word is normalized to get the base form by deleting any weak or hamza letters and; then the obtained form is included in the base form solution list.

This step is applied to the current stem solution list {('قالتو', [1]), ('قال', [2]), ('قالت', [3, 4])}.

The first stem in this list is discarded as it does not match with any Arabic pattern. The processing of the second stem produces the base forms: {'قال' and 'قل'}; which after removing the middle weak letter (i.e. *Alef*) of the first one it becomes normalized to the second (i.e., 'قل'). The processing of the third solution produces the base forms {'قالت' and 'قلت' which is similarly normalized to 'قلت.'

Ultimately, the base form solution list consists of the base forms {('قل', [2]), ('قلت', [3, 4])} (cf. the 2nd, 3rd and 4th solutions in *step 5*).

Step 8: Match the Correct Answer Base Word Form with the Learner Answer Base Word Forms and Determine the Analyses of the Ill-Formed Learner Input

This step obtains first the base word forms of the roots stored with the question. Then, it matches

each of these base forms with each base form in the learner's answer. This process is deterministic such that once a match is found all other forms from the solution list are discarded and the final word analysis is generated.

Applying this step on the base form solution list, the match succeeds with the first base word form[11] (i.e. 'قل'). This yields the final solution "*Null prefix with first person singular perfect verb in active voice with extra Waw in the suffix*" as the only possible word analysis for the erroneous word قالتو* / qAlo-tw / (told-I).

A comparison between the features of the correct word أقول />a-quwl/ (I-tell) and the features derived from the analysis of the incorrect word قالتو* /qAlo-tw/ (told-I) shows that the learner has made three errors:

1. **Verb tense** error since the correct tense is *imperfect* while the incorrect one is *perfect*,
2. **Short vowel substituted by long vowel** error since there is an extra *Waw* character in affix representation, and
3. **Verb conjugation** error since there is an extra character at position 2 of the stem قال /qAla/ and this character does not match the diacritic sign of the correct word which is ضمة /u/.

The system will provide an appropriate feedback describing these errors.

EXPERIMENT

We conducted an experiment that measures how successfully the proposed model generates *all* possible analyses of erroneous Arabic verbs that are used later to diagnose SLLs errors. The *quantitative* measures are used. These measures rely on collecting different test sets written by real SLLs in a typical teaching/learning environment. It was necessary that these learners have different backgrounds (i.e., differ in their first language) to test if the system is general enough and not aimed to a specific sort of learners. The different types of errors and the exact source of errors in the test set are *subjectively* identified by a human specialist to produce the reference set. The test set is then fed into the morphological analyzer and the detected and undetected errors are reported. These errors are based on analyses generated by the proposed model. The recall rate[12] for each error type is calculated. This measure has been used in evaluating similar research (cf. Wagner, Foster, and Genabith 2007; Sjöbergh and Knutsson 2005; Faltin 2003).

The above mentioned methodology is applied on a real test set that consists of 116 real Arabic sentences. The number of words per sentence varies from 3 to 15 words, with an average of 5.1 words per test sentence. The total number of words in all test sentences are 587 words, 118 of them have lexical verb errors. However, 60 of er-

Table 6. Evaluation results

Error Type	N	fully Diagnosed		General Error indication	
		N	%	N	%
CL	8	8	100	0	0
VL	24	19	79.2	5	20.8
VC	21	14	66.7	7	33.3
CP	7	6	85.7	1	14.3
Total	60	47	78.3	12	20

roneous verbs are *word formation* errors. Others are either errors in the word choice or errors at the interface between lexical and grammar which are irrelevant to this paper. Table 6 summarizes the evaluation results.

The last column in Table 6 shows the cases of general error indication (i.e. the system fails to detect the exact source of error the learner made). These cases arose because of ambiguity; the system does not have enough knowledge of what the student meant to express. For instance, consider the erroneous word أجوب. It is not clear whether the learner meant the word to be: 1) the imperfect verb أجيب />u-jiyb/ (I-answer), 2) or imperfect verb أجوب />a-juwb/ (I-explore).

CONCLUSION

Arabic is a highly derivational language that makes it a challenge to SLLs. Therefore, SLLs not only make errors done by native speakers but also others that arise due to competence issues. Consequently, using methods and tools designed for a native speaker spell checking is not a good way to proceed, especially for highly derivational and inflectional languages such as Arabic. Therefore, the nowadays methods and tools should be refined to meet the SLLs needs. In the absence of a complete computationally erroneous Arabic corpus that can be used to evaluate the proposed model, we have to manually collect the test set from the real teaching environment. The test set was relatively small but it was sufficient to show that the approach and techniques employed in this chapter have successfully generated all possible analyses of ill-formed verbs written by SLLs of Arabic, in particular, when it comes to difficult constructions such as Arabic weak verbs. From a pedagogical point of view, the achieved rich analyses enable feedback elaboration that helps learners to understand better their knowledge gab.

REFERENCES

Abd Alghaniy, K. E. (1998). *Arabic and Malaysian languages from phonological and morphological perspective: A contrastive analysis approach*. Master thesis, Cairo University, Egypt.

Ahmed, M. A. (2000). *A large-scale computational processor of the Arabic morphology, and applications*. Master thesis, Cairo University, Egypt.

Al-Sughaiyer, I. A., & Al-Kharashi, I. A. (2004). Arabic morphological analysis techniques: A comprehensive survey. *American Society for Information Science and Technology Journal*, *55*(3), 189–213. doi:10.1002/asi.10368

Ali, M. B. (1998). *Linguistic analysis of mistakes by students at the University of Malaya: An error analysis approach*. Master thesis, Cairo University, Egypt.

Attia, M. A. (2006). An ambiguity-controlled morphological analyzer for modern standard Arabic modeling finite state networks. In *Proceedings of the Challenge of Arabic for NLP/MT Conference,* 2006. The British Computer Society, London.

Beesley, K. R. (2001). Finite-state morphological analysis and generation of Arabic at Xerox Research: Status and plans in 2001. In *Proceedings of the Arabic Language Processing: Status and Prospect,* (ACL 2001). Toulouse, France, (pp. 1-8).

Bowden, T., & Kiraz, G. A. (1995). A morphographemic model for error correction in non-concatenative strings. In *Proceedings of ACL 1995*, Boston, Massachusetts, (pp. 24-30).

Buckwalter, T. (2002). *Buckwalter Arabic morphological analyzer,* version 1.0. Linguistic Data Consortium, University of Pennsylvania, (LDC Catalog No. LDC2002L49). ISBN 1-58563-257-0.

Darwish, K. (2002). Building a shallow morphological analyzer in one day. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages*, (ACL 2002), Philadelphia, PA, USA, (pp. 47-54).

El-Sadany, T. A., & Hashish, M. A. (1989). An Arabic morphological system. *IBM Systems Journal*, *28*(4), 600–612. doi:10.1147/sj.284.0600

Elmi, M. A., & Evens, M. (1998). Spelling correction using context. In *Proceedings of ACL 1998,* Montreal, Canada, (pp. 360-364).

Faltin, A. V. (2003). *Syntactic error diagnosis in the context of computer assisted language learning*. PhD thesis, University of Geneva, Switzerland.

Habash, N., & Rambow, O. (2006). MAGEAD: A morphological analyzer and generator for the Arabic dialects. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics,* Sydney, Australia, (pp. 681–688).

Hsieh, C.-C., Tsai, T.-H., Wible, D., & Hsu, W.-L. (2002). Exploiting knowledge representation in an intelligent tutoring system for English lexical errors. In *Proceedings of the International Conference on Computers in Education ICCE 2002*, Auckland, New Zealand, (pp. 115-116).

Jassem, J. A. (2000). *Study on second language learners of Arabic: An error analysis approach*. Kuala Lumpur, Malaysia: A.S. Noordeen.

Jurafsky, D., & Martin, J. H. (2008). *Speech and language processing: An introduction to natural language processing, computational linguistic and speech processing*. Prentice Hall Series in Artificial Intelligence.

Sjöbergh, J., & Knutsson, O. (2005). Faking errors to avoid making errors: Machine learning for error detection in writing. In *Proceedings of RANLP 2005*, Borovets, Bulgaria, (pp. 506-512).

Soudi, A., Cavalli-Sforza, V., & Jamari, A. (2001). A Computational Lexeme-based Treatment of Arabic Morphology. In *Proceedings of the Workshop on Arabic Language Processing: Status and Prospects,* (ACL 2001), Toulouse, France, (pp. 155-162).

Tschichold, C. (2003). Lexically driven error detection and correction. *CALICO Journal*, *20*(3), 549–559.

Wagner, J., Foster, J., & Genabith, J. V. (2007). A comparative evaluation of deep and shallow approaches to the automatic detection of common grammatical errors. In *Proceedings of EMNLP-CoNLL 2007*, Prague, Czeck Republic, (pp. 112-121).

Wright, W. (1967). *A grammar of the Arabic language* (3rd ed.). Cambridge University Press.

ENDNOTES

1. For transliteration, we refer the reader to Buckwalter (2002).
2. The asterisk indicates an incorrect word or sentence.
3. Faltin (2003) defines diagnosis term as "identification of the cause of error" while correction is "a thing substituted to what is wrong." In the following, we show an example that illustrates the difference between diagnosis and correction. (b) is a possible diagnosis of error of the example (a); while (c) is a correction of it.
   (a) أريد أن أدرس لغة جديدة و لذلك * اختارت أن أدرس العربية.
   (I want to learn a new language so I chose to learn Arabic)
   (b) The Weak letter (ا) /A/ in the hollow (middle weak) verb اختار /{ixotAra/ (chose) cannot be used with the first person suffix pronoun ت /t/ because the last letter in verb ساكن /r/is ر (consonant)

(c) The correct sentence is:
أريد أن أدرس لغة جديدة ولذلك اخترت أن أدرس العربية ;
where the hollow letter is curtailed.

4 The stem is a result of applying some roots into some patterns.

5 The first ninth forms are very common while the rest are very rare.

6 These examples are collected from different real materials which are committed by different Arabic SLLs.

7 A *base word* form is a, normalized, stem form after removing all weak and hamza letters to facilitate the matching of different verb conjugations of the same root without taking into consideration the lexicographic change (i.e. variants) that may happen to these irregular forms.

8 The compatibility table that encodes the relations between prefixes and suffixes is taken from the Buckwalter's Arabic morphological analyzer (Buckwalter 2002).

9 The FS includes the following features: *lexical category*, *pattern*, *tense*, *voice*, *mood*, *subject* and *object person*, *number*, *gender*, which is not shown due to space limitation.

10 This solution represents a perfect verb in the third person singular masculine active voice.

11 The base word form of the correct root ق-و-ل /q-w-l/ is قل after removing the weak letter و /w/.

12 The percentage of each error type in the test set that actually diagnosed by the system.