A General Evaluation Framework for Text Based Conversational Agent

Mohammad Hijjawi
Faculty of Information Technology
Applied Science Private University
Amman, Jordan

Zuhair Bandar
School of Computing
Manchester Metropolitan University
Manchester, UK

Keeley Crockett
School of Computing
Manchester Metropolitan University
Manchester, UK

***Abstract*—This paper details the development of a new evaluation framework for a text based Conversational Agent (CA). A CA is an intelligent system that handle spoken or/and text based conversations between machine and human. Generally, the lack of evaluation frameworks for CAs effects its development. The idea behind any system's evaluation is to make sure about the system's functionalities and to continue development on it. A specific CA has been chosen to test the proposed framework on it; namely ArabChat. The ArabChat is a rule based CA and uses pattern matching technique to handle user's Arabic text based conversations. The proposed and developed evaluation framework in this paper is natural language independent. The proposed framework is based on the exchange of specific information between ArabChat and user called "Information Requirements". This information are tagged for each rule in the applied domain and should be exist in a user's utterance (conversation). A real experiment has been done in Applied Science University in Jordan as an information point advisor for their native Arabic students to evaluate the ArabChat and then evaluating the proposed evaluation framework.**

Keywords—Artificial intelligence; Conversational Agent and evaluation

I. Introduction

Different terms can be used to define a system has the ability to handle user conversations such as Conversational Agent (CA), dialog system and chatterbot. CAs are playing significant roles in different applications, for instance, in marketing, education, help desk, entertainment, e-commerce, information retrieval and generally in business [1].

Basically, it can be considered that the first try to build a CA was in 1950 by Alan Turing and it called the imitation game or Turing test [2]. Turing test tried to determine if a computer program could think or at least imitating a human behaviour. In the Turing test, an interrogator sends a series of teletype questions to a hidden participant through a computer link. Then the interrogator has to distinguish whether the hidden participant is human or machine based upon the hidden participant's replies [2].

Comparing what Turing expected in his article [2] and what we have today, we could conclude that Turing's expectations have not been met. Although, it is now more than sixty years since Turing stated his beliefs and despite the fact that computer storage capacities exceeded his request (1 GB), no computer program has been able to pass the Turing test(imitation game) successfully [3, 4]. According to [5], in order for a computer program to pass Turing test, the machine must acquire the same level of intelligence as a human in all cognitive tasks. However, since the first CA (imitation game or test [2]) which Alan Turing tried to make the machine to have chatting with human, several types has been raised. These types targeted different kinds of conversations starting from text typing conversation, spoken conversation and mixed among them conversation. Due to this difference, number of approaches has been proposed and used to develop CAs which are Natural Language Processing (NLP), Pattern Matching (PM) and Semantic Sentence Similarity (SSS) measures.

All of the CA's building approaches (except PM) are natural language dependant which means it needs to understand the targeted natural language. Understanding and computing the natural language is quit complex and needs different huge research effort from the language scholars before. Given this, most of CAs has been developed using the PM approach for its simplicity and as it natural language independent. In addition, the pattern matching is not expensive computationally as it does not need a complex pre-processing steps to understand the sentence (user's utterance). Consequently, number of CAs such as ArabChat [1], InfoChat [6], ELIZA [7] and ALICE [8] used this approach to handle conversations for applications deal with large numbers of users in a real-time environment like the Internet [1]. Basically, this approach (the PM) based on matching a conversation with a pre-structured patterns to find the suitable one. Then, the response that related to the best matched pattern will be replied [1] The NLP which is defined in computing as "the computational processing of textual materials in natural human languages" [9] is based on understanding a sentence. Technically, NLP based CAs uses grammar rules and a list of attribute/value pairs to extract the conversation's speech act type from the sentence [10]. Then, it use these extracted information to fill a template-based response [10]. However, extraction such information is not easy at all as it depends on many linguistic factors [10]. In a rich language especially the sematic languages such as Arabic, this extraction will be harder to process [3, 10]. The SSS approach is based on checking the similarity level in semantic between two sentences [11]; the first sentence is the conversation itself and the second is a scripted pattern inside the CA. the most closed pattern in semantic(meaning), its response will be replied as an answer to the conversation. The SSS approach is based on computational semantic based manual built databases such as WordNet [11, 12]. However, such database has been established in 2006 [11] and the research in SSS in general is still a young research area in the Arabic language [3, 12].

All of these approaches (NLP, PM and SSS) has different advantages and disadvantages as discussed in [3, 10-12]. However, most of the evaluation methodologies for these approaches has been done manually by checking the CA's logs [3]. Evaluating CAs is not an easy task as it depends on number of factors are not easy to measure [3, 4].

Generally, there are many types of systems that deal with text based sentences formed in a specific natural language, such as Information Retrieval (IR), Natural Language Processing (NLP), Question Answering (QA) and Conversational Agents systems. However, the evaluation process of these systems varies due to the differences in their working mechanism and their output. When researchers of IR systems want to evaluate their system, they might be interested in ranking the returned documents according to the entered keywords. Therefore, they usually use special metrics for evaluation, such as the "recall" (the percentage of retrieved documents that are relevant) and "precision" (the percentage of relevant documents that are retrieved) metrics [13]. NLP systems might be evaluated by comparing the output with a prebuilt perfect result document, called the "gold standard". Usually, QA systems use the same metrics as IR systems ("recall" and "precision" metrics). QA systems are IR systems with an extra processing module to analyse the retrieved documents and extract a response [13].

Applying such evaluation techniques to evaluate CAs may not be useful due to the differences in their working mechanism and their output. Although QAs and CAs have the same output (response for the entered utterance), they differ in their working structures.

There exists two primary approaches to evaluating CAs: objective and subjective evaluation approaches. The objective approach can be done without recourse to human judgment. This approach is based on systematic and scientific measures to evaluate a CA [14].Conversely, the subjective approach can be performed with a recourse to human judgment by asking him/her about his/her opinion of using a CA [15].

The subjective approach usually utilises a user questionnaire to evaluate the CA. This questionnaire might be used to ask the user (after using the CA) about several aspects of the CA such as the CA usability, naturalness or his/her overall satisfaction of using the CA. However, it is impossible to rely on user to give his/her opinion regarding CA internal components' performance as he/she has no idea about them. A special type of a CA evaluation, based on human (judges) to determine the most human-like CA among competitors, is the Loebner Prize competition [16].In 1990, the Loebner Prize was established in collaboration with the Cambridge Centre for Behavioural Studies. This prize aims to encourage researchers to develop Conversational Agents. The Loebner competition uses expert human judges to evaluate the competing CAs using the Turing test. Passing the Turing test means that the program's responses should be indistinguishable from human conversation. This method of evaluation is not academically rigorous, and not all CAs can participate. Since the Leobner competition was established, many Conversational Agents have competed for it and not one CA has passed the Turing test. Unfortunately, some of these Conversational Agents focus merely on passing the test, rather than on advancing the field of Conversational Agents [4].

The objective evaluation approach evaluates a CA as a whole system (black box approach) or evaluates the CA's components individually (glass box approach). A black box evaluates the system as a whole, based on user satisfaction. This is usually done by evaluating inputs and outputs without considering any internal details[17]. The black box focuses on the performance of the system in terms of number of aspects, such as achievement task and the cost of that achievement [17]. The glass box deals with internal details by evaluating the individual components of a system [18]. An example of the glass box approach to evaluation is undertaken to measure the error rate on the sentence recognition module which is included in a spoken CA called ARPA [18]. A black box evaluation approach was used to evaluate the SUNDIAL CA [19]. This approach determined the SUNDIAL‘s user satisfaction by determining the task and its cost. The cost of the determined task might be based upon number of utterances needed to achieve the task, the elapsed time to complete the task and the quality of interaction among conversation entities.

PARADISE [14] (PARAdigm for DIalogue System Evaluation), is aa framework used for evaluating spoken CA. PARADISE relies on a comparison between agents through achieving the maximum user satisfaction.

Maximum user satisfaction means maximum task success with the minimum cost. PARADISE measures the task success per dialogue or sub-dialogue by determining the information requirement needed to exchange between the agent and the user. This information, compared with a prebuilt confusion matrix, is collected via controlled experiments for these agents that "summarizes how well an agent achieves the information requirements of a particular task for a set of dialogues instantiating a set of scenarios" [14].

PARADISE calculates the task cost by measuring two factors: firstly, task efficiency, which might be represented through determining number of utterances that takes to complete the task and the elapsed time that it needs; secondly, measuring the quality of the task, which might be determined, based upon the agent response delay and utterances' recognition errors rate (spoken utterances). PARADISE considers a small number of the total utterances needed to achieve the task better than a large number. This might be true with a CA that provides information for a train schedule between cities, for example. In contrast, this might be not true for other CAs that are designed to handle open conversations (the user converses in general about the selected domain's topic) between the CA and user. Therefore, a CA that considers the largest number of the total utterances might perform better, assuming that a larger number might mean that the user is more interested in using the CA.

Evaluating a CA is a divergent problem due to the number of metrics that can be used to evaluate it. For instance, a CA can be evaluated using usability metric [20, 21], user satisfaction metric [14, 22], response quality metric [23], ease of use metric [15], conversation duration metric [24], task completion level metric [25] and natural agent behaviour metric [15]. Each of these metrics has its own characteristics, objectives and its techniques and thus using all of these metrics might be not useful for evaluating a specific CA. According to [26], the best CA evaluation should be related to the nature of the CA’s task and the users’ needs. For instance, evaluating a ticket booking CA differs from evaluating a psychiatrist CA. The fundamental purpose behind a system’s evaluation is to improve its performance. The lack of a comprehensive evaluation framework has been a limiting factor in the growth of Conversational Agents [4]. In addition, different CAs might needs different approaches to evaluate [26].

a CA evaluation plays an important role for all participants building and using the CA [27]. It is important for CA developers “to tell if their system is improving”, and for CA’s integrators “to determine which approaches should be used where”, and for consumers also “to identify which system will best meet a specific set of needs” [27]. Therefore, a combination of objective measures and subjective measures will be better for evaluating a Conversation Agent.

II. THE SELECTED CASE STUDY ARABCHAT

In this paper, a specific text based CA called ArabChat [3] has been chosen to evaluate the proposed evaluation methodology. The ArabChat is a related research work for the paper’s authors so it easy to access and this is the reason why it has been chosen in this research. The ArabChat is an Arabic based CA which means it handle Arabic conversations. This is the reason why the proposed and developed evaluation framework in this paper called the “ArabChat Evaluator”. However, the developed evaluation methodology can work for any CA for all natural languages which means it is language independent.

The ArabChat uses the Pattern Matching technique to handle the Arabic textual conversations. The development of ArabChat needs to meet three requirements: scripting language, engine and brain. The scripting language will be used to script the specific domain aspects in order to represent them. While, the brain is a structured store or knowledge base that is used to store the domain’s scripts. The engine handles user’s utterances (conversations) that target the scripted domain.

The ArabChat is a rule-based Conversational Agent and it fundamentally is comprised of a novel scripting engine and a rule-based scripting language structured in a novel way to handle the topics (contexts) of conversations. Each context (main topic) has several rules (sub-topic) and each rule has several patterns (to be matched with user sentence) and responses. Each context has a default rule to be fired when no rules matched a user’s utterance.

ArabChat is a turn-based Conversational Agent, which means each one of the conversation’s parties (user and ArabChat) has its turn for conversation. Once the user enters his/her utterance, ArabChat processes this utterance and replies with a suitable response. The conversation remains on-going until one of the conversation’s parties terminates it.

The ArabChat was deployed and published in ASU (Applied Science University) in Jordan to work as an information point advisor for their users (registered students, unregistered students and employees).

A comprehensive evaluation methodology consisting of objective and subjective approaches has been used to evaluate the ArabChat. The subjective approach has been done through asking the ArabChat’s users about their opinions from different aspects by filling an online questionnaire [3]. Where the objective approach has been conducted through automatic evaluation techniques and manual analysing and consists of the “Glass box” and “Black box” approaches [3]. The “Glass box” approach evaluated ArabChat components individually. The ArabChat obtained a 67.836% of general user satisfaction [3]. This result can give a general overview of ArabChat performance, but it does not give a full indicator about its performance. Hence, the “Black box” approach using the proposed and developed methodology in this paper will be used to evaluate ArabChat and giving more accurate indication.

The ArabChat was evaluated depending on the ratio of matched and unmatched utterances [1]. This technique might give a general overview about the ArabChat’s performance. However, it cannot give an accurate result as the utterances might matching wrong rules. Therefore, in this paper a new framework will be modelled and developed to evaluate the ArabChat in a more accurate way. The next section is describing the proposed and developed framework.

III. THE “ARABCHAT EVALUATOR”

The “ArabChat Evaluator” is based on the black box evaluation approach which means testing and evaluating the ArabChat CA as one unit. The “ArabChat Evaluator” is based on a comparison process between the user’s utterance and ArabChat’s response in terms of the existing of “Information Requirements” (discussed later) words without dealing with any internal component details.

Generally, in a conversation between a user and a CA words need to be exchanged between them. Regardless of the CA type (spoken or textual), these words are found in the user’s utterance. In the ArabChat case, these words are in the text form. Not all of the utterance’s words are important to check but some of them are important (keywords) to check. For instance, the utterance “من فضلك, ما هو ايميل رئيس الجامعة؟” “Excuse me, what is the university president’s email?” contains 7 words. The only important words (keywords) are 3 words which are “ايميل” “email”, “رئيس” “president” and “الجامعة” “university” which they construct the topic “Email of university’s president”. These important words will be called in this evaluation “Information Requirements”.

The “ArabChat Evaluator” aims to evaluate the ArabChat‘s performance through the analysis of the quality of ArabChat’s response, which might indicate the user’s satisfaction. The quality of a response means how much a replied response is related to the processed utterance.

The “ArabChat Evaluator” is a separate system from ArabChat and works in an offline mode whenever it needs to evaluate ArabChat. Before proceeding with discussing the “ArabChat Evaluator” methodology, it is important to discuss the “Information Requirements” that “ArabChat Evaluator” is based on.

A. The “Information Requirements”

Each rule in ArabChat has its own topic to handle. An utterance that causes a rule to fire (the utterance matched one of the rule’s patterns) should contain some keywords related to the rule’s topic. For instance, a rule X is designed to reply to users asking about fees of a computing course in ASU. In order to fire the rule X, an utterance should contain at least two keywords, which are “سعر” “fee” and “حاسوب” “computing”, structured in a suitable way in the utterance. These two keywords (“سعر” and “حاسوب”) are considered as “Information Requirements” to fire rule X.

The “Information Requirements” is part of a rule structure and it contain a list of numbers with each number representing a list of different keywords. These keywords are stemmed and grouped semantically in separate groups, as presented in Table 1. This table represents a sample of the whole ArabChat “Information Requirements” list. The first group in Table 1 has 4 words, all of which might convey the same meaning (“fee”). Although, the fourth group has 3 words with 3 different meanings, all of them might be semantically related, and thus they are put in the same group. For the above mentioned rule (X), its “Information Requirements” parameter (according to Table 1) is (1,3) which represent the keywords (“سعر” “fee” and “حسب” “computing”). Implicitly, it is possible to consider “Information Requirements” as a list of keywords. ArabChat has list of “Information Requirements” that contain all ArabChat applied domain’s keywords. These keywords or these parameters does not involved in the user’s conversation handling process. As mentioned before, the ArabChat engine is based on the pattern matching technique to handle the conversations.

TABLE I. A SAMPLE OF ARABCHAT “INFORMATION REQUIREMENTS” LIST

#	Group words	Group words in English
1	سعر, ثمن, كلف, رسم	Price, cost, fee
2	مدد, جدد, اجل	Delay, postpone
3	حسب, برمج,كمبيوتر	Compute, program, computer
4	تاريخ, وقت, ساعة	Date, time, hour

B. The “ArabChat Evaluator” framework

The “ArabChat Evaluator” is based on the “Information Requirements” that are exchanged between a user and ArabChat in order to evaluate the ArabChat. The “ArabChat Evaluator” works in offline mode and in isolating from the ArabChat. This means that it only used when it needs to evaluate the ArabChat which means it does not affect the ArabChat’s performance when handling users’ conversations.

The mechanism used to determine the “Information Requirements” of an utterance differs from that used to determine the “Information Requirements” of a response. The mechanism used to determine the “Information Requirements” of an utterance is based on checking the utterance’s words, as it will be described later in this section, while the mechanism used to determine the “Information Requirements” of a response is based on the “Information Requirements” parameter of a rule that belongs to this response. Each response in ArabChat’s domain should belong to a specific rule. Each rule has an “Information Requirements” parameter. The “Information Requirements” parameter is not involved in the pattern matching process that the scripting engine adopted to match an utterance and then fire a rule. As discussed before, this parameter is just used for evaluation purposes, which means it used after all users finish their conversations with ArabChat. In contrast, during a user conversation, ArabChat accumulates the “Information Requirements” parameter contents related to the fired rules and stores them in ArabChat logs. Figure 1 shows the “ArabChat Evaluator” framework.

The “ArabChat Evaluator” reads the contents of the “Brief Log” (located in ArabChat brain) record by record in order to acquire its input (utterance and response) and produce an output (evaluation results). The “Brief Log” has 3 blank fields: “response evaluation”, “Patterns scripting evaluation”, and “conversation evaluation”. These blank fields will be filled by the “ArabChat Evaluator” for each utterance. Filling these fields means evaluating the processed record (utterance). Each record represents a conversation between a user and ArabChat. The “ArabChat Evaluator” starts its work by reading the first unevaluated record (its three fields are blank) in the “Brief Log”. Then, it moves to the next unevaluated record and so on. Table 2 shows a customised sample of the components of the “Brief Log” before the evaluation process begins.

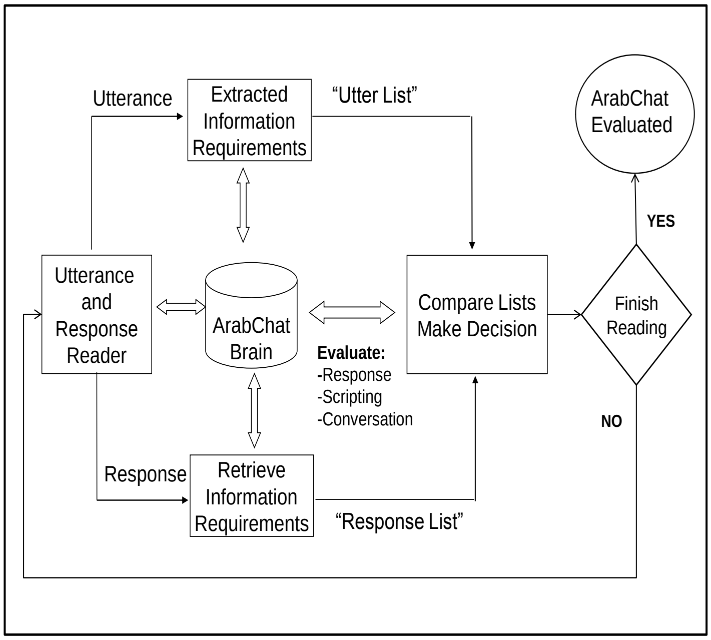

Fig. 1. The “ArabChat Evaluator” framework

TABLE II. A CUSTOMISED SAMPLE OF "BRIEF LOG" COMPONENTS BEFORE THE EVALUATION PROCESS

#	Utterance in Arabic	Utterance in English	Response in Arabic	Response in English	Information Requirement	Response evaluation	Patterns Scripting evaluation	Conversation evaluation
1	كم سعر ساعه تخصص الحاسوب	What is the price of a credit hour of computing course.	سعر الساعة المعتمدة لتخصص علم الحاسوب هو 75 دينار أردني	The price of a credit hour for computing course is 75 JD which is equaled to 106 USD	8,8,48			
2	كم عدد الساعات لتخصص التمريض والمحاسبة	What is the price of a credit hour of nursing accounting.	عدد الساعات المعتمده لتخصص علم الحاسوب هو 132 ساعة معتمدة	Total number of credit hours of Computer Science is a 132 hours.	29,29			
3	لقد سمعت من أصدقائي بأنك ثرثار	I have heard from my friends that you are talkative	تقدم جامعة العلوم التطبيقية العديد من التخصصات في شتى المجالات وقد صمم هذا النظام ليقوم بالتسهيل عليك عملية الاستفسار والدردشة مع موظفي الجامعه فالرجاء أن تخاطبني باللغة العربية الفصحى.	ASU provides many of specialists in different fields and this system is designed to facilitate your communications with the university, so please speak with me using modern Arabic.				
4	متى يبدأ التسجيل في الجامعة	When is the registration will start in the university	الحد الأدنى لمعدل القبول والتسجيل في التخصصات العلمية هو 80% في الثانوية العامه من الفرع العلمي	The minimum rate of acceptance and registration in scientific disciplines is 80% in the high school section of scientific	11,33			

According to Figure 1, the "ArabChat Evaluator" performs the following steps for each unevaluated conversation in the "Brief Log":

1) Read the utterance and the response using the "Utterance and Response Reader" module.

2) Extracts the "Information Requirements" from the utterance using "Extracted Information Requirement" module

3) Retrieves the "Information Requirements" for the response using the "Retrieve Information Requirements" module

4) Undertakes a comparison between the two generated "Information Requirements" lists and then take its decision (evaluating a conversation) using the "Compare Lists Make Decision" module

After evaluating the first unevaluated conversation in the "Brief Log", the "ArabChat Evaluator" checks the entries of the "Brief Log". If unevaluated records still exist, it starts reading and repeats the previous steps until it is finish checking all the log's records and then it evaluates the ArabChat. The following components for the "ArabChat Evaluator" framework will now be discussed:

- **The "Utterance and Response Reader" module:** The "ArabChat Evaluator" starts reading the utterance and the response using the "Utterance and Response Reader" module from the "Brief Log" located in ArabChat's brain. Then, in order to retrieve its "Information Requirements", the "ArabChat Evaluator" sends the utterance and the response to the "Extracted Information Requirement" and the "Retrieve Information Requirements" module respectively.
- **The "Extracted Information Requirement" module:** extracts the "Information Requirements" from the utterance by tokenising the utterance and converting it into a list of words. Then a stemming process is done on the list of words that converts it to a stemmed list of words, called the "Utter List". The stemming process is based on an Arabic based stemming algorithm with a good performance and it explained in [28]. Then, the "Utter List" elements are matched with ArabChat "Information Requirements" list elements. If a matching occurs, the group number of the matched word in the ArabChat "Information Requirements" list is replaced by the matched word in "Utter List". Otherwise, the unmatched word in the "Utter List" is removed. Finally, the "Utter List" contains only numbers that represent the groups matched to the utterance's stemmed words. Eventually, "ArabChat Evaluator" removes the duplication of the same number from the "Utter List", if they exist.

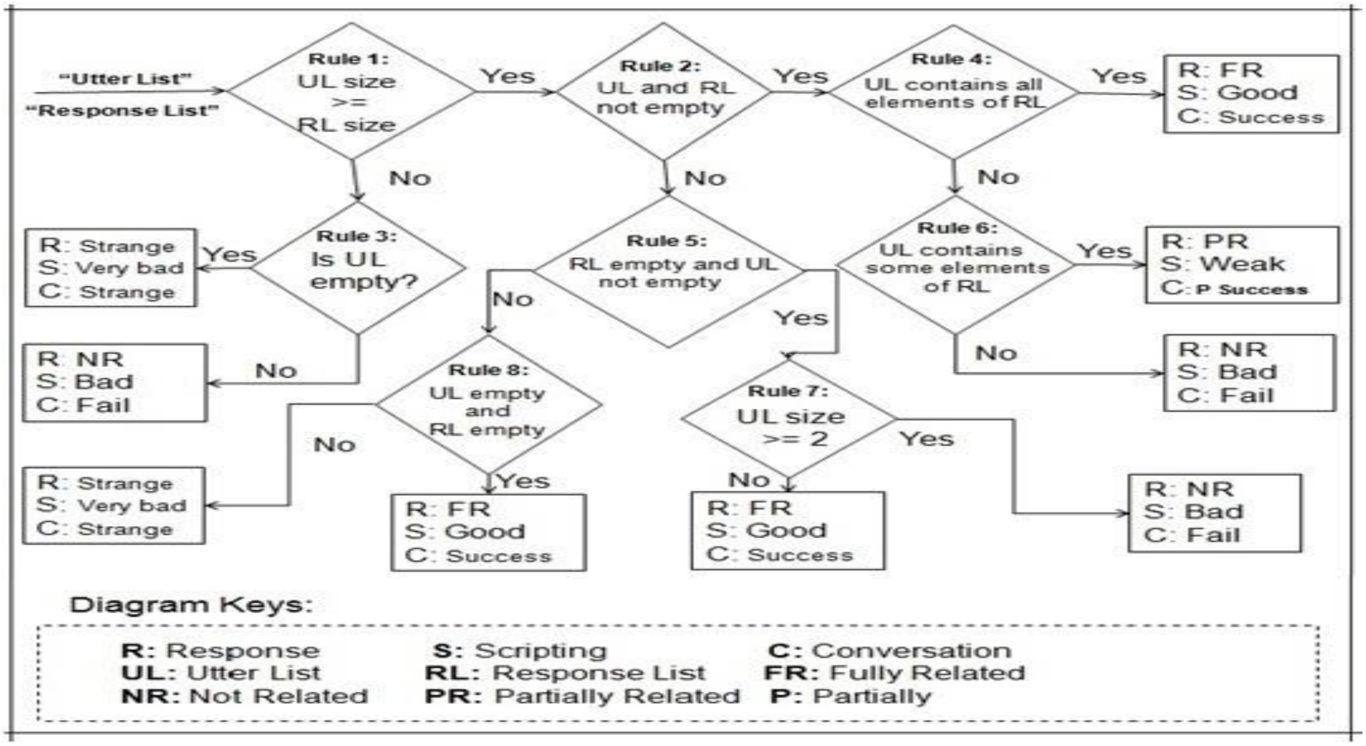

Fig. 2. The "Compare Lists Make Decision" module methodology

- **The "Retrieve Information Requirements" module:** retrieves the "Information Requirements" for the rule that the processed response belongs to. However, the "Information Requirements" is already stored as a collection of numbers in the "Brief Log" that was accumulated by ArabChat during users' conversations. Then, it starts removing the duplicated numbers that might be caused by the accumulative process and calls it the "Response List".

- **"Compare Lists Make Decision":** both of the generated lists ("Utter List" and "Response List") are sent to the "Lists Comparing Make Decision" module in order to start the comparison and evaluation of the processed conversation. Figure 2 represents the "Compare Lists Make Decision" module methodology.

The "ArabChat Evaluator" evaluates three aspects of each conversation, "Response", "Scripting", and "Conversation". For each aspect, different results might be generated as presented in Table 3. In this table, each evaluation aspect has 4 potential results. For instance, the "Response" aspect has 4 results which are either "Strange", "Not related", "Partially related" or "Fully related".

The "Response" evaluation aspect's result will determine the results of the rest aspects (the "Scripting" and "Conversation"). For instance, if the "Response" result is "Not related", then the "Scripting" results will be "Bad" and the "Conversation" results "Fail". More detailed explanations for these evaluation aspects and the differences between their results will be described later in this section when the "Compare Lists Make Decision" module methodology is described.

TABLE III. EVALUATION ASPECTS RESULTS

Evaluation aspect	**Evaluation result**
Response	Strange
	Not related
	Partially related
	Fully related
Scripting	Very bad
	Bad
	Weak
	Good
Conversation	Strange
	Fail
	Partially success
	Success

The following rules that presented in Figure 2 explain the methodology of the "Compare Lists Make Decision" module:

Rule 1: Check the "Utter List" and "Response List" sizes. If "Utter List" size >= "Response List" size, go to Rule 2; otherwise go to Rule 3.

Rule 1 description: rule 1 is the first rule that deals with the two lists and is responsible for checking their sizes. When the "Utter List" is not empty, it means the user entered at least one keyword that already matched one of the applied domain's keywords. When the "Response List" is not empty, it means that the ArabChat fired a rule after a matching occurred between the user's utterance and a pattern. Default rules have no "Information Requirements" values; i.e. if a default rule is fired, the "Response List" will be empty.

When the size of the "Utter List" is greater than or equal to the size of the "Response List", it concludes on one of two meanings: firstly, both lists have elements and that the "Utter List" elements are greater than the "Response List" elements, which means that the user has entered a greater amount of keywords than required to fire the processed a rule (the fired rule that has the processed response). Secondly, both lists are empty (their sizes are zeros) and consequently are equalled.

Rule 2: Check the "Utter List" and "Response List" contents. If none of them are empty, continue to Rule 4; otherwise continue to Rule 5.

Rule 2 description: Rule 2 comes after Rule 1 in two conditions; either both lists are empty or the "Utter List" size is greater than the "Response List" size. Therefore, Rule 2 is responsible for limiting these probabilities by checking if the two lists are not empty, if the two lists are not empty, a user entered at least one keyword and ArabChat fired a rule other than the default rule.

Rule 3: Check the "Utter List" contents. If empty, then the evaluations for the three aspects are:

Response: Strange.

Patterns Scripting: Very bad.

Conversation: Strange.

If the "Utter List" is not empty, then the evaluations for the three aspects are:

Response: Not related.

Patterns Scripting: Bad

Conversation: Fail.

Rule 3 description: Rule 3 comes after Rule 1 confirms that "Response List" size is greater than "Utter List" size. However, Rule 3 has two probabilities; either the "Utter List" size is smaller than "Response List" size or the "Utter List" is empty. Therefore, Rule 3 checks the "Utter List" size, whether it is empty or not. If empty, it means that a user has not entered any keyword related to the applied domain. Therefore, the "ArabChat Evaluator" will evaluate the conversation as "Strange", because if the user entered no keywords, the ArabChat should fire a default rule and then the "Response List" should be empty. As a result, the response evaluated was "Strange", thus indicating that the patterns scripting for the targeted rule is "Very bad". In contrast, if the "Utter List" is not empty, it means that a user has entered at least one keyword and the "Information Requirements" of the fired rule, whether default or not, is less than what the user entered. Therefore, the "ArabChat Evaluator" decided to evaluate the conversation as "Fail" because the response is "Not related" and thus the patterns scripting is "Bad".

Rule 4: Check the "Utter List" and "Response List" contents. If all "Response List" elements are in "Utter List" list, then the evaluations for the three aspects are:

Response: Fully related

Patterns Scripting: Good

Conversation: Success.

Otherwise (not all "Response List" elements are in "Utter List" list), continue to Rule 6.

Rule 4 description: Rule 4 comes after Rule 2 if it is agreed that the two lists are not empty. Therefore, the "ArabChat Evaluator" tests through Rule 4 if all elements of the "Response List" are in the "Utter List" list. If so, the ArabChat fires a rule that meets all the utterance requirements of the user, and thus, the response is evaluated as "Fully related" and the conversation is evaluated as "Success". Consequently, the pattern scripting for the fired rule is "Good". In contrast, if not all elements of the "Response List" are in the "Utter List", the "ArabChat Evaluator" will continue to Rule 6, which is responsible for testing if some elements of "Response List" are in the "Utter List".

Rule 5: Check the "Utter List" and "Response List" contents. If the "Utter List" is not empty and "Response List" is empty, continue to rule 7; otherwise continue to Rule 8.

Rule 5 description: Rule 5 comes after Rule 2 on the condition that at least one of the two lists is empty. Therefore, Rules 5 is used to test if the "Utter List" is not empty and the "Response List" is empty. If so, this means that the ArabChat fired a default rule but a user had entered at least one keyword. Consequently, the "ArabChat Evaluator" continues to Rule 7 to check if the "Utter List" size is greater than or equal to 2 (Threshold), which means the user entered enough "Information Requirements" in his/her utterance. Otherwise, "ArabChat Evaluator" will continue to Rule 8 which is responsible to test if both lists are empty. Experimentally, it was determined that the minimum number of keywords that should be in a matched utterance is 2, which is considered a threshold point.

Rule 6: Check the "Utter List" and "Response List" contents. If some "Response List" elements are in the "Utter List", then the evaluations for the three aspects are:

Response: Partially related.

Patterns Scripting: Weak.

Conversation: Partial success.

Otherwise (no element of "Response List" is in "Utter List" list**),** then the evaluations for the three aspects are:

Response: Not related.

Patterns Scripting: Bad.

Conversation: Fail.

Rule 6 description: Rule 6 comes after Rule 4 on the condition that not all "Response List" elements are in the "Utter List". Therefore, Rule 6 is used if some (one or more but not all) of the "Response List" elements are in the "Utter List". If so, a user has entered keywords in his/her utterance and ArabChat fired a rule that met some of the user's utterance requirements. Thus, the "ArabChat Evaluator" evaluates the response as "Partially related" and the conversation as "Partial success" because the fired rule replied to some of the user's requirements but not all of them. Consequently, this indicates that the patterns scripting for the fired rule is "Weak". Otherwise, the user entered keywords and ArabChat fired a rule not related to the user's utterance requirement at all. Therefore, the "ArabChat Evaluator" evaluates the response as "Not related" and the conversation as "Fail", as the scripting of the fired rule pattern is "Bad".

Rule 7: Check the "Utter List" size. If "Utter List" size >= 2, then the evaluations for the three aspects are:

Response: Not related

Patterns Scripting: Bad

Conversation: Fail.

Otherwise ("Utter List" size < 2**),** then the evaluations for the three aspects are:

Response: Fully related.

Patterns Scripting: Good.

Conversation: Success.

Rule 7 description: Rule 7 comes after Rule 5 on the condition that the "Utter List" is not empty while the "Response List" is empty. Therefore, Rule 7 tests if the "Utter List" size is greater than, or equal to 2 (the threshold point). If so, this means a user entered a minimum of 2 keywords, and ArabChat fired a default rule because the "Response List" was empty. The "ArabChat Evaluator" evaluates the response as "Not related" and the conversation as "Fail", because the scripting of the fired rule patterns is "Bad". Otherwise, the user enters less than 2 keywords, which is below the determined threshold point. Therefore, the "ArabChat Evaluator" evaluates the response as "Fully related" and the conversation as "Success", thus the pattern scripting is "good" as long as the utterance is outside the scripted domain. Entering an amount of keywords less than the threshold with an empty "Response List" means that a user entered an utterance outside the applied domain and ArabChat is only responsible to reply to utterances inside the applied domain, thus the conversation is considered successful.

Rule 8: Check the "Utter List" and "Response List" contents. If both lists are empty, then the evaluations for the three aspects are:

Response: Fully related.

Patterns Scripting: Good.

Conversation: Success.

Otherwise**,** then the evaluations for the three aspects are:

Response: Strange.

Patterns Scripting: Very bad.

Conversation: Strange.

Rule 8 description: Rule 8 comes after Rule 5 on the condition that the "Utter List" is not empty while the "Response List" is empty. Therefore, Rule 8 is used if both lists are empty. If so, this means that a user entered an utterance outside the applied domain, and ArabChat fired a default rule. Thus, the "ArabChat Evaluator" evaluates the response as "Fully related" and the conversation as "Success" and patterns scripting as "Good". Otherwise, the "Response List" was not empty, while the "Utter List" was empty. In other words, a user entered no keywords and the ArabChat fired a rule that requires keywords in the processed utterance to be fired. However, it might be impossible for this case to happen. If it does, there is something strange in patterns scripting or in the rule "Information Requirements" list. Therefore, "ArabChat Evaluator" evaluates the response and the conversation as "Strange" and thus, the patterns scripting result is "Very bad".

As the "ArabChat Evaluator" is mainly used to evaluate the quality of response generated by the ArabChat scripting engine, it may be a good indicator of the whole ArabChat performance. When a response is evaluated as "Fully related", it means that the whole utterance's "Information Requirements" are replied. Consequently, this means that the scripting engine fires the best rule, indicating that the scripting engine methodology worked properly. On the other hand, firing the best rule indicates that the rule scripting in general, and especially its patterns scripting, are scripted well. Therefore, the scripting is evaluated as "good", and the conversation is evaluated as "Success". However, the "Fully related" response might be caused by the user entering an utterance outside the applied domain. ArabChat is only responsible to reply utterances related to the applied domain. Regardless of the reason of evaluating a response as "Fully related", it is obvious that not only the response is evaluated. Implicitly, ArabChat's scripting engine and the whole ArabChat is evaluated by this process.

By applying the "ArabChat Evaluator" to the utterances presented in Table 2, the evaluation results are achieved and presented in Table 4. The "ArabChat Evaluator" can be run at any time in order to evaluate the unevaluated utterances in the "Brief Log". Each time the "ArabChat Evaluator" runs, it checks that the evaluation fields (response, pattern scripting, and conversation) of the utterances are empty and starts evaluating them.

As discussed earlier, the "ArabChat evaluator" aims to evaluate the ArabChat in terms of its generated quality of response. Thus, it acts as an indicator of ArabChat's scripting engine performance and reflects the quality of pattern scripting. Subsequently, user satisfaction with ArabChat can be determined using the same methodology (ArabChat evaluator).

TABLE IV. A CUSTOMISED SAMPLE OF THE “BRIEF LOG” COMPONENTS AFTER THE EVALUATION PROCESS

#	Utterance	Utterance in English	Response in Arabic	Response in English	Information Requirement	Response evaluation	Scripting evaluation	Conversation evaluation
1	كم سعر ساعه تخصص الحاسوب	What is the price of a credit hour of computing course.	سعر الساعة المعتمدة لتخصص علم الحاسوب هو 75 دينار أردني	The price of a credit hour for computing course is 75 JD which is equaled to 106 USD	8,8,48	Fully related	Good	Success
2	كم عدد الساعات لتخصص التمريض والمحاسبة	What is the price of a credit hour of nursing accounting.	عدد الساعات المعتمده لتخصص علم الحاسوب هو 132 ساعة معتمدة	Total number of credit hours of Computer Science is a 132 hours.	29,29	Partially related	Weak	Partially success
3	لقد سمعت من أصدقائي بأنك ثرثار	I have heard from my friends that you are talkative	تقدم جامعة العلوم التطبيقية العديد من التخصصات في شتى المجالات وقد صمم هذا النظام ليقوم بالتسهيل عليك عملية الاستفسار والدردشة مع موظفي الجامعه فالرجاء أن تخاطبني باللغة العربية الفصحى.	ASU provides many of specialists in different fields and this system is designed to facilitate your communications with the university, so please speak with me using modern Arabic.		Fully related	Good	Success
4	متى يبدأ التسجيل في الجامعة	When is the registration will start in the university	الحد الأدنى لمعدل القبول والتسجيل في التخصصات العلمية هو 80% في الثانوية العامه من الفرع العلمي	The minimum rate of acceptance and registration in scientific disciplines is 80% in the high school section of scientific	11,33	Not related	Bad	Fail

The “ArabChat evaluator” was run in order to read the reported records from the “Brief Log” and fill the empty fields, which are “Response Evaluation”, “Scripting Evaluation”, and “Conversation Evaluation”. Each record represents one conversation (utterance) between a user and ArabChat. The “ArabChat evaluator” analyses these conversations one by one as described above. Users’ conversations are categorised by the user sequence numbers. Therefore, it is possible to determine the satisfaction per user and the average rate of general user satisfaction. The user satisfaction is determined based on the evaluated parameter (Conversation Evaluation), which indicates the status of a user conversation with ArabChat. However, determining the average user satisfaction through the descriptive evaluated results (“Success”,”Partially success”, “Fail” and “Strange”) is not an easy task. Therefore, a new technique was developed, called the “ArabChat Evaluation Calculator”, which converts these descriptive evaluated results into numeric values and calculates the user satisfaction value.

The “ArabChat Evaluation Calculator” is based on the number of utterances per user in order to assign a numeric value for the evaluated results (“Success”, “Partially Success”, “Fail” and “Strange”). In this technique (the “ArabChat Evaluation Calculator”), the “Strange” conversation will be considered as “Fail” conversation. Equations 1, 2 and 3 are used to assign the equivalent numeric values for the evaluated results:

$$\mathbf{NVSC} = \frac{\mathbf{100}}{\mathbf{Num.Utters}} \tag{1}$$

$$\mathbf{NVPSC} = \frac{\mathbf{NVSC}}{\mathbf{2}} \tag{2}$$

$$\mathbf{NVFC} = \mathbf{0} \tag{3}$$

Where,

NVSC: Numeric Value Success Conversation.

NVPSC: Numeric Value Partially Success Conversation.

NVFC: Numeric Value Fail Conversation.

Num. Utters: Number of utterances.

Equation 1 represents the converting mechanism for the evaluated result “Success” into its equivalent numeric value by assuming that the user satisfaction is 100%. Therefore, a division process has been performed, as Equation 1 showed, by taking into consideration the total number of utterances for the evaluated user. For instance, if a user entered 5 utterances, the “ArabChat Evaluation Calculator” calculates the NVSC as 100/5 and thus NVSC=20.

Then, the “ArabChat Evaluation Calculator” applies Equation 2 in order to calculate the NVPSC according to the equation, as 20/2 and thus, NVPSC=10. Finally, the NVFC is calculated according to Equation 3, which always assigns a zero value to the NVFC for “Fail” and “Strange” conversations.

Naturally, user satisfaction might improve or contract during prolonged use of ArabChat. Therefore, determining the user satisfaction by calculating the average of the numeric evaluated results might not be quite accurate. Instead, an exponential average calculation method has been applied on the numeric results in order to calculate a more natural result. The exponential average calculation method is based on the consecutive conversations of the evaluated results for the same result type, such as consecutive “success”, “partially success” or “fail conversations”. The following examples show how the “ArabChat Evaluation Calculator” based on the exponential average calculation assigns the equivalent numeric values for the consecutively evaluated results:

The First success conversation = NVSC.

The Second consecutive success conversation = NVSC $^{1.01.}$

The Third consecutive success conversation = NVSC $^{1.02}$

The Fourth consecutive success conversation = NVSC $^{1.03}$

The Eleventh consecutive success conversation = NVSC $^{1.10}$

And so on.

The First partially success conversation = NVPSC.

The Second consecutive partially success conversation = NVPSC $^{1.01.}$

The Third consecutive partially success conversation = NVPSC $^{1.02}$

The Fourth consecutive partially success conversation = NVPSC $^{1.03}$

The Eleventh consecutive partially success conversation = NVPSC $^{1.10}$

And so on.

The First fail conversation = NVFC.

The Second consecutive fail conversation = NVFC$^{.}$

The Third consecutive fail conversation = NVFC.

The Fourth consecutive fail conversation = NVFC.

The Eleventh consecutive fail conversation = NVFC

And so on.

The next section discusses the evaluation results of the “Black box approach” after applying the “The ArabChat Evaluator” on the conversations of ArabChat experiment’s users.

IV. The Evaluation

The evaluation for the proposed and developed methodology has been done through conducting an real experiment on the selected CA(ArabChat). The experiment was conducted to test the full ArabChat capabilities from different aspects through applying the developed methodology (ArabChat Evaluator). As mentioned before, this proposed and developed framework will be able to test the ArabChat engine in terms of its ability to match utterances properly and it can test the user satisfaction in general.

ArabChat was deployed on the ASU website [29] and accessed by all qualified users such as registered students, non registered students, and employees. ArabChat was available online and in use for 23 days.

The ArabChat handled 1766 utterances from 203 users, an average of 8.699 utterances per user.

Evaluation results and discussion

After applying the “ArabChat Evaluator” and the “ArabChat Evaluation Calculator” on the ArabChat users’ conversations in the experiment (203 users with 1766 utterances), the average of ArabChat users’ satisfaction is = 64.31%.

A manual checking for all users’ conversations has been done in order to classify them as serious users and unserious users. The serious user who keep talking inside the selected applied domain (information point advisor for ASU). Where the unserious user who just try to trick the ArabChat, saying something funny or his/her utterances has impolite words. This manual checking has been raised that an 8.267% of users’ conversations were placed in the second category which might reveal the existence of unserious users who negatively affected the evaluation result (users’ satisfaction is 64.31%). As a result, this outcome (64.31%) could be considered as a reasonable result of the average of user satisfaction.

V. Summary

The fundamental purpose behind a system’s evaluation is to improve its performance. As discussed in this paper, the lack of a comprehensive evaluation framework has been a limiting factor in the growth of Conversational Agents. In addition, different types of CAs might require different frameworks of evaluation.

Furthermore, devising an automatic method for evaluating CAs is not an easy task as a user utterance might have a rich semantic meaning, which is hard to automatically detect. In addition, CA conversations vary among users even for closed applied domains.

Generally, chatting with a CA does not mean that a user will keep entering either questions or non-questions only. The natural conversations between a user and a CA should consist of both (questions and non-questions). Nevertheless, the amount of question and non-question utterances might be based on the following factors:

1) The topical nature of a CA's applied domain; for instance, an entertainment domain might differ from an information point advisor.

2) The users, if they are familiar with the nature of a CA. It can be concluded from experiment 1 of ArabChat that many users consider ArabChat a question answering system. As a result, a large amount of questions were entered. In addition, 92.3% of experiment 1's users confirmed that they had never used any similar service before, which points to a lack of experience in handling these services.

3) The way a CA forms its response might also encourage a user to ask questions or continue chatting with non-question utterances.

The ArabChat was evaluated depending on the ratio of matched and unmatched utterances [1]. This technique might give a general overview about the ArabChat's performance. However, it cannot give an accurate result as the utterances might matching wrong rules. Therefore, in this paper a new framework has been modelled and developed to evaluate the ArabChat in a more accurate way. By the proposed framework, the evaluation focused on the "Information Requirements" that should be shared between the utterance and the fired rule. According to the conducted experiment and the evaluation of the ArabChat based on the proposed framework (ArabChat Evaluator and ArabChat Calculator) and based on the experiment's results, it can be concluded that ArabChat successfully handled conversations for ASU students.

ACKNOWLEDGMENT

The authors are grateful to the Applied Science Private University, Amman, Jordan, for the full financial support granted to this research.

REFERENCES

[1] Hijjawi, M., et al. ArabChat: An Arabic Conversational Agent. in proceeding of the 6th International Conference on Computer Science and Information Technology (CSIT). 2014. Amman, Jordan: IEEE Explore.

[2] Turing, A., Computing machinery and intelligence. Mind, 1950: p. 433-460.

[3] Hijjawi, M., ArabChat: An Arabic Conversational Agent. PhD thesis, in School of Computing. 2011, Manchester Metropolitan University: Manchester. p. 241.

[4] Goh, O., A framework and evaluation of Conversational Agents. PhD thesis, in Information Technology school. 2008, Murdoch University.

[5] Russell, s. and P. Norvig, Artificial Intelligence A Modern Approach. . Vol. Third edition. . 2010: Pearson Education.

[6] Sammut, C. and D. Michie, InfochatTM Scripter's Manual, Convagent Ltd. . 2001: Manchester.

[7] Weizenbaum, J., ELIZA-A computer program for the study of natural language communication between man and machine. Communications of the ACM., 1966. Vol 10.: p. PP 36-45.

[8] Wallace, R. ALICE: Artificial Intelligence Foundation Inc. . 2008 [cited; Available from: http://www.alicebot.org.

[9] Crystal, D., Dictionary of linguistics and phonetics., Blackwell., Editor. 2008.

[10] Habash, N., Introduction to Arabic Natural Language Processing, ed. U.o.T. Graeme Hirst. 2010: Morgan & Claypool.

[11] O'Shea, K., Z. Bandar, and K. Crockett, A Novel Approach for Constructing Conversational Agents using Sentence Similarity Measures. 2008.

[12] Almarsoomi, F., et al., Arabic Word Semantic Similarity. International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering. World Academy of Science, Engineering and Technology, 2012. Vol:6, No:10.

[13] David, D., S. Karen, and J. rck, Natural language processing for information retrieval. . Commun. ACM., 1996. Volume 39: p. pp. 92-101.

[14] Walker, M., et al. PARADISE: A framework for evaluating spoken dialogue agents. In Proc. of 35th ACL. 1997.

[15] Sanders, G. and J. Scholtz, Measurement and evaluation of embodied conversational agents. MIT Press, 2000.

[16] Hugh, L. Loebner's Home Page. Available at: www.loebner.net. 2016 [cited Accessed in 2016.

[17] Maier, E., M. Mast, and S. LuperFoy, Dialogue Processing in Spoken Language Systems., in ECAI'96 Workshop. 1996: Hungary.

[18] Hirschman, L., The Roles of language processing in a spoken language interface. National Academy Press Washinton- Voice Communication Between Humans and Machines, 1995: p. pp217-37.

[19] Simpson, A. and F. Norman. Black box and glass box evaluation of the SUNDIAL system. in Proceedings of the 3rd European Conference on Speech Communication and Technology. 1993.

[20] ANDERSEN, V., et al., A methodological approach for designing and evaluating intelligent applications for digital collections. . Applied Artificial Intelligence,, 2003. Vol 17(Issue 8-9).

[21] Lamel, L., Bennacef, S., Gauvain, J. L., Dartigues, H. and Temem, J. N. , User evaluation of the MASK kiosk. Speech Commun., 2002. Vol 38(1): p. PP 131-39.

[22] Xiang, Y. and C. Yam, Design and evaluation of Elva: an embodied tour guide in an interactive virtual art gallery. Research Articles. Comput. Animat. Virtual Worlds., 2005. Vol 16(2): p. pp. 109-19.

[23] Goh, O., et al., A Black-box Approach for Response Quality Evaluation of Conversational Agent Systems. . World Academy of Science Engineering and Technology., 2007. Vol 3: p. PP 195-203.

[24] Kopp, S., et al., A conversational agent as museum guide: design and evaluation of a real-world application. Lecture Notes in Computer Science. ed., Springer-Verlag, 2005.

[25] Le Bigot, L., E. Jamet, and J. Rouet, Searching information with a natural language dialogue system: a comparison of spoken vs. written modalities. Applied Ergonomics, 2004. Vol 35(6): p. pp. 557-64.

[26] Abu Shawar, B. and E. Atwell. Different measurements metrics to evaluate a chatbot system. . in Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies, Rochester, New York: Association for Computational Linguistics. 2007.

[27] Lynette, H. and T. Henry, Overview of evaluation in speech and natural language processing. IN Survey of the state of the art in human language technology. ed., Cambridge University Press, 1997.

[28] Hijjawi, M., et al. An Application of Pattern Matching Stemmer in Arabic Dialogue System. . in 5th International KES Conference on Agents and Multi-agent Systems – Technologies and Applications. 2011. Manchester, UK: Elsiver.

[29] ASU. Applied Science University- ArabChat. 2011 [cited 2011; Available from: Available at: WWW.ASU.EDU.JO.