Available online at www.sciencedirect.com
ScienceDirect
Journal of Electrical Systems and Information Technology xxx (2016) xxx–xxx
www.elsevier.com/locate/jesit

An intelligent tutoring system for teaching the grammar of the Arabic language

Mona H. Mahmoud *, Sanaa H. Abo El-Hamayed

Electronics Research Institute, Cairo, Egypt

Received 12 August 2015; received in revised form 13 February 2016; accepted 17 April 2016

Abstract

In this research work, an Intelligent tutoring system (ITS) is presented to simulate the behavior of the educational process. Any intelligent tutoring system consists of a Tutoring Module, a question selector, an Expert Module, a student model, and a graphical user interface. This work is presented in parallel with implementing a project called “Arabic Grammar Tutor” that is appreviated as “AG_TUTOR”. A part of this project is adopted and discussed in this paper. This part consists of the first three modules of the ITS. These modules are: the Tutor Module, the Question Selector Module, and the Expert Module. Moreover, the knowledge base and/or domain knowledge will be also conducted. Such modules are implemented and tested. The curriculum of the Arabic grammar of the fourth grade; elementary schools in Egypt; is adopted as a domain knowledge. Moreover, some sort of text analysis will be considered.

Keywords: Intelligent tutoring systems; Tutoring Module; Question Selector Module; Expert Module; Domain knowledge; Arabic grammar

1. Introduction and related work

Different intelligent tutoring systems (ITS) were implemented in several subject domains. Examples of such systems are briefly mentioned as shown below:

There were two ITS projects that functioned based on the conversational dialog: *AutoTutor* and *Why2-.Atlas*. Some intelligent tutoring systems were also presented by several researchers. The idea behind those projects and systems was that the programs would begin with leading questions for the students and would give out answers.

* Corresponding author.
E-mail address: monah1957@hotmail.com (M.H. Mahmoud).
Peer review under the responsibility of Electronics Research Institute (ERI).

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jesit.2016.04.001

a) *The AutoTutor's* domain is the computer technology. It is a computer tutor that simulates the course patterns and educational techniques of a real human tutor via a dialog with the learner using natural language. It has been developed incrementally where the latter has a 3D interactive interface and has been implemented using visual basic. NET and C# programming language. Using the natural language concept in this project means that the tutoring occurs in the form of a conversation, with human input presented using either voice or free text input. To handle this input, the Auto Tutor project uses computational linguistics algorithms including latent semantic analysis, regular expression matching, and speech act classifiers (Al Emran and Shaalan, 2014).
b) *Why2-Atlas* is an ITS that deals with physics principles as a domain knowledge. The students input their work in paragraph form and the program converts their words into a proof by making assumptions of student beliefs that are based on their explanations. In doing this, misconceptions and incomplete explanations are highlighted. The system then addresses these issues through a dialog with the student and asks the student to correct his/her essay. A number of iterations may take place before the process is completed (Vanlehn et al., 2002).
c) *Web-based Intelligent Language Tutoring Systems (German Tutor)* is constructed to form the grammar practice for a course in German via a web-based environment. Intelligence appeared through a parser that parses the German grammar which is the learner's input. The system's student model provides students with adaptive feedback that is suited to their expertise along with some proposed exercises. Intelligent and adaptive mechanisms were built on a separate server side where the answer is processed. The system has been evaluated through testing it with 19 students within 1 h class. 84% of the students reported that the system was very robust by providing them the immediate feedback and free grammar practice (Al Emran and Shaalan, 2014).
d) *Beetle II System:* is a tutorial dialog system designed to accept unrestricted language input with two different tutorial planning and dialog strategies. The domain of the system is the basics of the electricity and electronics. A natural language dialog parser has been used in order to parse any input from the student as well as to extract an applicable semantics from each statement and identify paraphrases that could bear similar meaning. Beetle II has been implemented to examine whether self-explanation could be handled by computers that are supported by NLP techniques. The system has been developed to ask the learners to illustrate their answers in order to give them a detailed feedback. The system helps to get students into the correct illustration without referring to the short-answer questions and without referring to the tutor after each tutorial response (Dzikovska et al., 2010).

Natural Language Processing (NLP) is one of the artificial intelligence fields which is interested in interpreting and processing human natural languages. NLP researchers aim to gather knowledge on how human beings understand and use language. ITSs vary in their capabilities according to their components and using of NLP tools. Systems use NLP tools have the ability to evaluate the student answer and diagnose his/her misconceptions (Chowdhury, 2003).

Dealing with the linguistic computation of Arabic language is a difficult task. The difficulty comes from many sources: (1) the complex of the Arabic syntax, (2) The omission of vowels in writing Arabic "altashkiil", (3) The free word order nature of Arabic sentence. For those reasons, few researches are involving Arabic-based tutoring systems (Shaalan, 2003).

Our system in this work AG_TUTOR simulates the behavior of instructors and students in the educational environment. AG_TUTOR is considered an adaptive learning system which uses computers as an interactive teaching machine. The system adapts the presentation of educational material according to students' learning needs, as indicated by their responses to questions and tasks. Also, the system has capabilities of NLP to analyze the student answer to diagnose his errors and medicate his problems (Samuelis, 2007).

The tutoring system is executed as a series of tutoring tasks. Each task is implemented using lessons explanation supported with examples and questions (Nkambou et al., 2010). To represent the Arabic grammar in an accurate form, an educational expert and a domain expert from the Ministry of Education in Egypt are consulted. A real academic data course of the Arabic grammar of the fourth grade of elementary schools in Egypt was adopted as a test-bed. In the proposed system, a framework is put for the knowledge base that includes all kinds of the needed knowledge (Mahmoud and Abo El-Hamayed, 2015; Prentzas and Hatzilygeroudis, 2009).

The organization of this paper will be as follows: Section 2 presents the domain knowledge of the AG_TUTOR while Section 3 presents Knowledge Base. Sections 4 and 5 discuss the Tutoring Module and Question Selector Module respectively. Moreover, The Expert Module is presented in Section 6. Finally, Section 7 concludes the whole work.

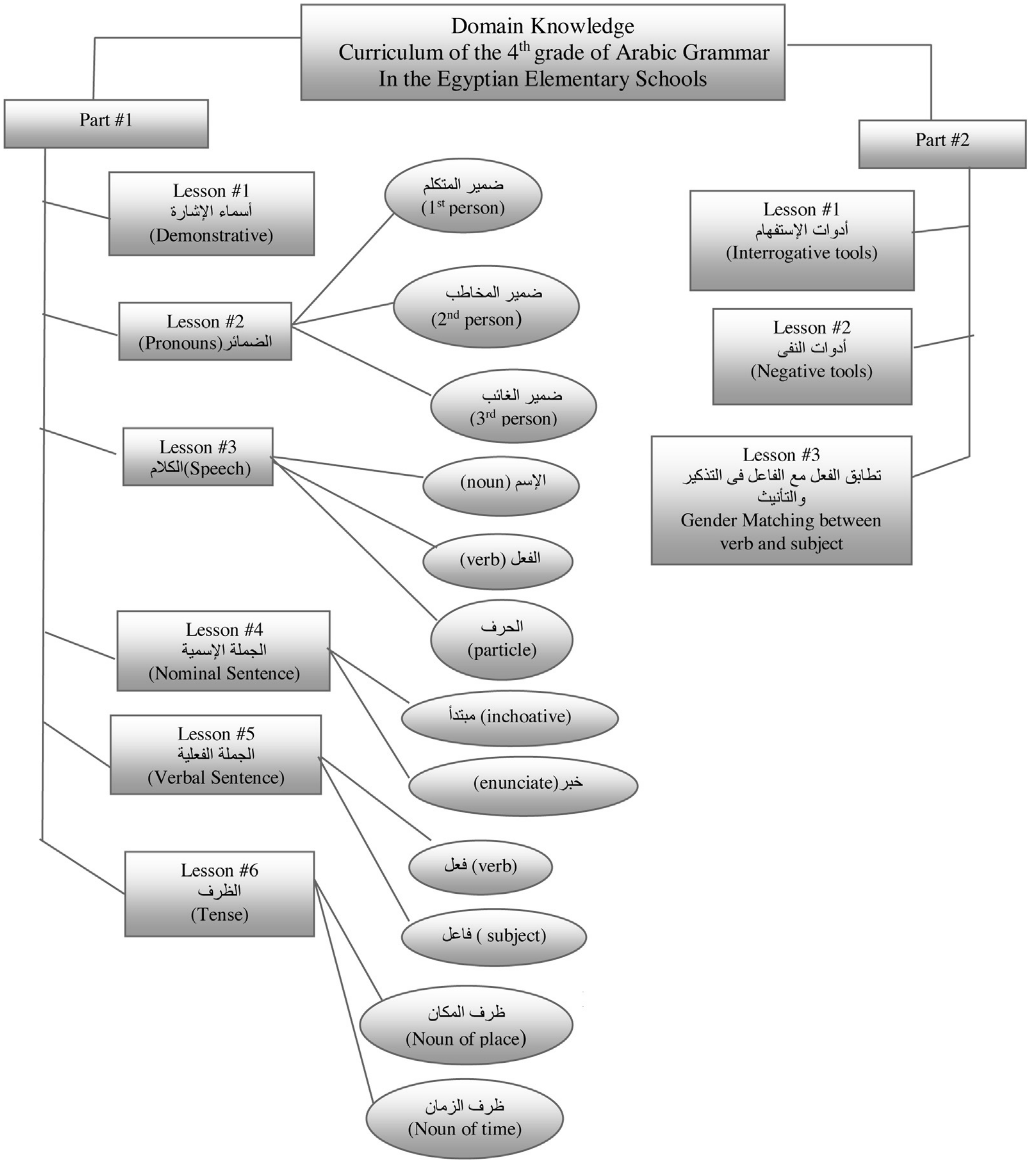

Fig. 1. The main lessons/concepts of the adopted domain.
Note: *A rectangle denotes a lesson or a concept. *An ellipse denotes a fragment of the lesson.

2. The domain knowledge in AG_TUTOR

Domain knowledge in artificial intelligence is the knowledge about the environment in which the target system operates. A domain model is created in order to represent the vocabulary and key concepts of the problem domain. The domain model also identifies the relationships among all the entities within the scope of the problem domain, and commonly identifies their attributes. An important advantage of a domain model is that it describes the scope of the problem domain (Brusilovsky and Cooper, 1998).

The adopted domain is the curriculum of the grammar of the Arabic language of the fourth grade of the elementary schools in Egypt. The knowledge of this curriculum is acquired from the Arabic instructor transcripts.

To design a structure for the domain the technology of adaptive hypermedia system is used. In AG_TUTOR, the domain is already transferred into concepts and fragments as shown in Fig. 1.

Please cite this article in press as: Mahmoud, M.H., Abo El-Hamayed, S.H., An intelligent tutoring system for teaching the grammar of the Arabic language. J. Electr. Syst. Inform. Technol. (2016), http://dx.doi.org/10.1016/j.jesit.2016.04.001

The structure of the domain of the Arabic grammar of Fig. 1 was designed as concepts and fragments like that (if a sample of 3 lessons is taken):

Concept name: speech الكلام

Its fragments: noun الإسم , verb الفعل , particle الحرف

Concept name: the nominal sentence الجملة الإسمية

Its fragments: inchoative مبتدأ , enunciate خبر

Concept name: الجملة الفعلية verbal sentence

Its fragments: فعل , فاعل verb, subject

And so on, all the grammar that will be taught was transferred to this form and then to database tables on the computer.

3. Knowledge base

A *knowledge base* (*KB*) is a technology used to store complex structured and unstructured information used by a computer system for artificial intelligence domain. A knowledge-based system consists of a knowledge-base that represents facts about the world and an inference engine that can reason about those facts and use rules and other forms of logic to deduce new facts or highlight inconsistencies (Roth et al., 1983).

AG_TUTOR knowledge is represented using the production rules method that consists of: facts and rules. The details of the facts and rules are shown below.

3.1. The data base

The facts are represented through a relational database. This database is implemented using Microsoft Access including eleven tables. In the following the tables and a sample from each of them are represented:

1. *Concepts table*: contains the names of the lessons such as: (nominal sentence, verbal sentence, ...).
2. *Components table*: contains the components of each rule such as: (verbal sentence consists of: verb and subject).
3. *Types table*: contains the type of the concept such as: in درس الضمائر (pronouns lesson).
4. *Examples table*: contains different examples on each lesson to help the student understand it.
5. *Problems table*: contains question heads such as: (choose the correct word from those between brackets).
6. *Questions table*: contains the questions below each question head or problem.
7. *Words table*: contains all the words that are included in all exercises that will be presented to the student and their feature codes. This table is called the dictionary. It has a huge number of words with the features of each word. Such features are type, gender, count and anatomy. Tables 8–11 have relationships with Table 7 and they contain the features of the words.
8. *Classify table*: it contains the type code which is a code for each type of the words such as noun, verb and particle that take codes 1, 2, 3 respectively.
9. *Gender table*: it contains the gender code that is for the gender of each word mainly 1 for male and 2 for female.
10. *Count table*: it has a code for the count of the word. The codes are 1, 2, 3 for single, double, and plural respectively.
11. *Anatomy table*: it has the codes for the types of the noun such as: human, animal, planet, inanimate (solid), and adjective which take codes from 1 to 5 respectively.

3.2. The production rules

A *production system* (or *production rule system*) is a computer program typically used to provide some form of artificial intelligence which consists primarily of a set of rules about behavior. These rules, termed *productions*, are a basic representation found useful in automated planning, expert systems and action selection. A production system provides the mechanism necessary to execute productions in order to achieve some goal for the system (Anon, 2016a).

Productions consist of two parts: a sensory precondition (or "IF" statement) and an action (or"THEN"). If a production's precondition matches the current state of the world, then the production is said to be *triggered*. If a production's action is executed, it is said to have *fired*. A production system also contains a database, sometimes called working memory, which maintains data about the current state or knowledge, and a rule interpreter. The rule interpreter must provide a mechanism for prioritizing productions when more than one is triggered (Anon, 2016a; Anohina, 2007).

At the AG_TUTOR system, the rules are represented through a big group of rules for all modules. The system has many types of rules:

- Group of rules for the Tutoring Module to allow the student moving through the tutor to get the information he/she needs.
- Two groups of rules for the Question Selector Module. A group to select a question or more randomly for a specific lesson and a group to allow the student to answer the question.
- Many groups of rules for the Expert Module. A group for each type of the questions.

As a sample from AG_TUTOR rules:

Rule #1: (from the question selector module)
To generate a random question head:

If (stacknum.is Empty()) stacknum.add (question_head);
Rs=db. Statement. Execute Query ("SELECT question_head FROM problems WHERE c_code =selected_concept_ID AND P_no = question_head AND type_code = typ_code_rand)

This rule is selecting a question head randomly from the problems table in the database for a specific lesson that the student selects and put it in the stack to represent it to the student.

Rule #2: (from the Expert module)
To solve the following question:

اختراسما مما يأتى وضعه فى مكانه المناسب :الشجرة .... (مورقة - الكتاب – الحصان)
Choose a noun from those between the brackets: The tree …………… (has leaves – is the book – is the horse)

If ((type of first_word.equals (type of second_word))&&(gender of first_word.equals (gender of second_word)) && (count of first_word .equals (count of second_word)) && (anatomy of first_word.equals (anatomy of secnod_word))) then {answer[0]= choices[i]}

Rule #3: (from the Expert module)
To solve the following question:

اسحب اسم الإشارة المناسب وضعه فى مكانه المناسب فيما يأتى:......... تلميذ (هذا – هذان- هؤلاء)
(Press on the correct answer between brackets: (this – these) --------- is a pupil)

if_((Gender of second_word .equals (Gender of first_word)) && (count of second_word . equals(count of first_word)))_ ***then*** _{answer[0]= choices[i]}_

Rule #4: (from the Expert module)
أعرب الجمل التالية :العصفورجميل , دخل التلاميذ الفصل
Parse the following sentences: the bird is beautiful, the pupils enter the class

if _((the first word is noun or esmeshara or damear) && (second word is noun))_ ***then*** _answer is: the first word is_ مبتدأ _& the second word is_ خبر
else if _((the first word is verb) && (second word = noun))_ ***then*** _answer is: the first word is verb &the second word is subject_

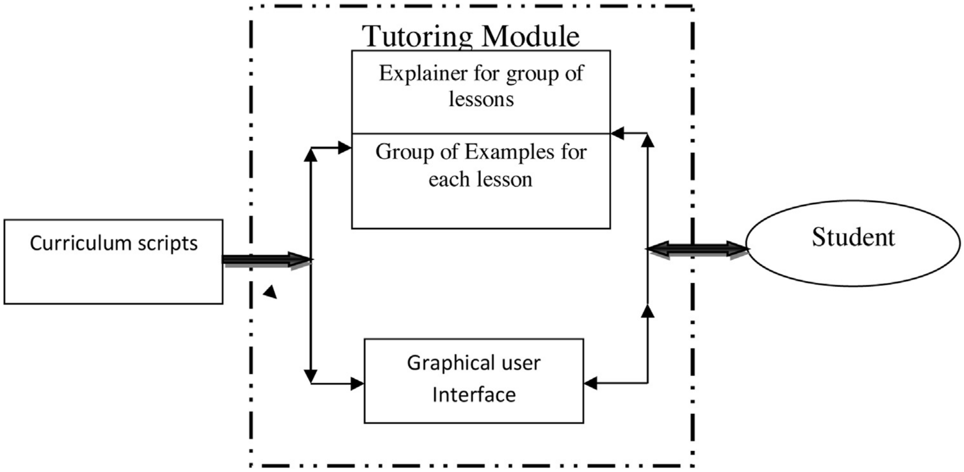

Fig. 2. Design of the Tutoring Module.

4. The Tutoring Module

The Tutoring Module is the instructional module that designs and regulates the instructional interactions with the students. The tutoring goal structure comes from instructor transcripts. Assessment is a very important function of the Tutoring Module. The function of the Tutoring Module is essentially to perform continuous assessment of the student, and thereby interact with the Expert Module to prescribe further action (Akbulut and Cardak, 2012; Anon, 2016b).

4.1. The design of the Tutoring Module

The input of the Tutoring Module in AG_TUTOR is the script that is made by the Arabic Expert who transferred the lessons of the curriculum to a form that can be implemented on the computer. The output of the module is group of snapshots that are linked to each other and represented to the student to give a very attractive explanation of the lessons, so, he can study the lesson he chooses. He can go to the next screen, go back to the previous one, go to other lessons or EXIT through a group of icons.

As shown in Fig. 2, The Tutoring Module consists of the following:

- An Explainer that represents the lessons through many screens for each lesson, each screen explains a part of the lesson with a high level of the graphics: pictures, sound and animation.
- An Example model: that represents group of examples for each lesson in an attractive form supported with multimedia facilities.
- A graphical user interface: an attractive graphical user interface allows student to interact with the system.

Fig. 3 shows a snapshot from the lesson of the demonstrative nouns (أسماء الإشارة).

5. Question Selector Module

The main goal of the Question Selector Module is to select a question randomly and display it to the student according to the lesson that he selects and gives him the chance to answer (Mills and Dalgarno, 2007).

The input of the Question Selector Module is the question bank in the database. This bank of questions is represented mainly through the problems as in Table 5, question table as in Table 6 and their relationships with other tables in the database. The question bank contains a huge number of questions. The bank is divided into many groups of questions as a group for each lesson. In AG_TUTOR, there are many types of questions mainly:

Fig. 3. A snapshot from the Tutoring Module.

1. Multiple Choices Questions (MCQ)
2. Match the related correct sentence
3. Press on something (like اسم إشارة أو ضمير مخاطب)
4. Fill in the space with the correct answer from the brackets
5. Get out a verb, a noun, or a particle or . . ..
6. Parse a sentence
7. Reorder a nominal sentence to be a verbal sentence and vise versa.
8. Generate the plural, double or single of a noun

5.1. Design of the Question Selector Module

As shown in Fig. 4, the Problem Selector Module consists of:

1. Graphics interface: which contains all graphic components needed for all questions including form pictures, labels pictures, and fonts.
2. The database: a group of tables that contain a huge number of questions.
3. Question requirements definer: a sub-module which is used to analyze the question requirements and passes all identical arguments, attributes, events & graphics to main program form.

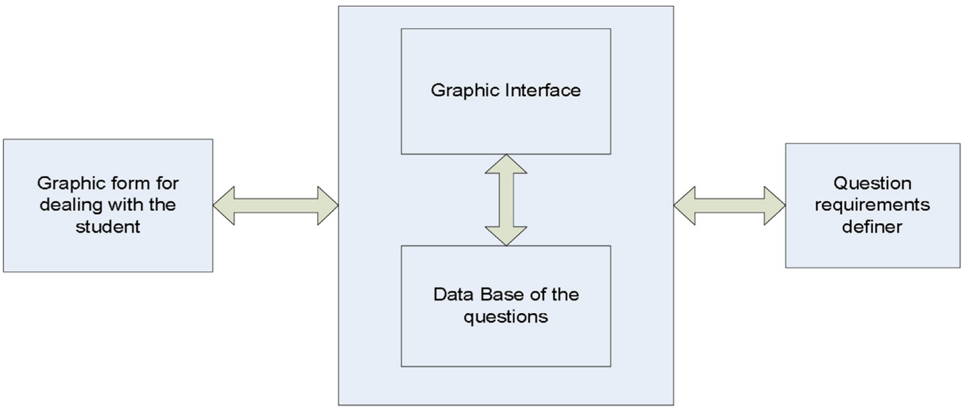

Fig. 4. The Question Selector Module.

Table 1
The concepts table.

Concept		
Concept_ID	Concept_Name	Concept_Description
2	الضمائر	–
4	الجملة الإسمية	هى جملة تبدأ باسم
5	الجملة الفعلية	هى جملة تبدأ بفعل

The primary key in this table is (concept_ID) (the concept or lesson no.) and is related to c_code in Tables 2–6.

Table 2
Components table.

c_code	comp_code	type_code	Comp_Name	Comp_Desc
4	1	1	مبتدأ	يقع فى أول الجملة
4	2	1	خبر	يأتى ليتمم المبتأ جملة مفيدة
5	1	1	فعل	ما دل على عمل يحدث فى زمن معين
5	2	1	فاعل	اسم يأتى بعد الفعل ويدل على من فعل الفعل أو قام به

The primary key in this table is (comp_code) (the component or fragment no.) and is related to (comp_ID) in Tables 3 and 4.

Table 3
Types table.

c_code	Comp_ID	Types_ID	Types_Name
2	1	1	ضمائر المتكلم
2	2	2	ضمائر المخاطب
2	3	3	ضمائر الغائب

The primary key in this table is (Types_ID) (the type of the concept) and is related to (type_code) in Tables 2, 5 and 6 or (Types_ID) in Tables 3 and 4 respectively.

Table 4
The examples table.

Examples				
c_ID	Comp_ID	Types_ID	ex_no	Example
1	1	1	2	هذا عالم مشهور
4	1	1	1	الشواطئ ساحرة
5	1	1	1	يفرح الناجح لنجاحه

The primary key in this table is (C_ID) (the lesson no.) and is related to (concept_ID) in Table 1.

4. Main form: this form handles runtime events and reacts with the student's actions to answer such as: mouse click, dragging objects, and writing.

In AG_TUTOR, the Question Selector module was implemented using Java netbeans IDE 7.0, ODBC API & unlaces API, SQL, and Microsoft Access for knowledge base. The knowledge base of this module consists of concepts, problems and questions tables (Tables 1, 5 and 6) from the database. The student interacts with the system through a friendly graphical user interface which runs under windows environment.

Fig. 5 shows a snapshot from the Question Selector Module that shows a question from the lesson of the demonstrative nouns (أسماء الإشارة).

Table 5
Problems table.

Problems			
c_code	type_code	P_no	question_head
1	1	1	اسحب اسم الإشارة المناسب وضعه فى مكانه المناسب فيما يأتى:
4	1	1	أعرب الجملة التالية:

The primary key in this table is (P_no) (the question head no.) and is related to (P_no) in Table 6.

Table 6
Question table.

questions				
c_code	type_code	P_no	q_no	question
1	1	1	1	.... تلميذ(هذا - هذان- هؤلاء)
1	1	1	2	.... طفلة صغيرة(هذه - هذان - هؤلاء)
4	1	1	1	العصفور جميل
4	1	1	2	العلم نور

Table 7
Words table.

Words					
w_code	word	type	gender	count	anatomy
148	العصفور	1	1	1	2
149	جميل	1	1	1	5
150	العلم	1	1	1	4
151	نور	1	1	1	4
152	تلميذ	1	1	1	1
153	طفلة	1	2	1	1
154	صغيرة	1	2	1	5

Table 8
Classification table.

classify	
type	type_code
اسم	1
فعل	2
حرف	3

Table 9
Gender table.

Gender	
gender	gender_code
مذكر	1
مؤنث	2

Table 10
Count table.

Count	
count	count_code
مفرد	1
مثنى	2
جمع	3

Table 11
Anatomy table.

anatomy	
anatomy	code
انسان	1
حيوان	2
نبات	3
جماد	4
صفة	5

Fig. 5. A snapshot from the Question Selector Module.

6. The Expert Module

Different systems use different techniques to model the Expert Module. In the AG_TUTOR, the Expert Module uses available rules of the curriculum to get the answer of the questions. It simulates human experts in decision making or the instructor in education. This module contains the rules of the domain knowledge that help to get the correct answer of the question. In this phase, an Expert Module is implemented for the domain knowledge of the two parts of the curriculum that contain nine lessons (Brusilovsky, 2002; Ramesh and Rao, 2010).

The expert knowledge module comprises of facts and rules of the particular domain to be conveyed to the student, i.e. the knowledge of the experts, which is generally derived from people who have years of experience in the domain.

In the AG_TUTOR the Expert Module is using the production rules for representing the knowledge base for answering the question. The Expert Module uses this knowledge to guide other parts of the system.

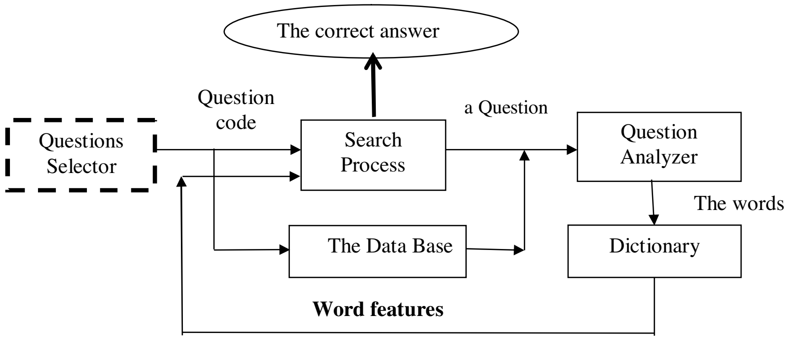

Fig. 6. Design of the Expert Module.

6.1. Design of the Expert Module

The methodology of this module is established on the idea of defining some features for the words that are used in the exercises that will be shown to the student. These words are put in a dictionary with their features as mentioned before. These features are represented through a specific code for each one as shown in Table 7 (the dictionary) (Ramesh and Rao, 2010).

As an example: the word “"تلميذ (pupil)” has the features noun, male, single and human. It will take a code: “1 1 1 1” and so on.

The idea of defining the features of each word is valuable because:

- This module can deal with singular, double and plural nouns, male and female. Also, an Arabic word may be one of different categories and for types. The types are noun–verb–particle, etc. More questions can be added and presented to the student without need to change the module.
- This idea will help us in implementing the student module in the next phase that will check the student answer and medicate his problems. This may enhance the system in recognizing the student belief

During the program running, the Question Selector chooses a random question and displays it to the student. Moreover, The question coding is sent to the Expert Module as an input. And the output of this module is the correct answer.

As shown in Fig. 6, the Expert Module consists of many components mainly:

- *Data Base*: is a group of tables that consists of Tables 5–11. Such tables contains the main data to present questions to the students as well as some relations among the question constructs.
- *The dictionary*: as mentioned before, the dictionary is a collection of words stored in Table 7 which has a huge number of words with their feature codes for each one. The dictionary till now contains about 700 words.
- *Question Analyzer*: it takes the question, analyzes it to words and gets the features of the words of that question by searching the database.
- *Search Process*: it has two searching functions:
  - Taking the question code; the output of the Question Selector; to get the question itself by searching the data base tables.
  - After analyzing the question, the system searches the dictionary to get the features of each word in the question. So, the system can get the correct answer that matches with those features.

Fig. 7 shows a snapshot from the Expert Module that shows the same question in Fig. 5 with the correct answer.

Fig. 7. A snapshot from the Expert Module.

7. Conclusion and discussion

In this research work, a part of an intelligent tutoring system is presented to enhance the educational process. The work is dedicated to those student at the fourth grade of the elementary school in Egypt. The work involves three main modules mainly: the Tutoring Module, the question selector module and the Expert Module. The tutoring system is developed and implemented to increase and/or enhance the skills to the students in the adopted domain. The system is briefly explaining the chosen domain which is the Arabic grammar for the fourth grade student. The system supports the explanation process using the multimedia facilities such as images, sound, animation and text as well. The system involves also the question analysis as well as the domain knowledge expert. The knowledge of the Expert Module is important for presenting the correct answer. The system also is supported by a graphical user interface to ease the interaction with the students. This work is considered prototype as it can be scaled up to cover other subject curricula. Finally, the system was implemented on the PC/machine Access for the database design. Using the Java, C# programming languages, and photoshop and illustrator software packages for the graphic and multimedia activities.

References

Akbulut, Y., Cardak, C.S., 2012. Adaptive educational hypermedia accommodating learning styles: a content analysis of publications from 2000 to 2011. Comput. Educ. 58, 835–842, Elsevier.

Al Emran, M., Shaalan, K., 2014. A survey of intelligent language tutoring systems. In: International Conference on Advances in Computing, Communications and Informatics (ICACCI), Delhi, India, pp. 393–399.

Anohina, A., 2007. Advances in intelligent tutoring systems: problem-solving modes and model of hints. Int. J. Comput. Commun. Control II (1), 48–55.

https://en.wikipedia.org/wiki/Production_system_(computer_science).

https://en.wikipedia.org/wiki/Intelligent_tutoring_system.

Brusilovsky, P., Cooper, D.W., 1998. ADAPTS: adaptive hypermedia for a web-based performance support system. In: The Second Workshop on Adaptive Systems and User Modeling on the World Wide Web, Pittsburgh, USA.

Brusilovsky, P., 2002. Developing Adaptive Educational Hypermedia Systems: From Design Models to Authoring Tools. School of Information Sciences, University of Pittsburgh, Pittsburgh, PA 15260, USA.

Chowdhury, G.G., 2003. Natural language processing. Annu. Rev. Inf. Sci. Technol. 37 (1), 51–89.

Dzikovska, M.O., Moore, D., Bental, J.D., Steinhauser, N.B., Campbell, G.E., Farrow, E., Callaway, C.B., 2010. Intelligent tutoring with natural language support in the Beetle II system. In: Sustaining TEL: From Innovation to Learning and Practice. Springer, Berlin, Heidelberg, pp. 620–625.

Mahmoud, M.H., Abo El-Hamayed, S.H., 2015. An expert module of an intelligent tutoring system. In: The Proceedings of the Asian Conference on Education and International Development, Osaka, Japan, pp. 359–370.

Please cite this article in press as: Mahmoud, M.H., Abo El-Hamayed, S.H., An intelligent tutoring system for teaching the grammar of the Arabic language. J. Electr. Syst. Inform. Technol. (2016), http://dx.doi.org/10.1016/j.jesit.2016.04.001

Mills, C., Dalgarno, B., 2007. A conceptual model for game based intelligent tutoring systems. The Proceedings of Ascilite Singapore: Mills and Dalgarno, 692–702.

Nkambou, R., Bourdeau, J., Mizoguchi, R., 2010. Advances in Intelligent Tutoring Systems, vol. 308. Springer-Verilog, Berlin, Heidelberg, pp. 63–80.

Prentzas, J., Hatzilygeroudis, I., 2009. A survey on recent patents regarding intelligent educational systems. Recent Pat. Comput. Sci., 214–222.

Ramesh, V.M., Rao, N.J., 2010. Tutoring and expert modules of intelligent tutoring systems. In: A Technical Report Presented to the International Institute of Information Technology, Electronics City, Bangalore.

Roth, F.H., Waterman, D.A., Lenat, D.A., 1983. Building Expert Systems. Addison-Wesley Publishing, ISBN 0-201-10686-8.

Samuelis, L., 2007. Notes on the components for intelligent tutoring systems. Acta Polytech. Hung. 4 (2), 77–85.

Shaalan, K., 2003. Development of computer assisted language learning system for Arabic using natural language processing techniques. Egypt. Inform. J. 4 (2), 131–155.

Vanlehn, K., Jordan, P.W., Rose, C.P., Bhembe, D., Boettner, M., Gaydos, A., Makatchev, M., Pappuswamy, U., Ringenberg, M., Roque, A., Siler, S., Srivastava, R., 2002. The architecture of Why2-Atlas: a coach for qualitative physics essay writing. In: The Proceedings of 6th International Conference on ITS, Spain, pp. 158–167.

Please cite this article in press as: Mahmoud, M.H., Abo El-Hamayed, S.H., An intelligent tutoring system for teaching the grammar of the Arabic language. J. Electr. Syst. Inform. Technol. (2016), http://dx.doi.org/10.1016/j.jesit.2016.04.001