Computer Assisted Language Learning
Vol. 18, Nos 1 & 2, February 2005, pp. 81–108

An Intelligent Computer Assisted Language Learning System for Arabic Learners

Khaled F. Shaalan*[1]
The British University in Dubai (BUiD), UAE

This paper describes the development of an intelligent computer-assisted language learning (ICALL) system for learning Arabic. This system could be used for learning Arabic by students at primary schools or by learners of Arabic as a second or foreign language. It explores the use of Natural Language Processing (NLP) techniques for learning Arabic. The learners are encouraged to produce sentences freely in various situations and contexts and guided to recognise by themselves the erroneous or inappropriate functions of their misused expressions. In this system, we use NLP tools (including morphological analyser and syntax analyser) and error analyser to issue feedback to the learner. Furthermore, we propose a mechanism of correction by the learner which allows the learner to correct the typed sentence independently, and allows the learner to realise that what the error is.

Introduction

Computer-assisted language learning (CALL)[2] addresses the use of computers for language teaching and learning. CALL emerged in the early days of computers. Since the early 1960s, CALL software was designed and implemented. The effectiveness of CALL systems has been made obvious by many researchers (Lam & Pennington, 1995; McEnery, Baker, & Wilson, 1995). Until quite recently, computer-assisted language learning was a topic of relevance mostly to those with a special interest in that area. Recently, though, computers have become so widespread in schools and homes and their uses have expanded so dramatically that the majority of language teachers must now begin to think about the implications of computers for language

*Corresponding author. Institute of Informatics, The British University in Dubai (BUiD), PO Box 502216, Dubai, UAE. Email: khaled.shaalan@buid.ac.ae

ISSN 0958-8221 (print)/ISSN 1744-3210 (online)/05/010081–28

DOI: 10.1080/09588220500132399

learning. Using computers provides a number of advantages for language learning (Warschauer, 1996):

- Repeated exposure to the same material is beneficial or even essential to learning.
- A computer is ideal for carrying out repeated drills, since the machine does not get bored with presenting the same material and since it can provide immediate non-judgmental feedback.
- A computer can present such material on an individualised basis, allowing students to proceed at their own pace and freeing up class time for other activities.
- The process of finding the right answer involves a fair amount of student choice, control, and interaction.
- The computer can create a realistic learning environment, since listening can be combined with seeing, just as in the real world.
- Multimedia and hypermedia technologies allow a variety of media (text, graphics, sound, animation, and video) to be accessed on a single machine. Hence, skills are easily integrated, since the variety of media makes it natural to combine reading, writing, speaking and listening in a single activity.
- Internet technology facilitates communications among the teacher and the language learners. It allows a teacher or student to share a message with a small group, the whole class, a partner class, or an international discussion list of hundreds or thousands of people.
- Incorporating NLP techniques provide learners with more flexible—indeed, more 'intelligent'—feedback and guidance in their language learning process.

More than a decade ago, Intelligent Computer-Assisted Language Learning (ICALL) started as a separate research field, when Artificial Intelligence (AI) technologies were mature enough to be included in language learning systems. The beginning of the new research field was characterised by Intelligent Tutoring Systems (ITS), which embedded some NLP features to extend the functionality of traditional language learning systems. The continuous advances in ICALL systems have been documented in several publications (Cameron, 1999; Gamper & Knapp, 2002; Holland, Kaplan, & Sama, 1995; Swartz & Yazdani, 1992).

By far the majority language learning programs have been developed for English, followed by Japanese, French, and German (Gamper & Knapp, 2002). Current Arabic (I)CALL systems have the weakness that learners cannot key in an Arabic sentence freely. Similarly, they cannot guide the learner to correct the most likely ill-formed input sentences. The learner just accepts the information which follows the programmed instruction that is pre-installed in the computer. For these systems to be useful, more research to combine NLP techniques with language learning systems is needed. Parsing, the core component in ICALL systems, allows the system both to analyse the learner's input and to generate responses to that input (Holland, Maisano, & Alderks, 1993). Allowing learners to phrase their own sentences freely without following any pre-fixed rules can improve the effectiveness of ICALL systems, especially when the expected learner answers are relatively short and well-focused (Boytcheva, Vitanova, Strupchanska, Yankova, & Angelova, 2004). Both the well- and ill-formed structure of the input sentence can be recognised. The learner should be allowed to correct the typed sentence independently.

This paper describes an ICALL system for Arabic using NLP techniques, called *Arabic ICALL*, which can solve the weaknesses of current Arabic (I)CALL systems. In Arabic ICALL, there are two main types of test items for interaction with the learner—*selection-type* that tends to elicit answers easily classified as right or wrong and *supply-type* requiring the learners to write a few words. The objective test method is used to assess the learner's knowledge or skills where each question has one (and only one) correct answer—and there is no ambiguity about what that correct answer should be. The present system guides learners to recognise by themselves the erroneous or inappropriate functions of their misused expressions. In other words, it helps learners to make use of their errors. It doesn't give them the correct answer directly but it enables them to try over and over again. In this system, we use NLP tools (including morphological analyser and syntax analyser) and error analyser to issue feedback to the learner. Furthermore, we propose a mechanism of correction by learners which allows the learner to correct the typed sentence independently, and allows learners to realise what the error is. Arabic ICALL follows the curriculum of Arabic grammar at the Egyptian primary schools.

The rest of this paper is structured as follows. First, related work on Arabic language learning programs is given. We also discuss limitations of current Arabic language learning systems as well as limitations of the Arabic ICALL system, and briefly describe our proposed Arabic ICALL system. The sections following on from this present the main components of the Arabic ICALL system. Finally, we conclude the paper and give directions for future work.

Related Work

The linguistic computation of an Arabic sentence is a difficult task (Othman, Shaalan, & Rafea, 2003). The difficulty comes from several sources: (1) the length of sentences and the complexity of Arabic syntax; (2) the omission of diacritics (vowels) in written Arabic "altashkiil"; (3) the free word order nature of Arabic sentence; and (4) the presence of an elliptic personal pronoun "alDamiir almustatir". For these reasons, there is very little research involving Arabic (I)CALL (Ditters, Oostdijk, & Cameron, 1993).

Research into Arabic (I)CALL can be classified by two approaches: the *Computer as a tool* and the *Computer as a tutor.* In the *Computer as a tool* approach, some computer programs can be used as a tool that does not necessarily provide any language material at all, but rather empowers the learner (usually a native speaker) to use or understand language. In the *Computer as a tutor* approach, the process of finding the right answer involves a fair amount of student choice, control, and interaction.

Computer as Tool

Hegazi, Ali, Abed and Hamada (1989) presented a way of representing Arabic syntax in Prolog as production rules. The system can detect some errors concerning Arabic syntax, and so can be used for an educational environment.

Abou Ela (1994) developed an expert system, the Arabic Syntax Analyzer (ESASA), which can be used as a tool to assist Arabic linguists in building Arabic grammar rules. The grammar is expressed using a declarative language called Grammar Writing Language (GWL). This tool is aimed at building Arabic natural language applications including CALL.

Using the Internet for publishing web-based CALL materials that contain non-Latin alphabets requires the solution of various technical problems. There are so many unknown factors associated with the operating system of a distant user that affect the browsing characteristics of these materials. Cushion and Hémard (2002) described how recent technological developments have provided the possibility of overcoming these technical problems in conjunction with the Java programming language and the Unicode character numbering system.

Shaalan (2003) developed an Arabic grammar checker, called Arabic GramCheck. Arabic GramCheck looks for common Arabic grammatical problems, describes the problem, and offers suggestions for improvement. This program is useful in pointing to problems believed typical of native speaker writing. Thus, the learner can avoid such problems in future.

Computer as Tutor

Gheith, Dawa, and Afifty (1996) developed Instructional Software for Teaching the Arabic Language (ISTAL) for grade one prep school. The system presents the curriculum as a simple concept associated with a set of generated sentences. Then, the system generates an exercise for the student and the student's answer is automatically evaluated by comparing it to the system's solution.

Recently Nielsen (2001) and Nielsen and Carlsen (2003) developed a system for learning Arabic, ArabVISL, at the University of Southern Denmark. The system is an interactive Web-based application. It allows students of Arabic as a foreign language to analyse Arabic sentences by using Arabic script and specific Arabic grammatical terminology.

The Interactive Language Learning Project at London Guildhall University has produced course materials for the University's Arabic classes (Cushion & Hémard, 2003). The system is designed for learning Arabic at the beginner level. This study focused on problems associated with learning a language with an unfamiliar alphabet. It discussed the possible use of CALL authoring as part of the learning process.

Mote, Johnson, Sethy, Silva, and Narayanan (2004) developed a speech-enabled computer learning environment designed to teach Arabic spoken communication to American English speakers, called Tactical Language Training System (TLTS). This system can detect errors in learner speech. The TLTS incorporates two speech-enabled learning environments: an interactive game called the Mission Practice Environment (MPE) that simulates conversations with native speakers, and an intelligent tutoring system called the Mission Skill Builder (MSB) for acquiring and practising communicative skills.

Limitations of Current Arabic Language Learning Systems

In Egypt, where there is a growing demand in using computers for teaching and learning, some publishers of off-the-shelf school textbooks provide students with either CD's or web sites that contain vocabulary and grammar practice. However, most of these systems have some common limitations, which are:

1. They often resemble the traditional workbook exercises from which they were adapted.
2. From a pedagogical perspective, the definition of acceptable answers to exercises is highly constrained. For instance, in the linguistic analysis (إعراب) questions, the learner can type his answer as follows: ‘‘مبتدأ مرفوع وعلامة رفعة الضمة’’ (inchoative is in nominal case and the diacritic sign is dam-mah). Nevertheless, the system would consider this response as a wrong answer since it stores the answer of this question as: ‘‘مبتدأ مرفوع بالضمة’’ (inchoative is in nominal case and the diacritic sign is dam-mah).
3. Error feedback commonly does not address the source of an error. For instance, the system displays the correct answer without any explanation of the student's mistake. This makes the system's feedback a generic catchall response.
4. For vocabulary exercises, the student is referred to the corresponding page in the textbook, which displays the word in question in a word list. In addition to the pedagogical limitations, the student has to consult the textbook, which is an unnecessary inconvenience given the potential of the Web.

Limitations of Arabic ICALL

Arabic ICALL system has been successfully implemented using SICStus Prolog on an IBM PC. The system has some limitations:

- The system as described is targeted at a particularly well-formed subset of Arabic, which would not extend well to more colloquial dialects. Even standard newswire is likely to frequently include pre-verbal subjects and adverbials which are not considered in this work. This restriction to a well-formed subset might be appropriate for people trying to learn Arabic in a formal style.
- As vowels are usually omitted in written Arabic, our system does not handle the vowled Arabic text where letters are written with diacritic signs.
- Although ordering words to form a sentence is a type of question normally classified under the objective test method, it is not included into our system due to the free word order nature of Arabic that is usually dependent on semantics.

- Since the task of automatic processing of free natural language in ICALL is hard, the objective test method is used such that the expected learner's answer is relatively short and well-focused.
- The present system does not diagnose spelling errors. It accepts only answers that are free of typographical errors. We have designed, but only partially implemented due to lack of time and fund, an Arabic Spell Checker (Shaalan, Allam, & Gomah, 2003). As the Arabic Spell Checker is not fully implemented, it was not integrated with Arabic ICALL. To be part of the final Arabic ICALL's error analyser, this integration should distinguish typographical (misspelling) errors from wrong answers.

The Proposed System Architecture

Figure 1 shows the overall architecture of the proposed Arabic ICALL system. This system consists of the following components: user interface, course material, sentence analysis, and feedback. The user interface provides the means of communications between the learner and the Arabic ICALL system. The course material includes educational units, an item (question) bank, a test generator, and an acquisition tool. The sentence analysis includes a morphological analyser, syntax analyser (parser), grammar rules, and lexicon. The feedback component includes an error analyser that is used to parse ill-formed learner input and to issue feedback to the learner.

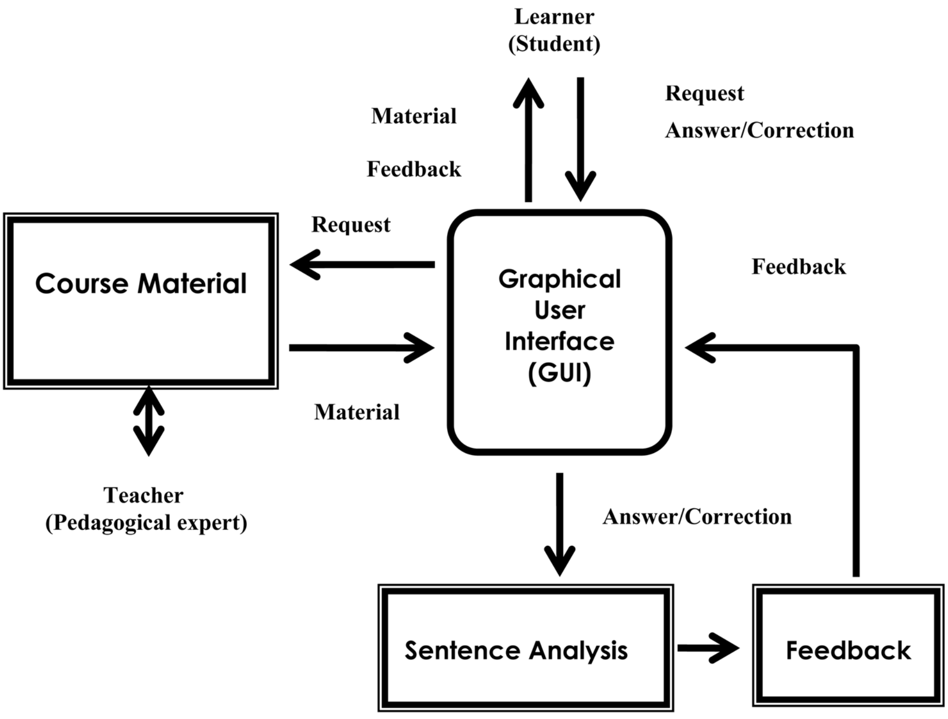


Figure 1. Overall architecture of the proposed Arabic ICALL system

User Interface

The user interface provides the means of communications between the learner and the Arabic ICALL system. It is used to present multimedia lessons (text, graphics, sound, and animation), to present the test items, to allow the learner to take the test, and to present feedback to the learner.

In (I)CALL, there are three possible approaches for reacting to the learner's response in order to give appropriate feedback to the learner: *pattern matching-based approach, statistical-based approach,* and *rule-based approach.*

The *pattern matching-based approach* requires that exercise authors enter many different correct and wrong answers with their associated feedback. This is a very tedious and time consuming task yet only provides appropriate feedback when the learner types in one of the expected answers. Thus, this approach requires a great deal of up front teacher knowledge, experience, and effort.

Both *statistical-based approach* and *rule-based approach* give some freedom to the language learners in the way they phrase their answers, while enabling the exercise author to enter only one possible correct answer, thus saving much time compared to the previous pattern matching answer coding approach.

The *statistical-based approach* uses statistical methods to acquire knowledge. Parameters are automatically learned (estimated) from a corpus that is labeled with the properties needed. Both a statistical model and language parameters should be specified by humans. The characteristics of the statistical-based approach are: (1) no strict sense of well-formedness in mind; and (2) have a large parameter space (e.g., 100,000 words using Tri-gram model requires $10^5*10^5*10^5$ parameters). The advantages of this approach are that it does not need computational models to be established and computation is easy.

The disadvantages of this approach are: (1) cannot restrict computation using heuristics of the linguistic theory; (2) requires a large amount of data to train the statistical model. Parsers and error diagnosis tools cannot be trained on raw data. The data must be tagged by humans, which is costly, time consuming, and sometimes ambiguous. Meaning, hand-tagged corpora is very expensive; (3) even ungrammatical permutation (i.e., sequence) of words of the ill-formed learner's answer are still probable (i.e., have some probability to occur). It is hard to decide whether a string of words is grammatical or not. Actually, strings that are not words at all still will have some probability in the mass probability of the model; (4) the large parameter space of statistical models is a serious problem when decoding (i.e., searching for the most probable structure to assign to a string of words). Statistical models do not distinguish between different forms of words, for example, play, plays, played, playing are not treated as related words originating from the verb play.

The *rule-based approach* provides detailed analysis of the learner's answer using linguistic (morphological and syntactic) knowledge. The characteristics of the rule-based approach are: (1) has a strict sense of well-formedness in mind; (2) imposes linguistic constraints to satisfy well-formedness; (3) allows the use of heuristics (such as a verb cannot be preceded by a preposition); and (4) Relies on hand-constructed rules rather than automatic training from data. The advantages of this approach are that it is easy to incorporate the linguistic knowledge, and it is easy to augment the grammar rules with heuristic rules, which are capable of detecting ill-formed input and providing appropriate feedback. The disadvantage of this approach is that it is not easy to obtain high coverage (completeness) of the linguistic knowledge. However, it could be useful for limited domain where errors in the input can be expected.

It is well-established that feedback is an essential prerequisite for effective learning. Both the pattern matching-based approach and statistical-based approach lack a systematic and automatic way in diagnosing the learner's ill-formed input and providing appropriate feedback. Data collection is costly and time consuming. On the contrary, the rule-based approach has the advantage of providing appropriate feedback because it performs detailed analysis for both well-formed and ill-formed answers. It is easy to acquire linguistic knowledge, and to specify linguistic constraints and heuristics. For these reasons, we decided to follow the rule-based approach in developing Arabic ICALL.

Course Material

The course material includes educational units, item bank, test generator, and acquisition tool (see Figure 2). Each educational unit is a collection of Arabic grammar lessons that addresses a common topic. The item bank (question bank) is a

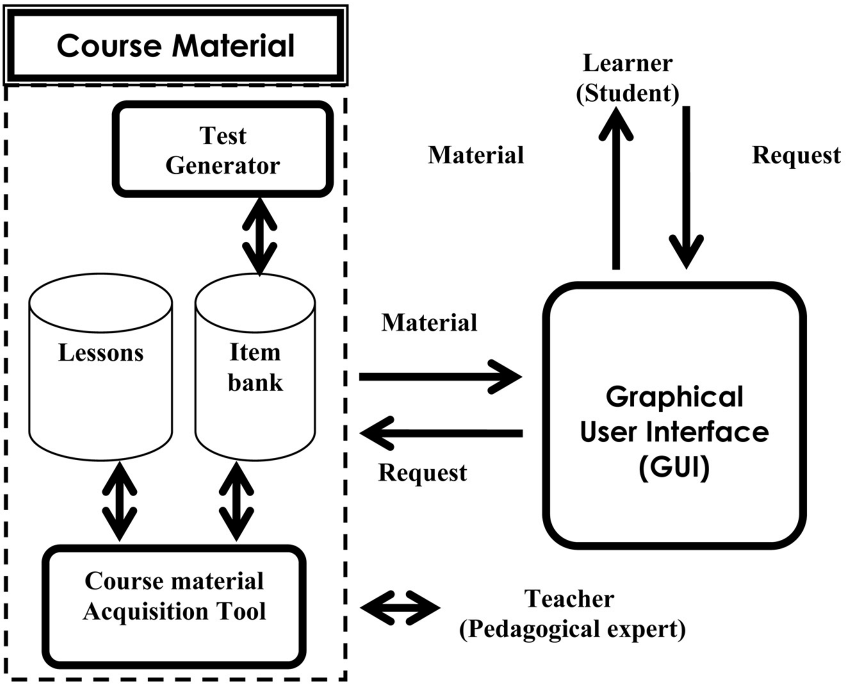


Figure 2. The proposed course material architecture

database of test items. The test generator withdraws test items as needed to develop a test. The acquisition tool allows the instructor to author and maintain lessons, and to create and maintain test items.

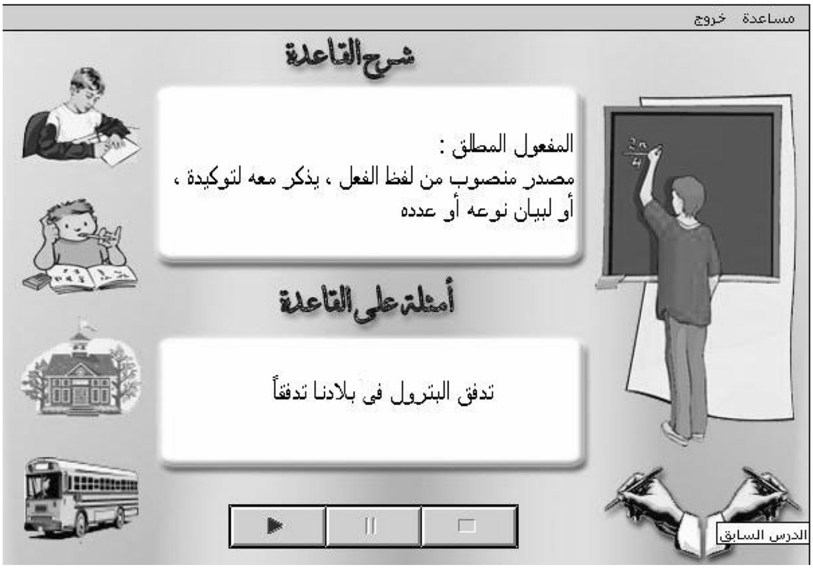
Primary Level Lessons of Arabic Language

The educational units include Arabic grammar lessons for the primary level. Specifically, they cover the following:[3]

> الأسماء (nouns) – النعت (adjective) – المثنى والجمع (dual and plural) – الأفعال(verbs) – الضمائر (pronouns) – الجملة الأسمية (nominal sentence) –المبتدأ والخبر (inchoative and enunciative) – أنواع الخبر (types of the enunciative) – الجملة الفعلية (verbal sentence) – العطف (conjunctions) – الظرف (adverb) – الحال (circumstantial accusative) – النداء (interjection) – المفعول لأجله (causative object) – المفعول المطلق (unrestricted object) – إن وأخواتها (Inna and her sisters) – أنواع خبر إن وأخواتها (types of predicate of Inna and her sisters) – كان وأخواتها (Kana and her sisters) – أنواع خبر كان وأخواتها (types of predicate of Kana and her sisters) – الحروف (articles)

Figure 3 shows an example of a lesson explaining the unrestricted object "المفعول المطلق". It consists of an explanation of this grammar rule, an example, sound functionality, lesson test, and some navigation aids.

The lessons are stored in a database. The system includes some instructional templates to allow for quick generation of instructional material. The structure of lessons consists of two database relations, namely, lesson relation and example relation.



شرح القاعدة

المفعول المطلق :
مصدر منصوب من لفظ الفعل ، يذكر معه لتوكيدة ،
أو لبيان نوعه أو عدده

أمثلة على القاعدة

تدفق البترول فى بلادنا تدفقاً

Figure 3. A lesson

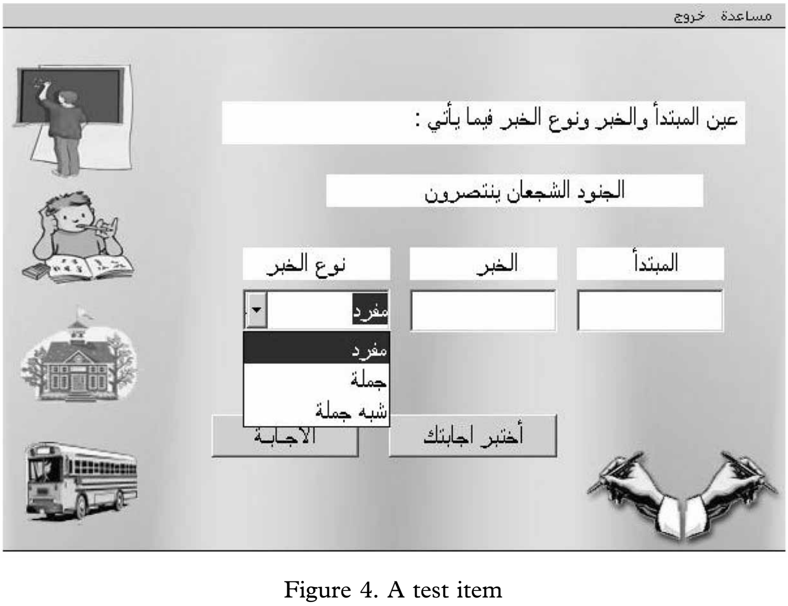
Item Bank

The item bank is a database of test items. This component is used to generate different types of test items each time the learner is allowed to take a test. The test generator selects test items in random order. The instructor determines the selection criterion, and all the test items that match this criterion are collected. Then, we apply a random function to present the selected test items to the learner.

Figure 4 shows an example of a test item. It consists of an explanation of a question header (identify the inchoative and enunciative, and the type of the enunciative in the following), a given sentence (the brave soldiers fought to victory), a learner input area (a pull down menu and two text boxes), an ‘‘Answer button’’ to generate the model answer, a ‘‘Check button’’ to check the learner’s answer, a hyperlink to the relevant grammar lesson, and some navigation aids. It is worth noting that the learner’s answer, whether correct or wrong, is compared against the system-generated answer.

In Arabic ICALL, there are two main types of test items for interaction with the learner: supply-type (short-answer/fill-in-the-blank) or selection-type (matching, true/false, identify, or multiple-choice) interactive questions. The objective test method is used to assess the learner’s knowledge or skills. From the linguistic point of view, the type of questions used in our Arabic ICALL system can be classified as follows:

1. Identify words according to certain morphological features or identifying constituents according to certain syntactic features
   - Examples:
   - Identify the verb, subject, and object in the following sentence
     ‘‘عين الفعل والفاعل والمفعول به في الجملة الآتية’’



عين المبتدأ والخبر ونوع الخبر فيما يأتي :

الجنود الشجعان ينتصرون

Figure 4. A test item

- Extract the adjective and the described noun in the following sentence
  ”أستخرج النعت و المنعوت في الجملة الآتية:“

2. Verb conjugation
   - Examples:
   - Give the correct present and imperative tense of the following verbs
     ”أكتب الفعل المضارع والأمر للأفعال الآتية “
   - Present tense—fill in the blank with the correctly conjugated form of the weak verb in parentheses
     ”الفعل المضارع – أكمل باستخدام الفعل المعتل المناسب مما بين القوسين“
3. Noun morphology
   - Examples:
   - What pronoun would you use to talk to the following people?
     ”ما هي الضمائر التي يمكن استخدامها مع الأشخاص التاليين“
   - Fill in the blanks with the correct form of the demonstrative noun
     ”أكمل باستخدام أسم الإشارة المناسب“
4. Identify the grammatical relation
   - Examples:
   - Identify the negation or prohibition in the following sentence
     ”عين النفي أو النهى في الجملة الآتية:“
   - Put the connective particle in the correct place in the following sentences
     ” ضع حرف العطف في مكانه الصحيح من الجمل الآتية:“
5. Linguistic analysis of words between brackets or a sentence
   - Examples:
   - What's the difference in linguistic analysis between the following? Give the reason ”هل تري فرقا إعرابيا في التالي؟ اذكر السبب“
   - Give the reason for the accusative end case of the words between brackets
     ” اذكر سبب نصب الكلمات التي بين القوسين:“
6. Transform the sentence category
   - Examples:
   - Change the following nominal sentence into verbal sentence
     ”الجملة الآتية أسمية أجعلها فعلية“
   - Advise your colleague of the following using imperative verb
     ”انصح زميلك بما يلي مستخدمًا فعل أمر:“
7. Agreement
   - Examples:
   - Rewrite the following sentences, changing the demonstrative noun from the singular form into the plural form, and change what is necessary to make it grammatically correct sentence
     ”أجعل الإشارة للجمع وغير مايلزم“
   - Is the agreement between the adjective and the described noun in the following sentence correct or incorrect? Give the reason
     ”هل التطابق بين النعت والمنعوت في الجمل الآتية صحيح؟ اذكر السبب “
8. Review test

- Examples:
- Complete the following passage by selecting the correct verb/correct adjective for the context
  ”أكمل القطعة التالية باختيار الفعل أو النعت المناسب للسياق“
- Are the following sentences grammatically correct?
  ”هل الجمل التالية صحيحة إعرابيا“

The structure of the item bank consists of three database relations, namely, question title relation, question content relation, and answer relation.

Sentence Analysis

Logic programming plays an essential role in NLP because it attempts to use logic to express grammar rules and to formalise the process of parsing (Gazdar & Mellish, 1990). A grammar specified this way is known as *logic grammar* since it represents rules as Horn clauses (Dougherty, 1994). Logic grammars can be conveniently implemented in Prolog. Prolog-based grammars can be quite efficient in practice (Allen, 1995). The Prolog interpretation algorithm uses exactly the same search strategy as the depth-first top-down parsing algorithm, so all that is needed is a way to reformulate grammar rules as clauses in Prolog. Definite clause grammars (DCGs) notation was developed as a result of research in natural language parsing and understanding (Pereira, Sheiber, & David, 1986). DCGs allow one to write grammar rules directly in Prolog, producing a simple recursive descent parser. During the construction of the Arabic parser, feature-structures are translated into Prolog terms. Because of this translation step, parsing can make use of Prolog's built-in term-unification, instead of the more expensive feature-unification. Prologs that conform to the Edinburgh standard have DCGs as part of their implementations. In the current system, grammar rules of Arabic are written in the DCG formalism, which is automatically translated into executable code in SICStus Prolog.

The sentence analysis in Arabic ICALL includes a morphological analyser, syntax analyser (parser), grammar rules and lexicon (see Figure 5). The sentence analysis works as follows. Learner written input is first fed into the interactive preprocessor, where it is grouped into words. The words in the input are then decomposed into roots and affixes by the morphological analyser, which obtains information about the subparts from the lexicon (for example, part of speech, number, case). The subparts so identified are then reunified into whole words and passed along to the syntactic parser. The Arabic parser, which is based on DCG formalism, tries to build a structure (usually, “parse tree”) based on the information from the lexicon concerning the grammatical relations between the words. The parser then applies a set of descriptive rules representing the grammar of Arabic until it finds the structure represented by the input sentence. This structure is passed to the feedback component that is equipped with an error analyser that identifies and records any errors made in the structural description that is generated.

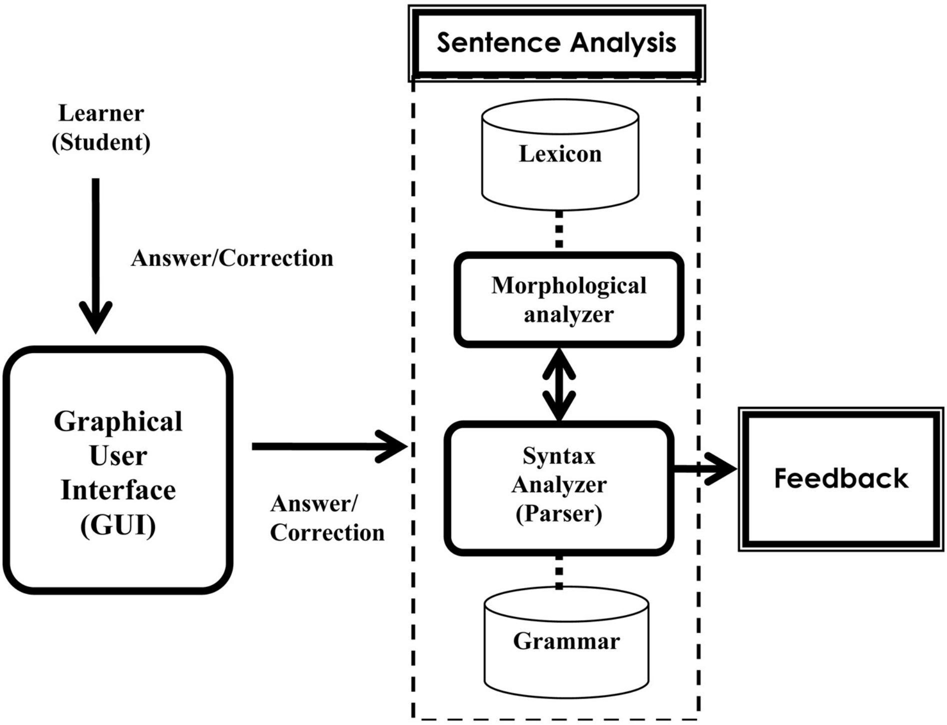


Figure 5. The proposed sentence analysis architecture

Morphological Analysis

The Arabic language is based on the Semitic root-and-pattern scheme of forming word roots, as well as the concatenation of root and affixes. We need a sophisticated morphological analyser that is capable of transforming the inflected Arabic word into its origin. To achieve this function we developed a morphological analyser for inflected Arabic words (see Rafea & Shaalan, 1993). The morphological analyser analyses the inflected Arabic word to extract the root and its features. An Augmented Transition Network (ATN) (Woods, 1970) technique was successfully used to represent the context-sensitive knowledge about the relation between a root and inflectional additions. The ATN consists of arcs. Each of which is a link from a departure node to a destination node, called states (see Figure 6). ATNs additionally employ *registers* which hold linguistic information. ATNs also allow actions to be associated with each arc, for instance, the setting of the register with an omitted affix, the conversion/addition of a weak letter. An exhaustive-search to traverse the ATN generates all the possible interpretations of an inflected Arabic word. The morphological analyser is implemented in Prolog and integrated with the Arabic DCG parser.

Figure 7 shows an example of analysing the verb “شاهدتك” (I saw you) using the ATN shown in Figure 6. This verb is broken down into the verb “شاهد” (saw), the first person pronoun “ت”, and the second person pronoun “ك ”.

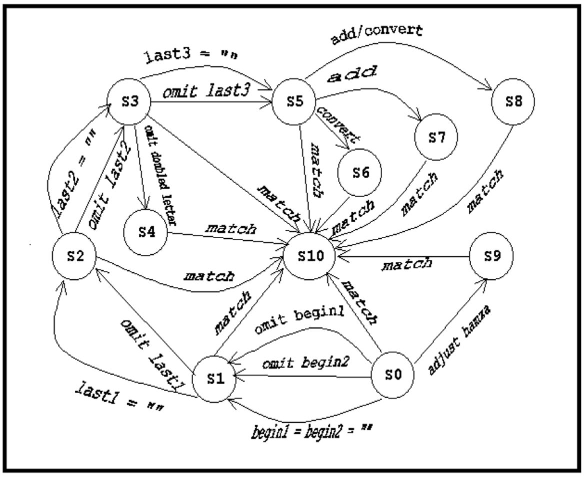


Figure 6. ATN representing the relation between the additions and root of an inflected Arabic word

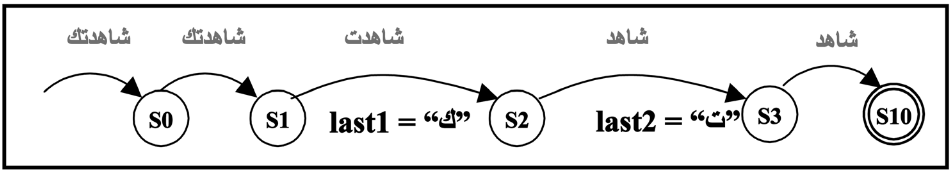


Figure 7. Analysis of the Arabic verb "شاهدتك" (I saw you) using ATN technique

An Arabic monolingual lexicon was also needed to successfully implement the morphological analyser. The lexicon is designed to reflect the word categories in Arabic—each with a different set of features.

The morphological analysis in Arabic ICALL system analyses the learner's answer in response to a generation question, such as fill-in-the-blank. This answer should meet certain morphological rules. These rules are used to guide the analysis of the learner's answer. This has the following advantages: minimising the ambiguity, facilitating the generation of the feedback in case of ill-formed input, and speeding up the analysis phase.

The Grammar Formalism

The grammar for Arabic contains the grammatical knowledge required to analyse a grammatically correct sentence. The grammar is being developed especially for learning Arabic. Currently, it concerns Arabic grammar at the primary level. We adopted general solutions as much as possible, as this increases the chances that the grammar can be used in other domains as well. Thus, in designing the grammar we seek a balance between short-term goals (a grammar which covers sentences typical for learning Arabic and is reasonably robust and efficient) and long-term goals (a grammar which covers the major constructions of Arabic in a general way).

Arabic grammar in Arabic ICALL is written in DCG formalism. The central formal operation in DCG is the unification of feature-structures. Table 1 describes the features used in the current grammar along with their possible values.

Grammar rules for grammatically correct sentence. A DCG rule has the following form:

Nonterminal-symbol → body

Where "body" is a sequence of one or more items separated by commas. Each item is either a non-terminal symbol or a sequence of terminal symbols written within square

Table 1. Features and their values for the Arabic grammar

Feature	Possible values	Comments
Gender	Masculine/feminine	
Number	Singular/dual/plural	
Definiteness	Definite/indefinite	Applied only to nouns
Special noun	Yes/no	Determine whether or not the noun is Inna and its sisters (إن وأخواتها)
Pattern	Form of pattern (wazen)	
End case	Accusative/nominative/genitive	Iarab الأعراب
Transitivity	Transitive/ intransitive	Applied only to verbs
Special verb	Yes/no	Determine whether or not the verb is Kan and its sisters (كان و أخواتها)
Affix	Affixes of the inflected word	
Current category	Category of the grammatical symbol being parsed	
Next category	Category the grammatical symbol that follows the current symbol	
Noun as adjective	Yes/no	Determine whether or not the noun can be used as an adjective.
Noun as annexation	Yes/no	Determine whether or not the noun can be annexed (إضافة)
Verb tense	Past/present/imperative	
Single word	Single form of the broken plural	
Infinitive	Verb Infinitive form	
Person	First/second/third	Applied only to pronouns
Noun refers to time or place	Time/place	Determine whether the accusative (ظرف) is related to time, related to place, or both

brackets ([and]). The meaning of the rules is that "body" is a possible form for a phrase of type non-terminal symbol. In the right side of a rule, in addition to non-terminals and lists of terminals, there may also be sequences of procedure calls, written within curly brackets ({and}). These are used to express extra conditions that must be satisfied for the rule to be valid.

In the following, we show an extraction of DCG rules used for parsing a grammatically correct Arabic verbal sentence.

verbal_sentence → simple_verbal_sentence (1)
verbal_sentence → prefixed_verbal_sentence (2)
verbal_sentence → special_verbal_sentence (3)
simple_verbal_sentence → verb, subject, object, unrestricted_object (4)

For simplicity, these rules do not include linguistic features such as gender, number and definition, which are assigned to each non-terminal.

Rule (4) illustrates a grammar rule for parsing a simple verbal sentence that consists of four constituents: a verb, a subject, an object and an *unrestricted object* "مفعول مطلق". An unrestricted object is a noun that originates from the infinitive verb. This kind of repetition is considered a mark of good style. In Arabic, repeating the verbal noun after the verb makes the sentence more emphatic. This is explained by the following example:

He **helped** me a great deal of ***help***
ساعدنى مساعدة عظيمة

*Grammar rules for linguistic analysis.*We have also developed another grammar that is used to parse the learner's answer in response to a question about the linguistic analysis of a given Arabic sentence. The parser takes the learner's answer and converts it into a quadruple abstract representation of the canonical form:

السبب	+	علامة الإعراب	+	الإعراب	+	الموقع الإعرابي
Reason	+	Analytic sign	+	End case	+	Analytic location

For example, the linguistic analysis of the word between brackets in the sentence

استمتعت بجو الريف (استمتاعاً)
I enjoyed the rural weather (very much)

is:

مفعول مطلق منصوب بالفتحة لأنه مفرد.

Unrestricted object is in *accusative case* and the diacritic sign is *fat-hah* because it is in *singular* form.

The following lists all of the possible learner's answers:

- مفعول مطلق منصوب وعلامة نصبه الفتحة لأنه مفرد
- مفعول مطلق منصوب وعلامة نصبه الفتحة وهو مفرد
- مفعول مطلق منصوب بالفتحة لأنه مفرد.

This will be parsed into the abstract representation:

مفرد	+	فتحة	+	منصوب	+	مفعول مطلق
singular	+	fat-hah	+	accusative	+	unrestricted object

This abstract representation is unique and unambiguous such that it facilitates comparing the learner's answer with the correct answer generated by the system.

To show how the linguistic analysis is generated, consider the following question as an example.

أعرب ما بين القوسين في الجملة الآتية:

- ألف العقاد (كتبا كثيرة).

Give the linguistic analysis of the words between brackets in the following sentence:

Al-aakad authored (many books).

Consider also the following DCG rule that describes the linguistic analysis of the words between brackets. These words are an object that is followed by an adjective such that the adjective agrees with the object (the noun it modifies) in number, gender, definition, and end case.

```
object(Words_bet_brackets, Rest, Analysis) → [Object], [Adjective],
   {get_analysis(Object, Gender, Num, Def, Words_bet_brackets, Rest1, End_case,
Analysis1, 'مفعول به'),
   get_analysis(Adjective, Gender, Num, Def, Rest1, Rest, End_case, Analysis2, 'نعت'),
   append(Analysis1, Analysis2, Analysis)
   }.
```

This rule takes the list of words to be analysed as input (the words between brackets) and produces as output both the rest of this list, if any, and the linguistic analysis of these words. Parsing of these words proceeds as follows. The first word is bound to the variable *Object*. The *get_analysis/9* procedure yields both the features and the linguistic analysis of this word. Similarly, the second word is bound to the variable *Adjective*. The *get_analysis/9* takes the recognised features of the first word and checks their agreement in number, gender, and definition features with features of the second word, and yields the linguistic analysis of this word. The end case of the adjective (كثيرة-many) is determined by the noun it modifies which is the object (كتبا-books) in this example. Finally, the linguistic analyses of both words are concatenated into a list that constitutes the answer to the question. The generated linguistic analysis of words “كتبا كثيرة" (many books) is the following list:

[[‘مفعول به’, ‘منصوب’, ‘فتحة’, ‘جمع تكثير’], [‘نعت’, ‘منصوب’, ‘فتحة’, ‘مفرد’]]
[[object, accusative, fat-hah, broken-plural], [adjective, accusative, fat-hah, singular]]

The definition of *get_analysis/9* procedure is as follows:

```
get_analysis(Word_being_parsed, Gender, Num, Def, [Word_bet_bracket| Rest],
Rest_words, End_case ,Analysis, Location):-
  morph(Word_being_parsed, lex(_,noun,Gender,Num,Def,Adj,_,_)),
(Word_bet_bracket = = Word_being_parsed - >
  get_Irab(Location,Num, End_case, Word_analysis),
  Rest_words = Rest, Analysis = [Word_analysis])
; Rest_words = [Word|Rest], Analysis = []).
```

This procedure returns the linguistic analysis of the word being analysed if it is one of the words that occur between brackets in the given question. Otherwise, an empty list is returned. The procedure *get_analysis/9* calls the procedure *morph/2* to get features of the input word. Then, it calls get*_Irab/3* to generate the quadruple abstract representation form. The features that result from the morphological analysis of the words between brackets in the above example are as follows:

- كتبا (books): noun, female, broken-plural, indefinite,...
- كثيرة (many): noun, female, singular, indefinite,adj,...

```
get_Irab(Location, Num, End_case, Word_analysis):-
  get_end_case(Location, End_case),
  get_analytic_sign(Num, End_case, Analytic_sign),
  Word_analysis = [Location, End_case, Analytic_sign, Reason].
```

The procedure *get_Irab/3* takes the location and number of the input word and generates its linguistic analysis. It uses two facts *get_end_case/2* and *get_analytic_sign/3*. The fact *get_end_case/2* determines the end case of the word from its location within the sentence. The fact get*_analytic_sign/3* determines the analytic sign of the word from its number and end case.

The linguistic analyses that result from processing *get_Irab/3* to the words between brackets in the above question are as follows:

- كتبا (books): [‘مفعول به’, ‘منصوب’, ‘فتحة’, ‘جمع تكثير’]
  [object, accusative, fat-hah, broken-plural]
- كثيرة (many): [‘نعت’, End_case, ‘فتحة’, ‘مفرد’]
  [adjective, End_case, fat-hah, singular]

Where *End_case* will be determined by the agreement between the adjective and the noun it modifies. This agreement takes place by the unification of the variable *End_case* in the body of the above DCG rule.

```
get_end_case ('مفعول به', 'منصوب').
get_end_case ('مبتدأ', 'مرفوع').
get_end_case ('نعت', _).
. . .
get_analytic_sign ('مثنى', 'مرفوع', 'ألف').
get_analytic sign ('جمع تكثير', 'منصوب', 'فتحة').
get_analytic sign('مفرد', 'منصوب', 'فتحة').
. . .
```

The Feedback System

Feedback is the computer's response to answers made by learners. Feedback gives students a feel of how well they are progressing through a lesson, thereby increasing their confidence levels. It also reinforces the subject matter. In Arabic ICALL, the feedback component includes an error analyser that is used to parse ill-formed learner input and to issue feedback to the learner (see Figure 8). We have augmented the Arabic grammar with heuristic rules (buggy rules) which are capable of parsing ill-formed input and which apply if the grammatical rules fail. The feedback component is implemented using SICStus Prolog. The feedback system compares the analysis of the learner's answer with the correct answer that is generated by the system. If there is a match, a positive message will be sent to the learner. Otherwise, a feedback message

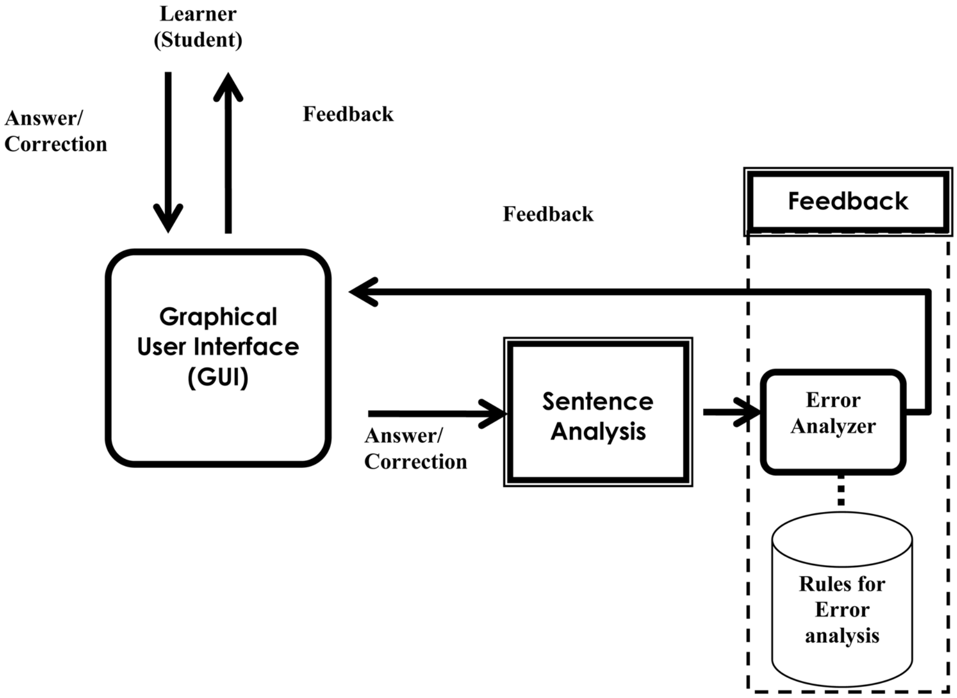


Figure 8. The proposed feedback architecture

will be sent to the learner. The learner can either read the feedback message and correct the typed sentence instantly, or restudy the related grammar items and then correct the sentence without further assistnace. In the following subsections, we show how the system catches the learner's errors and how it handles the ill-formed natural language input.

Rules for Error Analysis

In our implementation, we have augmented the Arabic grammar with heuristic rules which are capable of parsing ill-formed input (buggy rules) and which apply if the grammatical rules fail. As an example, consider the following question to complete a sentence with a suitable unrestricted object ''المفعول المطلق'':

أكمل بمفعول مطلق مناسب:

- •أبر أبى ________.

Complete the following with the correct unrestricted object

- ________.I am kind to my father

The following is an analysis of the possible learner's answer along with the corresponding feedback:

- A word that is not a noun. Issue a message describing that the unrestricted object should be a noun.
- A word that is a noun but does not originate from the infinitive verb. Issue a message describing that the unrestricted object should be the infinitive of the verb.
- A word that is both a noun and originates from the infinitive verb but is defined. Issue a message describing that the unrestricted object should be undefined.
- A word that is a noun, originates from the infinitive verb but needs the end case ''Alef Tanween'', and is undefined. Issue a message describing that a missing end case of the unrestricted object.
- A Correct answer. Issue a positive message.

From the above analysis of the possible learner's answer, we augmented the grammar rule of the unrestricted object by the heuristic rules (buggy rules), defined by check_uo_correctness/4, that handle every possible ill-formed construction as follows.

```
unrestricted_object(UO,Infinitive) – > [Word],
{morph(Word, lex(Stem,Category, _, _, Def, _, _, _)),
check_uo_correctness(Infinitive,Feedback,Word,[Stem,Category,Def]),
(Feedback = = >-’ إجابة صحيحةUO = unrestricted_object(Word)
      ; UO = incorrect_unrestricted_object(Word,Feedack) )
}.
check_uo_correctness(_, Feedback, Word, [_,Category,_]):- not Category = noun,!,
error_flagging(not(noun),‘‘المفعول المطلق’’, Feedback,[Word]).
```

```
check_uo_correctness(Infinitive, Feedback, Word,[Stem, noun,_]):- not Stem = Infinitive,
!,
error_flagging(not(infinitive), ‘‘المفعول المطلق’’, Feedback,[Word]).
check_uo_correctness(Infinitive, Feedback, Word,[Infinitive, noun, defined]):-
   error_flagging(not(undefined_noun), ‘‘المفعول المطلق’’, Feedback, [Word]),!.
check_uo_correctness(Infinitive, Feedback, Word,[Infinitive, noun, undefined]):-
name(Word, Str),
name(Infinitive,Str1),
(need_alaf_tanween(Infinitive) - >
   (append(Str1,‘‘ا’’,Str) - > Feedback = ‘‘إجابة صحيحة’’
   ; error_flagging(need_alaf_tanween, ‘‘المفعول المطلق’’, Feedback, [Word])
   )
; Feedback = ‘‘إجابة صحيحة’’
).
```

Error Handling Mechanism

Learner’s responses which have special handling mechanisms in case of ill-formed learner input are: linguistic analysis, classification into categories, sentence transformation, and completing a sentence. They are discussed in the following subsections.

Handling of linguistic analysis. Linguistic analysis questions can apply either for an entire sentence or a part of it. The latter is usually a sequence of words between brackets. The following description outlines the steps for handling of linguistic analysis:

- Parse the given sentence (or the sequence of words between brackets) and generate its linguistic analysis in a quadruple abstract representation form.
- Convert learner answer into the abstract representation form.
- Compare the learner’s answer with the generated answer to issue the appropriate feedback message.

Example:

هل تري فرقا إعرابيا فيما بين القوسين؟

○ استمتعت بجو الريف (استمتاعاً).

○ أذهب إلى الريف (استمتاعاً) بجوه.

What’s the difference in linguistic analysis of the words between brackets?

- I enjoyed the country weather (very much)
- I go to the countryside (to enjoy) its weather

The parser is used to analyse each of the input sentences. The generated correct linguistic analyses of the words between brackets are the following:

- First word: [‘مفعول مطلق’, ‘منصوب’, ‘فتحة’, ‘مفرد’]
  [unrestricted object, accusative, fat-hah, singular]

- ■ Second word: [‘مفعول لأجله’, ‘منصوب’, ‘فتحة’, ‘مفرد’]
  [causative object, accusative, fat-hah, singular]

The difference, in this case, is in the *analytic location* (i.e., the first argument in the quadruple abstract representation). The learner’s answer is also converted to the quadruple abstract representation. The comparison between these representations will issue the appropriate feedback that describes the source of the error. The possible source of the errors could be: incorrect analytic location, incorrect end case, incorrect reason, or a partially correct answer.

Handling of classification into categories. Classification into categories questions can apply either for identifying morphological categories or for identifying syntactic constituents (possibly, a complete sentence). The following description outlines the steps for handling classification into morphological categories:

- Morphologically analyse the words in the given sentence and determine the words features.
- Generate an N lists, a classification of the words according to the questions words.
- Assign the learner answer to N Lists.
- Compare the learner’s answer with the generated answer to issue the appropriate feedback message.

Example:

عين الاسم والفعل والحرف في الجملة الآتية:

○ وقف التلاميذ احتراماً للمعلم

Identify the category of each of the words in following sentence

- ■ The students stood up respecting the teacher

The morphological analyser is used to analyse each inflected Arabic word to recognise its category. Then, according to the word category, the words are classified into three lists. The following is the generated correct answer:

- ■ Verb: [[‘وقف’, verb, male, singular, past, . . .]] (stood)
- ■ Noun: [[‘المعلم’, noun, male, singular, def, . . .], [‘احتراما’, noun, male, singular, undef, . . .],

  [‘التلاميذ’, noun, male, plural, def, . . .]] (the teacher, honoring, the students)
- ■ Particle: [[‘ال,’ particle, def_article, . . .]] (the)

The learner’s answer is also assigned to three lists containing verbs, nouns, and particles, respectively. The comparison between the corresponding lists will issue the appropriate feedback that describes the source of the error. The possible source of the errors could be: missing words from the respective morphological category, or assigning a word to an incorrect morphological category.

The following description outlines the steps for handling of classification into syntactic constituents:

- Parse the given sentence and determine the parse tree.
- Generate an N lists, a classification of the sentence's constituents according to the questions words.
- Assign the learner answer to N Lists.
- Compare the learner's answer with the generated answer to issue the appropriate feedback message.

Example:

عين المبتدأ والخبر مبينا نوعة في الجملة الآتية:

○ الجنود الشجعان ينتصرون

Identify the inchoative and enunciative, and the type of the enunciative in the following sentence.

■ The brave soldiers fought to victory

The parser is used to analyse the input sentence into a parse tree as follows:

```
nominal_sentence(inchoative(noun('الجنود'  noun, male, plural, def, ...),
                                          adj('الشجعان', noun, male, plural, def, adj, ...)),
        enunciative(verbal_sentence(verb(‘ ينتصرون’, verb, male, plural, present, ...))))
```

Then, according to the parse tree, the words are classified into three lists. The following is the generated correct answer:

■ inchoative: [noun('الجنود', noun, male, plural, def, ...), adj('الشجعان', noun, male, plural, def, adj, ...)] (the brave soldiers)

■ enunciative: [verbal_sentence(verb('ينتصرون', verb, male, plural, present, ...))] (make victory)

■ enunciative type: [verbal_sentence]

The learner's answer is also assigned to three lists containing inchoative, enunciative, and enunciative type, respectively. The comparison between the corresponding lists will issue the appropriate feedback that describes the source of the error. The possible source of the errors could be: incorrect constituent type (analytic location), or assigning a syntactic constituent to an incorrect category.

Handling of transformation of a sentence. Transformation questions require the learner to change/rewrite the form of a sentence. The following description outlines the steps for handling of transformation of a sentence:

- Parse the given sentence and determine the parse tree; apply a tree-to-tree transformation to generate the transformed parse tree.
- Parse the learner's answer to determine the parse tree.
- Compare the parse tree of the learner's answer with the parse tree of the generated answer to issue the appropriate feedback message.

Example:

الجملة الآتية أسمية أجعلها فعلية:

○ المعلم يشرح الدرس

Change the following nominal sentence into verbal sentence

○ The teacher teaches the lesson

The parser is used to analyse the input sentence into a parse tree as follows:

```
nominal_sentence(inchoative(noun('المعلم', noun, male, singular, def, ...)),
        enunciative(verbal_sentence(verb('يشرح', verb, male, singular, present, ...),
                object(noun('الدرس', noun, male, singular, def, ...)))))
```

Then, the parse tree of the nominal sentence is transformed to the following verbal sentence.

```
verbal_sentence(verb('يشرح', verb, male, singular, present, ...)),
        subject(noun('المعلم', noun, male, singular, def, ...)),
        object(noun('الدرس', noun, male, singular, def, ...)))
```

In addition, words of the transformed parse tree is grouped in a list as follows:

- ■ List of words: [verb('يشرح', verb, male, singular, present, ...), noun('المعلم', noun, male, singular, def, ...), noun ('الدرس', noun, male, singular, def, ...)]

The learner's answer is also analysed into a parse tree and words are grouped into a list. The comparison between the corresponding representations will issue the appropriate feedback that describes the source of the error. The possible source of errors could be: extra words, missing words, grammatically incorrect sentence, or incorrect transformation of a word (incorrect verb: tense, number, gender, etc; incorrect noun: number, gender, definition, etc).

Handling of Fill-in-the-blanks. Fill-in-the-blank questions can apply for the generation of isolated words with a particular form, or to the generation of words to complete a sentence that achieves feature agreement among its constituents. The following description outlines the steps for handling of rewriting of isolated words with particular morphological form:

- Morphologically generate the given words and determine their features.
- Morphologically analyse the learner's answer and determine the words features.
- Compare the parse tree of the learner's answer with the parse tree of the generated answer to issue the appropriate feedback message.

Example:

ثن واجمع سالما الكلمات الآتية:

○ المهندس

○ الثمرة

○ صحراء

What is the correct dual and regular plural of the following words?

- Engineer
- Fruit
- Desert

The morphological generator is used to synthesise the given words into the dual and plural forms taking into consideration the possible end case. The following is the generated correct answer:

- Dual list: [['المهندسان,' noun, male, dual, def, nominal, ...], [['المهندسين', noun, male, dual, def, accusative, ...], ['الثمرتان', noun, female, dual, def, nominal, ...], ['الثمرتين', noun, female, dual, def, accusative, ...], ['صحراوان', noun, female, dual, undef, nominal, ...], ['صحراوين', noun, female, dual, undef, accusative, ...]]
- Plural list: [['المهندسون', noun, male, plural, def, nominal, ...], [['المهندسين', noun, male, plural, def, accusative, ...], ['الثمرات', noun, female, plural, def, ...], ['صحراوات', noun, female, plural, undef, ...]]

The morphological analyser is used to analyse each inflected Arabic word of the learner's answer. These words are classified into two lists containing dual and plural forms, respectively. The comparison between the corresponding lists will issue the appropriate feedback that describe the source of the error. The possible source of errors could be: different word, incorrect word category (switch dual forms with plural forms), incorrect generation of a word (incorrect verb: tense, number, gender, etc; incorrect noun: number, gender, definition, etc).

The following description outlines the steps for handling of fill-in-the-blank to achieve agreement among sentence constituents:

- Analyse the sentence and determine its constituents.
- Morphologically generate the missed words.
- Parse the learner's answer to determine the parse tree.
- Compare the parse tree of the learner's answer with the parse tree of the generated answer to issue the appropriate feedback message.

Example:

أكمل بمفعول مطلق مؤكد للفعل:

○ أبر أبي ________.

Complete the sentence with an unrestricted noun that makes the sentence more emphatic?

- I am kind with my father ________.

The parser is used to analyse the partial input sentence and determine its constituents as follows:

```
verbal_sentence(verb('أبر', verb, male, singular, present, intrans, 'بر' ...),
          subject(noun('أبي', noun, male, singular, undef, ...)))
```

The morphological generator uses the infinitive verb ‘بر' (kindness) of the main verb to synthesise the unrestricted object `برا' (extremely kind). The learner's answer is also analysed into a parse tree. The comparison between the corresponding representations will issue the appropriate feedback that describe the source of the error. The possible source of errors could be: different word (sense or category), incorrect morphological generation of a word (incorrect verb: tense, number, gender, etc; incorrect noun: number, gender, definition, etc), incorrect syntactic generation of a word (`s' not in emphatic form, does not originates from the infinitive verb, etc)

Conclusions

In this paper, we described the development of an ICALL system for learning Arabic by students at primary schools or by learners of Arabic as a second or foreign language. NLP tools can be useful for ICALL and hence usefully used in Arabic ICALL for reacting to the learner's response in order to give appropriate feedback to the learner. Learner-system communication in free natural language is computationally the most challenging and pedagogically the most valuable scenario in Arabic ICALL. The deep syntactic analysis of the learner's answer, whether correct or wrong, is compared against a system-generated answer. This enables feedback elaboration that helps learners to understand better their knowledge gab.

Arabic ICALL has two main types of test items for interaction with the learner: selection-type and supply-type requiring the learners to write a few words. The objective test method is used such that there is no ambiguity about what that correct answer should be.

The rule-based approach is used to give some freedom to the language learners in the way they phrase their answers, while enabling the exercise author to enter only one possible correct answer, thus saving much time compared to the previous pattern matching answer coding approach. While the statistical-based approach is costly and has some difficulties in distinguishing between well-formed and ill-formed input, the rule-based approach has the advantage of providing detailed analysis of the learner's answer using linguistic (morphological and syntactic) knowledge. NLP tools in Arabic ICALL include a morphological analyser and a syntax analyser. These tools are used to analyse inflected Arabic words and Arabic sentences, and provide a linguistic analysis of an Arabic sentence. The error analyser is responsible for diagnosing and handling of ill-formed input.

Arabic ICALL was implemented using SICStus Prolog, Visual Basic, Flash and Microsoft Access. The system is transportable and capable of running on an IBM PC which allows the learner to use it to learn Arabic language anywhere and anytime.

We plan to enrich the present system—for example, add a student or course management facility that allows us to have full performance and record-keeping features, add more multimedia instructional material to allow learners fully comprehend what they learn in natural settings, make the system available on the Internet to serve remote learners worldwide (especially learners of Arabic as a second language), and extend the grammar coverage to include more advanced grammar levels.

Notes

1. The author is on leave of absence from the Faculty of Computers and Information, Cairo University.
2. CALL is also known as computer-assisted instruction (CAI), computer-aided instruction (CAI), or computer-aided language learning.
3. We refer to Cachia (1973) for the translation of Arabic terminology.

References

Abou Ela, M. (1994). *Knowledge-based techniques in an Arabic syntax analysis environment*. Unpublished doctoral thesis, Institute of Statistical Studies and Research, Cairo University, Egypt.

Allen, J. (1995). *Natural language understanding* (2nd ed.). Redwood City, CA: The Benjamin/Cummings Publishing Company.

Boytcheva, S., Vitanova, I., Strupchanska, A., Yankova, M., & Angelova, G. (2004). Towards the assessment of free learner's utterances in CALL. *Proceedings of InSTIL/ICALL2004—NLP and Speech Technologies in Advanced Language Learning Systems*, 17–19 June 2004, Universita Ca Fascari, Venice, Italy, 187–190.

Cachia, P. (1973). *The monitor: A dictionary of Arabic grammatical terms*. Beirut: Librairie du Liban.

Cameron, K. (Ed.). (1999). *CALL-media, design and applications*. Lisse, The Netherlands: Swets & Zeitlinger.

Cushion, S., & Hémard, D. (2002). Applying new technological developments to CALL for Arabic. *Computer Assisted Language Learning (CALL): An International Journal*, *15*(5), 501 – 508.

Cushion, S., & Hémard, D. (2003). Designing a CALL package for Arabic while learning the language ab initio. *Computer Assisted Language Learning (CALL): An International Journal*, *16* (2/3), 259 – 266.

Ditters, E., Oostdijk, N., & Cameron, K. (1993). *Processing Arabic: Computer assisted language learning*. Oxford, England: Intellect Books.

Dougherty, R. (1994). *Natural language computing: An English generative grammar in prolog*. Mahwah, NJ: Lawrence Erlbaum Associates.

Gamper, J., & Knapp, J. (2002). A review of intelligent CALL systems. *Computer Assisted Language Learning (CALL): An International Journal*, *15*(4), 329 – 342.

Gazdar, G., & Mellish, C. (1990). *Natural language processing in prolog: An introduction to computational linguistics*. Massachusetts, MA: Addison-Wesley.

Gheith, M., Dawa, I., & Afifty, M. (1996). ISTAL: Instructional software for teaching the Arabic language. *Proceedings of the First International Conference on Computer and Advanced Technology in Education*, Egypt, 41 – 66.

Hegazi, N., Ali, G., Abed, E. M., & Hamada, S. (1989). Arabic expert system for syntax education. *Proceedings of the Second Conference on Arabic Computational Linguistics*, Kuwait, 596 – 614.

Holland, M. V., Maisano, R., & Alderks, C. (1993). Parsers in tutors: What are they good for? *CALICO Journal*, *11*(1), 28 – 46.

Holland, M. V., Kaplan, J. D., & Sama, M. R. (Eds.). (1995). *Intelligent language tutors: Theory shaping technology*. Mahwah, NJ: Lawrence Erlbaum Associates.

Lam, F. S., & Pennington, M. C. (1995). The computer vs the pen: A comparative study of word processing in a Hong Kong secondary classroom. *Computer Assisted Language Learning (CALL): An International Journal*, *8*(1), 75 – 92.

McEnery, T., Baker, J. P., & Wilson, A. (1995). A statistical analysis of corpus based computer vs. traditional human teaching methods of part of speech analysis. *Computer Assisted Language Learning (CALL): An International Journal*, *8*(2), 259 – 274.

Mote, N., Johnson, L., Sethy, A., Silva, J., & Narayanan, S. (2004). Tactical language detection and modeling of learner speech errors: The case of Arabic tactical language training for American English speakers. *Proceedings of InSTIL/ICALL2004—NLP and Speech Technologies in Advanced Language Learning Systems*, Italy. Retrieved May 2005 from http://sisley.cgm.unive.it/ICALL2004/link13.htm.

Nielsen, H. (2001). ArabVISL: Arabic grammar on the Internet. *Proceedings of the Fifth Nordic Conference on Middle Eastern Studies*, Lund. Retrieved May 2005 from http://www.hf-fak.vib.no/institutter/smi/pal/paltoc.html

Nielsen, H., & Carlsen, M. (2003). Interactive Arabic grammar on the Internet: Problems and solutions. *Computer Assisted Language Learning (CALL): An International Journal*, *16*(1), 95 – 112.

Othman, E., Shaalan, K., & Rafea, A. (2003). A chart parser for analyzing modern standard Arabic sentence. *Proceedings of the MT Summit IX Workshop on Machine Translation for Semitic Languages: Issues and Approaches*, USA. Retrieved May 2005 from http://www-2.cs.cmu.edu~alavie/semitic-MT-wshp.html

Pereira, F., Shieber, C., & David, H. (1986). Definite clause grammars for language analysis—a survey of the formalism and a comparison with augmented transition networks. In B. Grosz, K. Jones & B. Webber (Eds.), *Readings in natural language processing* (pp. 24 – 101). San Francisco, CA: Morgan Kuffmann.

Rafea, A., & Shaalan, K. (1993). Lexical analysis of inflected Arabic words using exhaustive search of an augmented transition network. *Software Practice and Experience*, *23*(6), 567 – 588.

Shaalan, K. (2003). Arabic GramCheck: a grammar checker for Arabic. *Egyptian Informatics Journal, Faculty of Computers & Information*, *4*(1), 94 – 111.

Shaalan, K., Allam, A., & Gomah, A. (2003). Towards automatic spell checking for Arabic. *Proceedings of the Fourth Conference on Language Engineering, Egyptian Society of Language Engineering (ELSE)*, Egypt, 240 – 247.

Swartz, M., & Yazdani, M. (Eds.). (1992). Intelligent tutoring systems for foreign language learning, chapter introduction. New York: Springer-Verlag.

Warschauer, M. (1996). Computer-assisted language learning: An introduction. In S. Fotos (Ed.), *Multimedia language teaching* (pp. 3 – 20). Tokyo, Japan: Logos International.

Woods, W. (1970). Transition network grammar for natural language analysis. *comm. ACM*, 10, 591 – 66.