

Adaptive Feedback Message Generation for Second Language Learners of Arabic

Khaled Shaalan

The British University in Dubai,
PO Box 345015 Dubai, UAE
khaled.shaalan@buid.ac.ae

Marwa Magdy

Faculty of Computers & Information,
Cairo University, 12613 Egypt
m.magdy@fci-cu.edu.eg

Abstract

This paper addresses issues related to generating feedback messages to errors related to Arabic verbs made by second language learners (SLLs). The proposed approach allows for individualization. When a SLL of Arabic writes a wrong verb, it performs analysis of the input and distinguishes between different lexical error types. The proposed system issues the intelligent feedback that conforms to the learner's proficiency level for each class of error. The proposed system has been effectively evaluated using real test data and achieved satisfactory results.

1 Introduction

Second language acquisition is a difficult task. There are various methods to acquire a new language and all of them require some form of feedback, a reaction to what has been said or written. The recent trend is to automate the feedback through Intelligent Language Tutoring System (ILTS).

The current trend concentrates on NLP tools and techniques geared towards the diagnosis of errors produced by SLLs and identifying the cause of their errors rather than providing the correct version directly.

This paper is about the generation of feedback message based on individual proficiency levels. The proficiency level measure is based on the progression in the learner answers. In particular, when a SLL of Arabic writes a wrong verb, it distinguishes between this set of lexical error types: *lexical category selection*, *pattern selection*, *tense selection*, *mood selection*, *subject-verb agreement*, *verb conjugation*, *connected pronouns* and/or *consonant*, and *vowel letters*. Nevertheless, it provides the intelligent feedback that conforms to the learner's expertise for each

class of error. There are three learning levels for each concept covered: *beginner*, *intermediate* and *advanced*. A learner who generally has mastered an Arabic concept might receive a hint just indicating the *class of error*. Whereas, the learner who generally knows the concept but still needs practice in its application the feedback is the *type of the error*. For the beginner learning level, the feedback is as specific as possible, the exact *source of the error* is provided.

The edit distance technique is employed to analyze the erroneous Arabic verb. The deep analysis of the learner input helps in accurately detecting the lexical errors and issuing the appropriate feedback to the learner.

To the best of our knowledge, very few researches has considered true diagnosis and issuing feedback of Arabic lexical errors. For example, Shaalan (2005a; 2005b) has developed an ILTS system for Arabic learners which just embed specific morphological analysis rules to provide feedback. In addition, there exist some systems that are designed for SLLs of other languages than Arabic which still keep the behavior of spell checkers (Faltin et al., 2005; Faltin, 2003; Rimrott, 2003; Hsieh et al., 2002).

The rest of this paper is structured as follows. Section 2 introduces an analysis of Arabic lexical errors. Section 3 describes the proposed model. Section 4 discusses the results. Section 5 gives concluding remarks.

2 Arabic Lexical Error Typology

To decide on the set of errors handled, we investigated the literature which defined the most frequent types of errors made by Arabic SLLs (cf. Ali 1998; Abd Alghaniy 1998; Jassem 2000). These errors can be classified into: *Errors in word formation*, *Errors in semantic or word choice* and *Errors at the interface of lexical and grammar*. Tables 1 through 4 provide details of lexical errors.

Error Type	Source of Error
Verb pattern Acronym: VP	Incorrect usage of root pattern. Wrong: اتوصل*/Âa-tawaS~al/ ¹ (I-arrive). Correct: أواصل /Âu-wASil/ (I-continue)

Table 1: Semantic or Word Choice Errors

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

Table 2: Word Formation Errors due to Morphology

Error Type	Source of Error
Consonant letters Acronym: CL	Incorrect usage of letters with a closely related pronunciation. Wrong: أصتطيع* /Âa-S.taTiyç/ Correct: أستطيع /Âa-s.taTiyç / (I-am-able).
Vowel letters Acronym: VL	Making short vowel a long one. Wrong: أصباحت* /ÂaS.baH-at/ Correct: أصبحت /ÂaS.baH-at / (be-came) Making long vowel a short one. Wrong: تزرين /ta-zuri-yna/. Correct: تزورين /ta-zwri-yna/ (you-visit)

Table 3: Word Formation Errors due to Phonology

3 System Overview

The proposed system is specially designed for individualized SLL of Arabic. The objective test method is used such that the expected learner's answer is relatively short and well-focused². The system contains the following components:

The *lexical error checker* is an NLP component that analyzes the learner's answer and detects possible source of errors. It gets the initial error detection assumptions about each word in the learner answer from the word analyzer mod-

¹ Habash et al. (2007) Arabic transliteration is used here to Romanize Arabic examples.

² There is only one possible correct answer

ule such as the one explained in (Shaalán et al., 2010a; Shaalan et al., 2011).

Error Type	Source of Error
Lexical category Acronym: LC	Switching a conjugated verb with its infinitive, e.g. Wrong: الصلاة* /AlSalAah/ (the-praying) Correct: أصلي /Âu-Sal~iy/ (I-pray)
Verb tense Acronym: VT	Switching an infinitive with its conjugated verb, e.g. Wrong word: بيعت* /biç.-tu/ (I-sold) Correct word: البيع /Albay. ç/ (the-selling)
Disagreement of a connected pronoun with the subject Acronym: SVD	Using incorrect verb tense, e.g. Wrong word: نريد* /nu-riyd/ (we-want) Correct word: أردنا /ÂarAda-nA/ (we-wanted)
Verb mood Acronym: VM	The disagreement may be in gender, number and person disagreement, e.g. Wrong word: ليؤدي* /liyu-wâd~iy/ (to-pray-he) Correct word: لأودي /liÂu-wâd~iy/ (to-pray-I)
	Using incorrect verb mood, e.g. Wrong: يأتي* /ya-Â.tiy/ (he-come [indicative]) Correct: يأت /ya-Â.t/ (he-come [jussive])

Table 4: Errors at the Interface of Lexical and Grammar

This module generates all possible word analyses for each ill-formed input. It uses constraint relaxation and edit-distance techniques to split each erroneous word into three possible segments: *prefix+stem+suffix*. Then the lexical error checker proceeds to detect source of errors using edit distance techniques. *Tutoring module* is responsible for initialization of the student model and issuing appropriate error specific feedback message suited to the learner's expertise level. The proposed system keeps a record of the learner's performance history. This information is held in the *student model*. The *item banking* component contains different types of questions to be issued to the learner.

3.1 Item Banking

The item banking is a database of test items. It includes different types of questions like Dictation, Word order, Build a sentence, Transform a sentence category, Word formation practice and Fill in blank.

Each question is accompanied by an associated list of concepts to test how well the learner has mastered them. Furthermore, each question has some parameters that help the system to diagnose errors. The parameter list is a list of *feature structures* (FSs) for all Arabic words in the correct answer. They include features: *correct word without diacritics*, *correct word with diacritics*, *root*, *pattern*, *type of verb*, *prefix string*, *suffix string*, *lexical category*, *tense*, *voice*, *mood*, *subject*, *object gender*, *number* and *person*.

3.2 Student Model

The student model used here contains only information about the proficiency level of the student. The *perturbation error* model is used to represent this knowledge. In this model, there exist one or more misconceptions for each concept in an introductory course for teaching Arabic weak verbs. For example, the *vowel letters* concept has two associated misconceptions: *make short vowel long one* and the vice versa.

For each concept along with its associated bug, the student model keeps a frequency of this error, to each student, which falls in the range of one of the three learning levels. The frequency of the bug is expressed by a number pair [S, T]; where the variable S represents how many times the student has made this error and the variable T represents the total number of times in which the student has met this concept.

3.3 Lexical Error Checker

This module gets its input from the word analyzer module. The input presents all possible initial analyses for each erroneous word in the learner answer. These analyses consist of five elements: prefix, stem, suffix, FS that describes the analyzed word, and an initial error indication. The later is a list that denotes: the required editing operation (e.g., insert) to the affix string, the actual character and the position where the operation should take place. For example, if the learner writes the wrong verb *قال* /qAl.-tw/ (told-I). The input of the lexical error checker in this case is as follows:

Prefix: Null, **Stem:** "قال", **Suffix:** "ت", **FS:** first person singular per-

fect verb, **Error indication:** [insert('و',5)]

The objective of the lexical error checker is to detect errors in the stem string and to confirm errors in affixes given from the word analyzer module. It contains the following components: error analysis, error classification, filtering module.

3.3.1 Error Analysis Module

This module proceeds with the analysis of all words in the learner's answer. It receives a list containing all possible word analyses from word analyzer module and all possible analyses that have the same root as the correct answer from the morphological analyzer module. And then generate the final analysis of the input words. The following shows how this module works:

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.

- قال^{تو} الحق دائما /qAl.-tw
AlHaq~ dAÿimAã/ (I always told the-truth).
- أقول^ا الحق دائما /Áa-quwl AlHaq~ dAÿimAã/ (I always tell the-truth).

Step1: the morphological analyzer does not result in any solution that has the same root as correct answer ق-و-ل /q-w-l/. However, applying the word analyzer module on the word *قال* /qAl.-tw/ (I-told) results in only one solution, 'first person singular perfect verb active voice with extra Waw in the affix', which becomes the output of step 1.

Step2: The input solution list could be minimized by a number of factors: learner's answer, question parameters, and error categories handled by the system. We derived a set of heuristic rules to discard irrelevant solutions. An example of these rules is given in Table 5.

Step3: For every solution in the list, the system morphologically generates a well-formed stem. A shallow morphological generator is developed that is based on the notion of a Morphological Form Hierarchy (MFH) or tree (Cavalli et al., 2000). The input of this module is a FS.

The transformation rules attached to each leaf node of the MFH effects the desired morphological transformations for that node. The output of the transformation is the *transformed stem* string

from the root string. Figure 1 shows an example of a rule attached to a node in the MFH.

Rule (1): Description
IF the affix string in the learner answer matches the correct affix string AND the FS of the correct answer does not match with the FS of the learner answer
THEN discard this solution.
Example: if the learner writes the word قالت /qAl.tu/ (I-told) instead of قلت /qul.tu/ (I-told), the system will extract four suffixes that have the same orthographic form but differ in their meaning. All these suffixes except 1 st singular suffix are discarded

Table 5: An example of a filtering rule

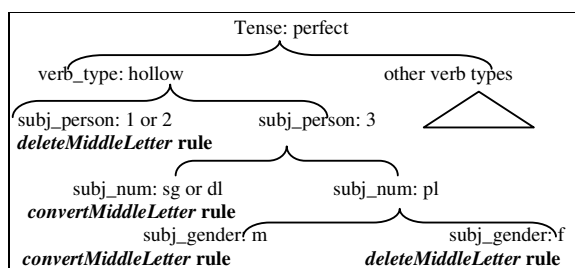


Figure 1: A subtree showing the stem change for perfect verbs of pattern فعل/faEala/

The rationale behind this module is that the specified FS is matched against the features defining each subtree until a leaf is reached. At that point, the transformation rule attached to the leaf node is tried. If no rule is found or none of the clauses of the applicable rule match, it returns the value of *root* unchanged. After applying this step on Example 1, it produces the stem **قُل** /qul/

Step4: In this step, the system matches the generated stem with the extracted (analyzed) one using *three-way-match* method (Elmi and Evens 1998). The inserted and deleted characters are only constraints to be weak letters³. Also, the converted characters should only be performed with another one that has similar pronunciation. The matching process works as follows: partition the two words according to the following patterns the generated stem pattern = *xuz* and extracted stem pattern = *xvz*. Where *x* is the initial segment, *z* is the tail segment; *u* and *v* are the error segments. First, the initial segment is selected. The tail segment is processed likewise. Finally, the error segments are the remaining characters of the two words.

³ This is because the learner may have a problem in either verb conjugation or vowel letters.

Applying this step on Example 1, the extracted stem is **قال** /qAl/ while the generated one is **قُل** /qul/. The matched initial segment is {ق} and the matched tail segment is {ل}. The error segment for the extracted stem is {} whereas it is empty for the generated stem. Therefore, the system concludes that there is some extra character **ا** /A/. This extra character does not match the diacritic sign of the generated word at this position (i.e. the added character is **ا** while the diacritic sign at second position is **ضمة** /u/).

Step5: Ambiguity is a standard problem in any NLP application. In ILTS, relaxing the constraints of the language in order to be able to analyze learner's answer generally produces more interpretations than systems designed for only well-formed input. The ambiguity problem mentioned here is discussed and partially solved in (Shalan et al., 2010b).

3.3.2 Error Classification Module

This module will recognize different error types from word analysis structure. It contains a set of *if-then* rules to recognize different error types. Examples of these rules are given below.

Rule Make Short Vowel long one Error:

IF there is an inserted character in the affix **OR** (there is an inserted character in the stem **AND** this character matches with the diacritic sign at this position of the correct word) **THEN** the error in vowel letters. The parameter of this error is ["short", "long"]

Notice that the learner might make multiple errors in his input. So, this module exhaustively tests *all* IF-statements to detect all possible error types the learner has made.

Applying this module on the input word **قالته** /qAl.tw/ (I-told), it detects that the learner has made three errors: 1) *Verb tense* error since the correct word tense is *imperfect* while the analyzed one is *perfect*, 2) *Make short vowel long one* since there is an extra character in affix, and 3) *Verb conjugation* error since there is an extra character at position 2 in the stem and this character does not match the correct diacritic sign.

3.3.3 Filtering Module

This module accommodates multiple errors, instructional feedback messages need to be prioritized by the system and displayed one at a time to the student to avoid multiple error reports.

The system maintains an error priority queue to rank feedback with respect to the dependency

of errors, e.g. verb tense error has higher priority than verb conjugation error.

3.4 Tutoring Module

This module is responsible for initializing the student model for each new registered student and issuing appropriate error specific feedback message suited to the learner's level. The initialization process is to set the frequency of all bugs in the student model to $[0, 0]$.

The feedback system is responsible for generating feedback messages that conform to the learner's expertise. It includes *error database* and *feedback message generator*. The error database contains a specification of all different errors categories handled by the system.

The feedback message generator module receives a number that defines proficiency level according to this error- beginner or intermediate or advanced. In addition, it receives the error type along with its parameters. Then, it proceeds as follows: for the advanced learning level, the feedback is to provide a hint to the class of the error. For the intermediate, it provides the type of error. For the beginner, the feedback refers to the exact source of the error. For example, the advanced learner will get the following message "error at the interface of lexical and grammar". While the intermediate will get "verb tense error". The beginner message is "incorrect use of perfect verb instead of imperfect"

4 System Evaluation

We conducted an experiment that measures how successfully the proposed model diagnoses errors and provides correct error specific feedback that conforms to the learning level. 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 system and the detected and undetected errors are reported. The recall rate for each error type is calculated.

The above mentioned methodology is applied on a real test set that consists of 116 real Arabic sentences. Table 6 summarizes the evaluation results. The first column in this table describes

the different error types while the second column presents the total number of occurrences of each error type in the test set. The rest of columns present the recall rate of fully diagnosed errors, partially-diagnosed, and general error indication, respectively.

Error Type	N	fully Diagnosed		Partially diagnosed		General Error indication	
		N	%	N	%	N	%
CL	8	8	100	0	0	0	0
VL	24	19	79.2	0	0	5	20.8
VC	21	14	66.7	1	4.8	6	28.6
CP	7	6	85.7	0	0	1	14.3
VP	14	8	57.1	3	21.4	3	21.4
LC	16	14	87.5	0	0	2	12.5
VT	17	11	64.7	0	0	6	35.3
SVD	24	17	70.8	5	20.8	2	8.3
VM	2	2	100	0	0	0	0
Total	133	99	74.4	9	6.8	25	18.8

Table 6: Evaluation Results

Notice, however, the error specific feedback message produced by the system in cases of partially diagnosed errors is the same for both the beginner and intermediate learning level. This is because the source of error was not detected by the system. While the feedback message in cases of general error indication is a catch-all error message for all learning levels.

The highly recall rate is for *consonant letters* and *verb mood* (100%). While the less recall rate is for *verb pattern* (57.1%). This is because of the ambiguity problem. The system has no direct knowledge of what the student meant to express. For example, if the learner writes the word علمت instead of تعلمت /ta-çal~am-tu/ (I-study). It is not clear whether the learner meant علمت /çalim-tu/ (I-knew) by using the pattern فعل /façil/ or علمت /çal~am-tu/ (I-taught) by using the pattern فعل /faç~al/. The system successfully detects that the error type is *verb pattern* but fails to identify the exact wrong pattern. Therefore the feedback message for both beginner and intermediate learner in this case is the same "incorrect use of verb pattern".

5 Conclusion

Learning Arabic language is a challenge because of its complex linguistic structure which poses a difficulty to SLLs. They not only make errors done by native speakers but also others that arise due to competence issues. Our study indicated that using methods and tools designed for a native speaker spell checking is certain to be inadequate.

quate, especially for highly derivational and inflectional languages such as Arabic. Therefore, we adopted methods and tools that meet the SLLs of Arabic needs. Moreover, those learners want to improve their language skills in order not to fall in the same mistakes very often. Therefore, it was appropriate that we developed a diagnosis system, letting the learners find out the correct solution for themselves. Error messages point the learner to the right direction for correction.

In order to evaluate our approach, we acquired a test data set from a real educational SLLs environment. In the absence of a complete computationally erroneous Arabic corpus, either for research or commercial purposes, we only could manually collect a relatively small test set. Fortunately, it was sufficient to show that approach and techniques employed in this paper have successfully analyzed ill-formed verbs written by SLLs of Arabic. Nevertheless, it shows the capability of issuing an intelligent feedback message that conforms to the learner proficiency level allowing the system to perform individualization in the teaching process.

The approach and techniques described in this research can be used with other Semitic languages which share similar morphological features of Arabic to provide appropriate feedback to their SLLs.

References

- Abd Alghaniy, K. E. 1998. Arabic and Malaysian Languages from Phonological and Morphological Perspective: A Contrastive Analysis Approach. Master Thesis, Cairo University, Egypt, 1998.
- Ali, M. B. 1998. Linguistic Analysis of Mistakes by Students at the University of Malaya: An Error Analysis Approach. Master Thesis, Cairo University, Egypt, 1998.
- Cavalli-Sforza, V., Soudi, A. and Mitamura, T. 2000. Arabic Morphology Generation Using a Concatenative Strategy. In Proceedings of the 1st Conference, NAACL 00. Seattle, Washington, pp: 86-93.
- Elmi, M. A. and Evens, M. 1998. Spelling Correction Using Context. In Proceedings of 36th ACL 98, Montreal, Canada, pp: 360-364.
- Faltin, A. V., L'haire, S. and Ndiaye, M. 2005. A Spell Checker for Language Learners of French and a Learner Corpus. In Proceedings of EUROCALL 05. Cracow, Poland.
- Faltin, A. V. 2003. Syntactic Error Diagnosis in the Context of Computer Assisted Language Learning. PhD Thesis, University of Geneva, Switzerland, 2003.
- Habash, N., Soudi, A., and Buckwalter, T. 2007. On Arabic Transliteration. In Arabic Computational Morphology: Knowledge-based and Empirical Methods. Soudi, Abdelhadi; van den Bosch, Antal; Neumann, Günter (Eds.), 2007. ISBN: 978-1-4020-6045-8
- Hsieh, C.-C., Tsai, T.-H., Wible, D. and Hsu, W.-L. 2002. Exploiting Knowledge Representation in an Intelligent Tutoring System for English Lexical Errors. In Proceedings Of 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. ISBN 983-065-093-6.
- Rimrott, A. 2003. SANTY: A Spell Checking Algorithm for Treating Predictable Verb Inflection Mistakes Made by Non-Native Writers of German. Term Paper for LING 807 – Computational Linguistics at Simon Fraser University (Burnaby, Canada).
- Shaalán, K., Magdy, M., Fahmy, A. 2011. Morphological analysis of ill formed Arabic verbs for second language learners. In McCarthy, P.M & Boonthum, C. (ed.), Applied Natural Language Processing and content analysis: Identification, Investigation, and Resolution (In Press).
- Shaalán, K., Magdy, M., Fahmy, A. 2010a. Morphological Analysis of Ill-formed Arabic Verbs in Intelligent Language Tutoring Framework. In the Proceedings of FLAIRS-23, Applied Natural Language Processing Track, Florida, USA, 2010.
- Shaalán, K., Samy, D. and Magdy, M. 2010b. Towards Resolving Morphological Ambiguity in Arabic Intelligent Language Tutoring Framework. In Proceedings of International Workshop on Supporting e-Learning with Language Resources and Semantic Data (LREC 2010), Valletta, Malta
- Shaalán K. 2005a. An Intelligent Computer Assisted Language Learning System for Arabic Learners, in Computer Assisted Language Learning: An International Journal, Taylor & Francis Group Ltd., 18(1 & 2): 81-108.
- Shaalán K. 2005b. Arabic GramCheck: A Grammar Checker for Arabic, Software Practice and Experience, John Wiley & sons Ltd., UK, 35(7):643-665.