Arabic Expert System Shell

Venus Samawi[1], Akram Mustafa[1], and Abeer Ahmad[2]
[1]Department of Computer Science, Al-albayt University, Jordan
[2]Department of Computer Science, AL-Nahrain University, Iraq

Abstract: *Most expert system designers suffer from knowledge acquisition complications. Expert system shells contain facilities that can simplify knowledge acquisition to make domain experts themselves responsible for knowledge structuring and encoding. The aim of this research is to develop an Arabic Expert System Shell (AESS) for diagnosing diseases based on natural language. The suggested AESS mainly consists of two phases. The first phase is responsible for automatic acquiring of human expert knowledge. The acquired knowledge is analyzed by Arabic morphological system. The Arabic morphological system analyzes the given Arabic phrase and finds the required keywords (roots). The suggested system is provided with the required domain dictionary to be used by the Arabic morphological system. The second phase is concerned with the design of inference engine together with user interface (based on natural language) that uses a backward chaining method (end-user interface).When AESS tested by experts and end users, it was found that AESS performance in constructing Knowledge-Base (KB) and diagnosing problems was very exact (the diagnostic ability of AESS is 99%.). Merging of morphological system with knowledge acquisition is very effective in constructing the target KB without any duplicate or inconsistent rules. The same technique could be used to build expert system shell based on any other natural language (English, French, etc.). The only difference is to build morphological system suitable to that language in addition to the desired domain dictionary.*

Keywords: *Expert system, knowledge acquisition, knowledge engineering, diagnosing expert system, Arabic morphological system.*

Received October 25, 2010; accepted March 1, 2011

1. Introduction

An Expert System (ES) is a methodology and programming approach that attempts to provide answers for complicated problems, or clarify uncertainties where human domain experts would need to be consulted. Expert system mainly consists of four components: Knowledge-Based (KB), inference engine, user interface, and knowledge acquisition [9, 17]. The KB consists of rules and facts that are acquired from knowledge, opinion, and experiences of experts. It defines the knowledge presentation schemes which specify the relationship between rules and facts representing related experiences [17, 20]. The process of building KB is part of knowledge engineering [11]. The inference engine provides a control structure that uses information supplied by the user and applies the knowledge in the KB to obtain the solution for a particular problem [20]. Knowledge acquisition is responsible for entering new rules to the rule-base, and updating the existing rules [9]. Expert system needs a complete development environment called shell to build KB applications and maintaining them. This will help human expert to construct the suitable KB without any help from expert system designer [13].

Shells contain facilities that can simplify knowledge acquisition. It provides a step-by-step methodology and a user-friendly interface for a knowledge engineer. The friendly interface allows the domain experts themselves to be directly involved in structuring and encoding the knowledge [11]. Non-programmer expert can understand shells without understanding the lengthy learning process [13]. El-Sadany and Hashish [7] developed a rule-based expert system. They used the associated morphological information to generate Arabic verbs and derivational nouns from stems. Al-muhtaseb and Aref [4] developed (Khabeer) an object-oriented Arabic expert system shell. Khabeer was used as a machine translation tool. Several phases of machine translation are demonstrated. These phases include lexical and morphological analysis, syntax analysis, knowledge representation and sentence generation. Al-Khateeb [1] developed a general diagnosing expert system shell at which the KB may contains the expertise of many experts. It was found that Rule-duplicate may occur in general diagnosing expert system shells. Pamela and Xenogene [21] designed eGanges which is a map-based, expert-friendly expert system shell that allows construction of nested rule or procedure maps using glosses of nodes in the maps. Gloss options include links between nodes in the same map and between parallel maps. eGanges is an expert system shell, mainly for the domains of law, quality control management, and education. Hussein *et al*. [11] proposed framework model for expert system shell which is based on the integration of rule-base and the case-based forms using blackboard which facilitate applying more than one problem solving methods and search techniques in inference engine of the expert system shell. The rule-base and case-base formats have been converted into tables.

Motivated by the need for shell that allows domain expertise to construct a suitable KB using natural language without any rule-duplication, an expert system shell for diagnosing using Arabic language is suggested. The suggested expert system shell helps experts (doctors for example) to construct the KB without the need for the expert system designer. Multiple KB systems could be constructed by different experts through providing suitable knowledge acquiring. The aim of this work is to avoid the problems of duplicate and inconsistent rules within the KB. This will be gained by using a morphological system to analyse the acquired knowledge. Domain dictionary is built to support the morphological system in finding the required keywords (roots). The researchers choose to construct Arabic morphological system that uses domain dictionary to find the required keywords (roots) since Arabic language is considered as a highly inflected language and thus it is one of the difficult natural languages to be handled automatically [22]. AESS was implemented using Visual Prolog.

2. Expert System Shell

An expert system shell may be either domain specific shell or general-purpose shell. Domain specific shell is used for developing specific application domains. General-purpose shell offers flexibility and generality to solve different types and areas of problems. It provides the following benefits [6, 14, 15]:

- Can be used for different applications.
- Allows fast expert system construction.
- No need for high programming skill.
- Cost reduction.

Shell system composed of the same ES components except the KB. Figure 1 illustrates the ES components and describes the shell system [6].

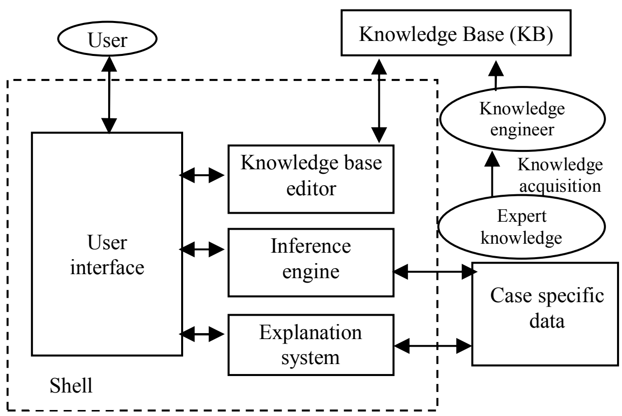

Figure 1. Expert system shell.

The first step in building any expert system is Knowledge acquisition. It is the process of transforming the extracted knowledge (domain experts' knowledge) to forms suitable to be saved in the Knowledge-Based System (KBS). Many researchers and practitioners have identified this process as a bottleneck that restricts the development of expert system. The knowledge should be validated and verified to improve its quality. Then the acquired knowledge is organized and encoded in the KB [15, 16].

In this work, Acquiring knowledge from domain experts is via friendly user interface using menu driven and natural language. The user interface allows the expert to enter Arabic phrases. The acquired knowledge is analysed by Arabic morphological system and finds the required keywords (roots) before saving then in the KB.

3. The Developed Arabic Morphological System

In linguistic, morphology is the study of the internal structure of words. In other words, morphology is simply the branch of linguistics that studies patterns of word-formation taken in their different uses and constructions. It also attempts to formulate rules that model the knowledge of the speakers of those languages [7, 18].

Arabic suffers from huge number of compound words. These compound words result in high Out-Of-Vocabulary (OOV) when using finite-sized lexicons. It contains words that are highly infected and derived. Therefore, the compound words need special techniques and algorithms to solve its morphological problems. In this work, the main objectiveof developing the Arabic morphological system is to convert an Arabic phrase into a list of keywords (roots) to overcome the rule duplication problem. Different forms of the same word (or even its synonym) will have only one representation (its root). The keywords of the entered phrase are compared with the existing rules to check if the input phrase represents a useful or duplicated knowledge. Finally, refine the knowledge [5, 7] The developed Arabic morphological systemconsists of two modules. The computational lexicon module and analysis module.

3.1. Computational Lexicon Module (Dictionary)

The dictionary module contains Arabic roots of domain words with their associated features in addition to noise words and popular synonyms for the domain words. The root is an ordered sequence of three or four valid Arabic alphabet i.e., the root can be either trilateral or tetra-literal [10].

3.2. Analysis Module

The analysis module contains the classical Arabic morphological rules grammar through removing the affixes (prefixes, infixes, and suffixes) from a word to get its stem. There are many morphological rules. For example, the rules used in the derivation of the verbs, and rules that are used in the derivation of the nouns from verbs as shown in equations 1 and 2 [2, 3, 19].

$$Verb=prefixes+stem+suffixes \quad (1)$$

$$Noun=prefixes+stem+suffixes \quad (2)$$

Arabic verbs and nouns are generated from stems; stem is formed by substituting the characters of the root to certain forms, called measures (scales) [2, 8]. The measure is a general module composed of an ordered sequence of characters. Some of these characters are "constants" and some are "variables". The "variable" characters are to be substituted with the characters of an Arabic root to generate a word called the "stem". For example the word (takateba) "تكاتب" is formed after applying the measure form (tafa'ala) "تفاعل" to the stem (kataba, write) "كتب" . This can be done by adding "ت" at the beginning of the root "كتب", then insert "ا" after "ك" as shown in example 1. The same thing is done to get the word (maktoob, written) "مكتوب" from the same root "كتب" after applying the measure form (mafool) "مفعول" as in example 2 [10, 19].

Examples 1:

ت ف ا ع ل= ت ك ا ت ب

ف ع ل (فعل)= ك ت ب (كتب، write)

Examples 2:

م ف ع و ل= م ك ت و ب

ف ع ل (فعل)= ك ت ب (كتب، write)

The analysis module has inference engine mechanism that uses the Arabic morphological rules and computational lexicon module to perform word analysis. When an Arabic phrase is obtained from human expert, the analysis module converts it into a list of keywords (roots).

4. The Architecture of AESS

AESS is mainly consists of four subsystems as illustrated in Figure 2, each of which is used to perform specific task.

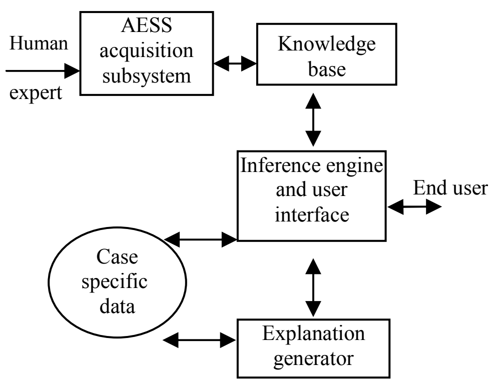

Figure 2. AESS architecture.

4.1. AESS Acquisition Subsystem

AESS acquisition subsystem is shown in Figure 3. The developed acquisition subsystem obtains knowledge from human expert by "automatic knowledge acquisition and knowledge engineer". Each Arabic phrase is analyzed and converted into list of keywords by "Arabic morphological module". "Refine knowledge" is used to check knowledge before storing it. The following subsections are the detailed explanation of each module.

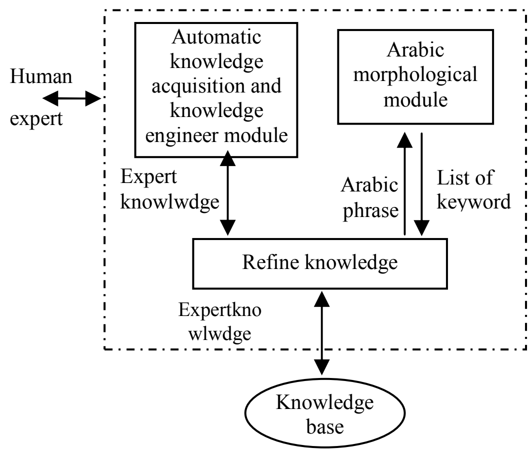

Figure 3. AESS acquisition subsystem.

4.1.1. Automatic Knowledge Acquisition and Knowledge Engineer Module

The automatic knowledge acquisition and knowledge engineer module is the interface between human expert and AESS. It enables human expert to create his KB, and input information about it. The input KB is composed of set of rules ("IF..Then" rule). This module decomposes the input rule into three parts (as shown in Figure 4, one or more of related premises (question or sub-conclusion), conclusion, and certainty factor. The explanation for each question (why, and how) will be acquired from human expert.

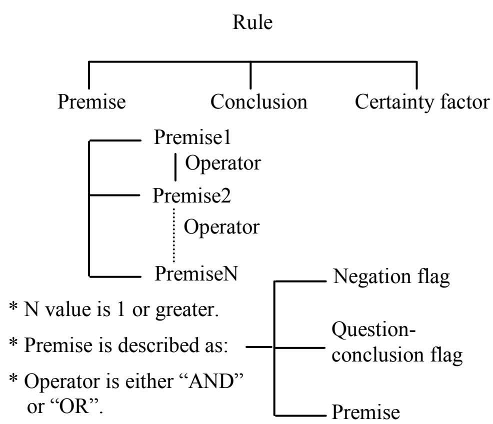

Figure 4. The representation of the input rule.

AESS interface is constructed depending on menu-driven and message windows. Menu is used to give the human experts the ability to choose the function that they want to perform. Also human expert has the ability to input his information and knowledge by making selection from an exist list or writing via edit controls (using natural language). AESS allows human expert to updatethe KB (change the content of any rule, change a specific conclusion or question, delete unwanted rule, delete not useful premises and conclusions).

4.1.2. Refine Knowledge

The system receives human expert knowledge from “automatic knowledge acquisition and knowledge engineer”, sends Arabic phrases (questions and conclusions) to the “Arabic morphological module”, and then receives list of keywords for each Arabic phrase. To refine knowledge:

- Validation task is used to prevent duplication and other errors that may happen as a result of updating KB. If the new knowledge (rule, question, or conclusion) is similar to an existing one, then human expert will be asked to if they are the same or not.
- Human expert sends knowledge to the KB, each question and conclusion will be found in two forms. First, the actual string which is obtained from human expert. Second, its corresponding list of keywords.

4.1.3. Arabic Morphological Module

AESS needs morphological module to accomplish two objectives. Discover the matching in meaning which introduce duplication, and discover the typing errors. AESS Arabic morphological module is decomposed into the main components of an Arabic morphological system (dictionary and rules analysis) as shown in Figure 5.

AESS dictionary is decomposed into domains. It provides separate dictionary for each domain of knowledge. Each dictionary contains roots (verbs), nouns, noise words, and synonym words (which have the same meaning related to this domain). Experimentally, AESS provided with medicine-domain dictionary.

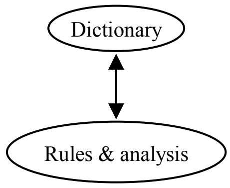

Figure 5. AESS architecture.

AESS Arabic morphological module performs analysis process, which depends on the classical Arabic morphological rules (grammar). It uses the grammar of deriving verbs in different formulas, and the grammar of deriving nouns from verbs. Stems are generated by substituting the characters of the root to certain measures (scales). The verb-measures and noun-measures are shown in Figure 6 and Table 1 respectively [2, 8, 10]. The verb-measures shown in Figure 6 are in the past tense. To make a verb in present tense, it must start by one of the following letters {“T” “ت”، “E” “ي”، “N” “ن”، “A” “ا”}.

As illustrated in equations 1 and 2, prefixes and suffixes could be attached to stems. Equations 3 and 4 show the decomposition of prefixes and suffixes.

$$Verb=prefix0+prefix1+stem+suffix0+suffix1 \qquad (3)$$

where

Prefix0={“ف”, “و”}, Prefix1={“س”, “ل”},
Suffix0={“ت”, “نا”, “ا”, “وا”, “ين”، “ي”, “ن”,“ان”, “ون”, “و”}
Suffix1={“ك”, “هم” “ني”, “هما”, “هن”, “هنّ”, “ه”, “ها”}

$$Noun=prefix0+prefix1+prefix2+prefix3+stem+ suffix0+ suffix1+suffix2 \qquad (4)$$

where

Prefix0={“ف”, “و”}, Prefix1={“ال”},
Prefix2={“ل”, “لل”}, Prefix3={“ب”، “ك”},
Suffix0={“ية”, “ة”}, Suffix1={“ت”} and
Suffix2={“ها”, “ك”, “ه”, “ي”, “هم”, “هن”, “هنّ”, “نا”, “كم”}

Table 1. Stems of nouns derived from verbs.

الأسماء المشتقة من الفعل Nouns That are Derived from Verb "Fi'l" (فعل)									
المصادر Gerunds					اسم الفاعل و المفعول Subjective and Objective Nouns	صيغة المبالغة Exaggeration Forms	اسم التفضيل Preference Nouns	اسم الزمان والمكان Durational and Positional Nouns	اسم الآلة Nouns that Describe Machines
مصدر الثلاثي Trilateral Gerund	مصدر غير الثلاثي Non-Trilateral Gerund	مصدر ثلاثي مزيد Affixed Trilateral Gerund							
		1 Affix	2 Affixes	3 Affixes					
فعال	فعلل	افعال	افتعال	انفعال	فاعل	فعّال	افعل	مفعل	مفعال
فعلة	فعللة	تفعلة	تفعلل	افعلال	افعل	مفعيل	فعلى	مفعلة	
فعيل	فعلال	افعلة	تفاعل	استفعال	مفعول	فعّيل			
فعالة		تفعيل				فاعول			
فعول		مفاعلة				فاعلة			
فعلان						فعّالة			

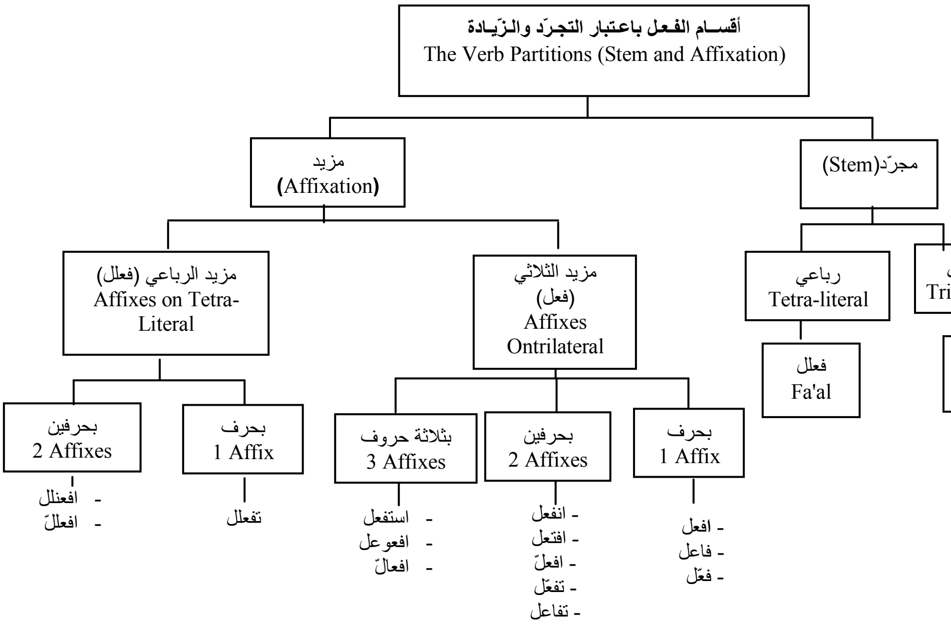

Figure 6. Classification of verbs in Arabic language.

The cases of prefixes and suffixes are described in Figures 7 and 8 with examples.

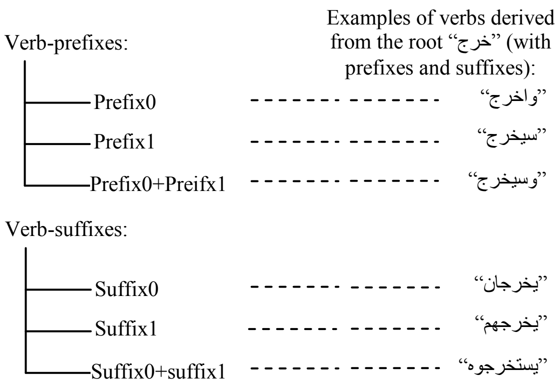

Figure 7. Verb-prefixes-and-suffixes treated by AESS.

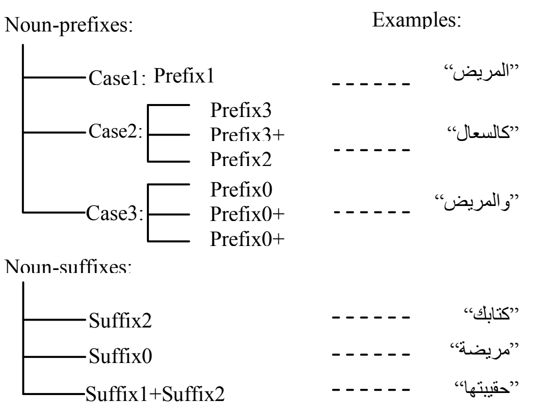

Figure 8. Noun-prefixes-and-suffixes treated by ESS.

AESS analysis process has inference engine mechanism that uses the dictionary and the previous grammars to perform word analysis. The following steps illustrate the analysis process. Steps are summarized by the two examples (form 1 and form 2) given in Table 2, at which two different forms of the same input phrase.

Table 2. Steps to convert an input phrase into list of keywords.

	Form 1		**Form 2**	
	Arabic	**English**	**Arabic**	**English**
Step 1: Input Phrase	ضغط الوريد البابي مرتفع	The portalvein pressureis high	يوجد ارتفاع في ضغط الوريد البابي	There is high pressure in portal vein
Step 2: Tokenizing	["ضغط"، "الوريد"، "البابي"، "مرتفع"]	["the, "portal", "vein", "pressure", "is", "high"]	["يوجد"، "ارتفاع"، "في"، "ضغط"، "الوريد"، "البابي"]	["there"," is", "high", "pressure", "in", "portal", "vein"]
Step 3: Removing Noise Words	["ضغط"، "الوريد"، "البابي"، "مرتفع"]	["portal", "vein", "pressure", "high"]	["ارتفاع"، "ضغط"، "الوريد"، "البابي"]	["high", "pressure", "portal", "vein"]
Step 4: Substitute Words with its Stem	["ضغط"، "ورد"، "باب"، "رفع"]	["portal", "vein", "press", "high"]	["رفع"، "ضغط"، "ورد"، "باب"]	["high", "press", "portal", "vein"]

- *Step 1. Tokenizing:* This step converts the input phrase (string) into a list of words (list of strings). First, the tokenizer eliminates unnecessary blank-spaces and punctuation marks. The resultant string is converted into a list of words.
- *Step 2. Removing Noise:* This step checks the list of words. Remove the noise (unnecessary) words based on the noise words stored in the dictionary. Example: Unattached prepositions such as: (("on", ala, "على"), ("from", min,"من"), ("in", fi,"في" ")) Connectives such as: (("or", a'au, "او"), ("or", m, "ام"), ("but", bal, "بل"), ("then" thum'a "ثم")). Others, such as: ("never", kut, "قط"), ("may", kud, "قد"). The above examples are noise-words for all domains of knowledge. Also there are words that are considered as noise toward a specific domain of knowledge.

- *Step 3. Get Keyword:* The job of this step is to obtain the roots of the remaining words by the use of morphological knowledge, roots that are stored in the dictionary, and the classical Arabic morphological rules mentioned before. There is no harm of using domain dictionary that contains limited vocabulary since the dictionary concerning a specific subject. The using of domain dictionary will be of acceptable size leading to fast stemming process.
- *Step 4. Error Checking:* If there is no keyword for a specific word, then error-checking is called to check the following two possibilities:
  - The word may be wrong. So, call the spell-checker procedure provided in this work to get the suggestions for the right word, and ask human expert to choose the correct one. For example if the input word is (kab "كاب"), the list of suggestions will be [(kootoob "كتب"), (kitab "كتاب"), (katib "كاتب"), (kaaseib "كاسب")].
  - It could be a new-word (its keyword does not belong to the dictionary). Ask the expert to determine its type (verb or noun) and add it to the dictionary.
- *Step 5. Replacing Synonyms:* This step checks each keyword in the list. If it has a synonym in the dictionary, then replace it with its synonym, such as:

(pain=Ache) وجع=ألم
(Lessen=decrease=minimize) خفَ=قلَ=ضعف

4.2. AESS Knowledge Base

AESS KB stores: "questions", "rules", "conclusions". External databases (provided by VProlog) are used to store ("questions", "rules", and "conclusions") in three separate files. B+trees indexing method is used to quickly look up information stored in the external chains. Each new question is stored in "question" file together with its explanation as the form below:
(*question-number, question-code, list-of-keywords, question-explanation*).
Each new conclusion is stored in "conclusion" file together with the numbers of its rules as the as the form below:
(*conclusion-number, conclusion-code, list-of-keywords, list-of-rules-numbers*).
If-section of each new rule is stored in "rule" file together with its rule-number and the Certainty Factor (CF) of the rule as the as the form below:
(*rule-number, list-of-subsequent-premises, CF*).
Each premise is represented by three components. First, flag recognizes whether premise is question or sub-conclusion. Second, is flag recognizes whether premise is negated or not. Third, is key-number of this premise. Experimentally, AESS provided with KB "Abdominal Pain". It belongs to medicine-domain.

4.3. AESS Inference Engine and User Interface Subsystem

The inference engine performs the task of communicating with the end-user (inexpert user). In AESS, reasoning is based on Stanford certainty factor method [9] to reflect uncertainty. Certainty can be applied to the rule (obtained from human expert), and to each question by the end-user. A dialog of (question/answer) begins between AESS and end-user. Menu driven is used to give the end user the ability to answer with ("yes", "no", "yes with certainty", "do not know", "why"). "yes" means that the question has the certainty of 1. "no" means the certainty is -1. "yes with certainty" gives end-users the ability to input certainty (they can choose it from the list). "do not know" means certainty is 0. And "why" means that the end-user wants to know why AESS asked him a specific question. AESS uses backward chaining technique, which selects a goal and finds the combination of rules that can achieve it. Each rule contains a list of subsequent premises. AESS inference engine tries to prove each premise. These premises may themselves be conclusion of other rules (sub-conclusion). If its result is not found in the working memory (case specific data), then it will be a new sub-goal, and the same procedure is applied recursively. AESS stores the checked conclusion together with its certainty in the working memory. If premise is a question, then search its answer in the working memory. If it is not found then end-user must answer the question. The answer is stored in the working memory. To achieve this method, AESS gets the user answer with its certainty, and then gets the certainty for each rule. Then combine-rule is used to calculate a single combined certainty.

The inference engine starts with the main (last) conclusion and tries to prove it. If it could not be proved, then try to prove the previous one and so on until the goal is achieved or the KB is finished.

Example:
Rule1: If fever and cold then influenza (0.9).
Rule2: If fever and cough then influenza (0.85).
Assume the user answers are as following:
Fever (yes/1).
Cold (yes with certainty 0.94).
Cough (yes with certainty 0.8).
Then get the certainty for each rule:
*Rule1: min [1, 0.94] * 0.9 → CF (rule1)=0.846*
*Rule2: min [1, 0.8] * 0.85 → CF (rule2)=0.68*
Finally, apply combine-rule to calculate single combine certainty [9]:
*CF(influenza)=CF(R1)+CF(R2)–(CF (R1)*CF (R2))*
*=0.846+0.68–(0.846 *0.68)=0.95*
Note that the combination of two rules together is stronger than each of them separately.

4.4. AESS Explanation Generator

The ability to explain reasoning is usually considered as an important part of any expert system. An explanation facility is useful on many levels. For example, it can help knowledge engineers to debug and test the system during development process. Also it can provide assurance to the users that system's knowledge and reasoning process is appropriate [12]. AESS provides two explanation facilities, "Why" and "How". To implement "Why", the explanation obtained from human expert is stored in "question" file. When end-user selects "Why" explanation, the inference engine searches "question" file for that question and displays its explanation to the end-user. "How" explanation is implemented by storing each question in working memory together with its answer. Intermediate results (sub-conclusions) are also stored together with their certainty factor. When the end-user selects "How", the system will display each asked question with its answer. Also the intermediate results in order will be displayed until the goal is achieved.

5. Experimental Results

AESS provides user-friendly interface (consists of menus and windows). It provides many facilities to help both human experts and end-users with writing, editing, and updating the knowledge they enter.
To test the AESS behavior and results, AESS was used to construct two diagnosing ESs in two different domains (medical and chemical). The two ESs knowledge-bases were constructed by domain human experts. There feedback was positive. The two systems were tested by end-users. When AESS tested by experts and end users, it was found that AESS performance in constructing KB and diagnosing problems was very exact. To study the behavior of AESS, the characteristics of the medicine-domain dictionary and the KB "Abdominal Pain" are stated bellow:

- Fill the dictionary with words needed in medicine-domain. AESS is provided with the ability of learning new words. The human expert may use words not exist in the dictionary. In this case, AESS allows the human expert to add the new words to the dictionary. If the word "عسر" , for example, does not included within the dictionary, and the following phrase is acquired "عسر الهضم", AESS consider "عسر" as wrong word (typing error) and suggest the right word "عشر". AESS provides the ability of choosing the suggested right word, or adds the word "عسر" to the dictionary.
- External database of visual-Prolog is used in dictionary construction leading to fast retrieval and high storage. The dictionary contains 1200 word (roots, nouns, noise words, and domain noise words). New words could be added and the dictionary size could grow without any problem.
- "Abdominal Pain" deals with abdomen diseases. It consists of 41 diseases with 60 (consistent and non-duplicated) rules. There are135 symptoms in these rules. AESS is provided with the ability of KB updating. Human expert can add new disease, delete existing disease, or edit disease.
- Natural language is used to enter the symptoms of a disease. As known, different users could use different phrases to enter the same symptoms. AESS has the ability to recognize these symptoms and give correct diagnosing answer with about 99% accuracy. Some examples of similar symptoms illustrated using different phrases and discovered (i.e., recognized as the same phrase) by AESS are shown in Table 3:

Table 3. Examples of similar symptoms entered by different users.

	Symptoms (English)	**Symptoms (Arabic)**
Example 13 Different Users	Pain in the abdomen in the epigastric region	الألم في البطن في منطقة الشرسوف
	Pain in the epigastric region of the abdomen	الألم في المنطقة الشرسوفية من البطن
	Epigastric abdominal pain	الألم بطني شرسوفي
Example 22 Different Users	Pain appears during sleep	يظهر الألم أثناء النوم
	Pain appears when I sleep	الألم يظهر عندما أنام
Example 33 Different Users	Sudden pain	الألم مفاجئ
	Pain appears suddenly	يظهر الألم بشكل مفاجئ
	Pain appears quickly	الألم يظهر بسرعة
Example 43 Different Users	High Fever	ارتفاع درجة الحرارة
	The fever is high	الحرارة مرتفعة
	Fever	حمى
Example 52 Different Users	Jaundice	يرقان
	There is jaundice	يوجد يرقان

6. Conclusions

It is very important to allow human experts to construct the KB of expert systems. This is cannot be Accomplished without using Experts System Shell (AESS). The developed system merges Arabic morphological system with knowledge acquisition to handle the rule duplication and rule inconsistency in the constructed KB. The Arabic morphological system is associated with domain dictionary that contains the expert domain vocabularies (word stems associated with their synonyms). The domain dictionary helps the morphological system in finding accurate keywords swiftly since it only requires dealing with the related domain words. AESS allows the human expert to add, update and modify the dictionary. The dictionary is constructed using external database provided by Visual Prolog making it scalable and fast. It is important to point out that the suggested Arabic morphological system could be replaced by any other natural language morphological system (English for example) associated with the suitable domain dictionary.
AESS is considered as general expert system shell used for solving diagnosing problems. The generality of AESS could cause losing of some diagnosing systems

power. For example, in medical field, one could lose information (concerning historical medical information, laboratory analysis information, and imaging procedures information) about the patients which could be helpful in the diagnosing process. On the other hand, the generality issue allows users to use AESS to build different ESs, and in turns they could be used to diagnose different problems. The diagnostic accuracy of AESS reaches 99%. AESS used fast inference engine strategy (backward chaining) and B+tree indexing method which is quickly look up information stored in external databases. Large KBs could be constructed without any space or time problem. Finally, AESS is provided with user-friendly interface to simplify the system usage for both human expert and the end user. As future work, semantic analyzer might improve the system performance. Also using speech recognition system to allow end users use voice instead of writing.

References

[1] Al-khateeb I., "Design and Implementation of Diagnosing Expert System Shell," *MSc Thesis*, Department of Computer Science, Al-Nahrain University, 2002.

[2] Al-kholy A., *قواعد تحويلية للغة العربية*, دار الفلاح للنشر والتوزيع, Jordan, 1999.

[3] Al-mashhadany K., "Building Diagnoses Expert System Shell Using Arabic Interface," *MSc Thesis*, Department of Computer Science, Al-Nahrain University, 2005.

[4] Al-muhtaseb H. and Aref M., "Khabeer As A Machine Translation Tool," *in Proceedings of the 1st Workshop on Information and Computer Science*, Saudi Arabia, pp. 101-109, 1996.

[5] Bing X., Kham N., Long N., Schwartz R., and Makhoul J., "Morphological Decomposition for Arabic Broadcast News Transcription," *in Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, France, pp. 1089-1092, 2006.

[6] Efraim T. and Jay A., *Decision Support System and Intelligent Systems*, NJ, A Simon and Schuster Company, 1998.

[7] El-sadany A. and Hashish A., "An Arabic Morphological System," *IBM System Journal*, vol. 28, no. 4, pp. 600-612, 1989.

[8] Essam A. and Abdullah B., "A Framework to Automate the Parsing of Arabic Language Sentences," *The International Arab Journal of Information Technology*, vol. 6, no. 2, pp. 191-195, 2009.

[9] George L., *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, Addison Wesley Longman, 2008.

[10] Hadi N., *الصرف الوافي*, الجامعة المستنصرية, Baghdad, 1989.

[11] Hussein O., Monzer Q., and Hazim F., "Framework Model for Shell Expert System," *International Journal of Computer Science and Network Security*, vol. 9, no. 11, pp. 56-68, 2009.

[12] Imam R., *مختار الصّحاح*, مكتبة لبنان, Lebanon, 1988.

[13] Jie Y., Chenzhou Y., and Xiaoli Z., "An Expert System Shell for Fault Diagnosis," *Journal of the International Federation of Robotics*, vol. 19, no. 6, pp. 669-674, 2001.

[14] Joseph G. and Gary R., *Expert Systems: Principles and Programming*, Thomson Course Technology, 2004.

[15] Keith D., *The Essence of Expert System*, Pearson Education Limited, 2000.

[16] Kendal L. and Creen M., *An Introduction to Knowledge Engineering*, Springer-Verlag London Limited, 2007.

[17] Kiong W., Abd-latif R., Mohd Z., and Azwan A., "Expert System in Real World Applications," available at: http://www.generation5.org/content/2005/Expert_System.asp, last visited 2005.

[18] Laurie B., *Introducing Linguistic Morphology*, 2nd Edition, Georgetown University Press, Washington, 2003.

[19] Moulana M., *From the Treasures of Arabic Morphology*, 1st Edition, Academy for Islamic Research Madrasah In'amiyyah, South Africa, 2006.

[20] Norlela I., Amiruddin I., and Riza A., "An Overview of Expert Systems in Pavement Management," *European Journal of Scientific Research*, vol. 30, no. 1, pp. 99-111, 2009.

[21] Pamela G. and Xenogene G., "A Map-Based Expert-Friendly Shell," Bourcier D., (ED.), *in Proceedings of Legal Knowledge and Information Systems*, Amsterdam, pp. 51-60, 2003.

[22] Ryding K., *A Reference Grammar of Modern Standard Arabic*, Cambridge University Press, UK, 2005.

Venus Samawi is an assistant professor in Al-albayt University, Department of Computer Science. Al-albayt University, Jordan. She received her BSc from University of Technology at 1987, and the MSc, and PhD degrees from Computer Science Department in Al-Nahrain University (Saddam University previously) at 1992 and 1999 respectively. She supervises three PhD students in system programming, pattern recognition, and network security. She also leads and teaches modules at BSc, MSc and PhD levels in computer science. She is a reviewer in IEEE conferences (ICIEA 2009 and 2011). Her special area of research is AI, neural networks, genetic algorithms, and image processing.

Akram Mustafa received his the BSc degree in computer engineering in 1997, MSc degree in computer engineering in 1999 from the National Technical University of Ukraine. And PhD degree in computer engineering from the National Technical University of Ukraine in 2003. From 2001 to 2002, he was an instructor of Computer Science Department at Al-Yarmouk University, and from 2004 up to now ‹he is an assistant professor of Computer Science Department at Al-albayt University. He served as a chairman of the Department of Computer Science at Al-albayt University and Department of Information System. His research interests include network, image processing, security and artificial intelligence.

Abeer Ahmad is lecturer in the Department Of Computer Science, AL-Nahrain University, Iraq. She holds MSc degree in computer science from AL-Nahrain University, 2005 and BSc in computer science from AL-Nahrain University, in 2002. She gains the third Iraq scientific prize in 2002. She has more than 5 years of experience as lecturer. His research interests and AI and natural language processing.