A Survey of Intelligent Language Tutoring Systems

Mustafa Al Emran
The British University in Dubai
Al-Buraimi University College
Al-Buraimi, Oman
malemran@buc.edu.om

Khaled Shaalan
The British University in Dubai
Dubai, UAE
khaled.shaalan@buid.ac.ae

***Abstract*— Intelligent Languages Tutoring Systems (ILTSs) plays a significant role in evaluating students' answers through interaction with them. ILTSs implements Natural Language processing (NLP) techniques in order to allow free input of words and sentences. ILTSs have the capability of identifying the input errors and provide an immediate feedback along with the errors source. It has been observed that ILTSs were not surveyed intensively; the reason that motivates us to conduct this research. Some NLP recent trends such as Latent Sematic Analysis and entailment were demonstrated. Different ILTSs have been discussed with a dedicated section about the development of Arabic ILTSs. Arabic share many of its characteristics with Semitic and morphologically rich languages. In our presentation we point out new trends that have been emerged while conducting survey.**

Keywords—Intelligent Language Tutoring systems; ILTS; ITS; Natural Language Processing, Arabic Langauge.

I. INTRODUCTION

Nowadays, computers are essential part in our lives. Computer applications have become widely used in different domains to provide useful facilities for their users. Examples of educational applications which find the functionalities of computers fit into their purposes are lesson tutoring, language testing, errors diagnosing and handling, and documents archival. These are good reasons that enable researchers and developers to think about designing an Intelligent Tutoring Systems (ITSs). These systems attempt to mimic the one-to-one human tutor in order to deliver learning material in the cyber space rather than using books and traditional learning environment. Natural Language Processing (NLP), as one of the Artificial Intelligence fields, plays a significant role in how computers could interpret and process human natural languages (text or speech) in order to develop useful learning applications.

In this review, we will shed the light on the relevant areas of Intelligent Language Tutoring Systems (ILTSs). ILTSs attempt to develop NLP techniques that focus on processing and assessing the learner's free text input. The use of ILTSs in education makes it easy to diagnose the leaner's error and provide convenient feedback along with the source of error (which is main goal of ILTS). We describe different ILTSs in this survey, including auto-tutor ([21], [22], [35]), web-based German Tutor [23], Beetle II [28], and CIRCSIM-Tutor ([29], [30]). It should be noted that Authoring Systems are different from ILITS as they just enable the author to represent his/her knowledge in natural language without resorting to any intelligent programming. As learning Arabic language is a challenge task, we discuss some characteristics and issues related to Arabic NLP which have led to the progress in Arabic ILTSs research. To the best of our knowledge, ILTSs have not yet been surveyed extensively, which has motivated us to conduct this survey.

The rest of this paper is structured as follows: Section II provides background about: Intelligent Tutoring Systems, Natural Language Processing with focus on two main issues (*Latent Semantic Analysis* and *Entailment issues*) as well as the characteristics that impact learning of the Arabic Language. Section III is divided into two sub-sections: the first one explores the development of ILTSs, while the second one mainly focuses on the progress made in Arabic ILTSs. Section IV gives some concluding remarks. Section V gives direction for future research.

II. BACKGROUND

A. Intelligent Tutoring Systems

Intelligent Tutoring Systems (ITSs) simulates the one-to-one human tutor for delivering knowledge interactively instead of using books and the traditional learning environment. To come up with the most learning outcomes, ITSs have incorporated several techniques such as: error identification and correction, and building consistent explanations [1] through integrating techniques of cognitive science and Artificial Intelligence [2], [3], [4]. There is a significant distinction between Computer-Assisted Instruction (CAI) and ITSs where the latter has the capability of interactive problem-solving and curriculum sequencing [5]. Different tutoring systems have been implemented to cover different languages such as: English, Arabic, Chinese, German and many others [34].

978-1-4799-3080-7/14/$31.00 ©2014 IEEE

B. Natural Language Processing

Natural Language Processing (NLP) is a domain of research that focuses on how computers interpret and process the human natural language (text or speech) in order to perform helpful applications [6]. NLP is based on Computer Science, linguistics and can also refer to computational linguistics. NLP has been claimed to include Natural Language Understanding (NLU) and Natural Language Generation (NLG) [7], [37], [40]. There are different NLP tasks that have been explored by many researchers. These tasks could include but are not limited to: part of speech tagger, tokenization, segmentation, stemming and parsing. They might be standalone tasks or integrated preprocessing tasks to larger tasks or development environments [19].

NLP applications can serve different purposes such as: Machine Translation, Speech Recognition, Artificial Intelligence, Expert Systems, Tutoring Systems (which is the main concern in this context), among others [6], [7]. Collectively, in order to process and analyze the available large amounts of text, NLP capabilities and expertise are required.

1. Latent Sematic Analysis

Latent Semantic Analysis (LSA) is an NLP technique that measures the similarity between any two fragments of text (a word, sentence, paragraph or document) [35] and also a technique for deriving relationships of anticipated contextual usage of words in a dialogue discussion [9]. LSA has been used in several applications such as: Auto-Tutor in order to evaluate learners' responses during the dialogue between the learner and the Auto-Tutor [10], [11]. LSA is able to differentiate various students' levels (good, vague and erroneous) [10].

Lexical (i.e. what has been written exactly) and *semantical* (i.e. what was intended) levels are one of the main challenges in NLP systems. For instance, some words have multiple uses in multiple contexts, while multiple words could have similar meaning; where LSA could handle these issues [8]. On the contrary, LSA has some limitations, such as not being able to recognize negation, encode word order, nor exactly determine what is being wrong in the incorrect response during the dialogue between the learner and the Auto-Tutor [11], [12].

2. Entailment Issues

Entailment can refer to the task of deciding whether the meaning of two pieces of text is entailed to each other (i.e. one refers to the other). Entailment has been involved in various applications such as: Information Retrieval, Machine Translation and Tutoring Systems. Evaluating learners' answers in ILTS is one of the main problems [11], [36]. When the Auto-Tutor poses a question to the student, there are some fixed expectations (i.e. correct answers) which correspond to that question. Entailment works on deciding whether the student's input text (i.e. the student's answer) will match those expectations. Let's consider the following example:

Expectation (E): Mustafa is interested in Mobile learning and Knowledge Management domains.

Answer (A): Mustafa is interested in both domains.

The challenging issue here is to decide whether (*E*) entails or logically indicate (*A*). There is a wide spectrum of techniques that could deal with entailment issues. Table 1 illustrates some of them.

TABLE I. ENTAILMENT ISSUES TECHNIQUES.

Weighted bag of words approach is proposed where an entailment relation among large fragments of text where the text size is a sentence or part of the sentence as to compare with the inference systems that are restricted to yes/no decisions based on first order logic [13].	*First order logic* along with *unification* approach is used in which the question is associated with an answer from a list of ideal answers (via few keywords and some basic semantics techniques). The highly the unification value, the highly the value at the top [14].	*Syntactic graph distance* technique has been used to handle textual entailment but it lacks to resolve/handle negations matters [15].

C. Aspects of Arabic Language

Arabic Language can be considered as both *important* (due to the Arabic culture, history and heritage) and *challenging* (due to its morphological and syntactic complexity) [17], [36]. Arabic is a *Semitic* language and is spoken by 330 million people [16]. Diacritization plays important role in properly pronouncing and disambiguating Arabic words [18] [38]. Arabic Language in use has been categorized into three types: *Classical Arabic* or *Quranic Arabic* (the formal version that had been used for 1500 years as the language of Islam), *Modern Standard Arabic* (the language used by media and education) and *Colloquial Arabic* (the informal daily spoken language by Arabs) [19].

Arabic NLP has been grown rapidly; a lot of applications have been developed in different domains such as: Information Retrieval, Information extraction, Machine Translation, Text to Speech, and Tutoring Systems (which is the main focus in this paper). As far as Arabic is concerned, researchers should consider the challenges and characteristics of the language of the language [16] such as ambiguity which is a major issues in improving that has to solved when developing Arabic NLP systems [20], [16], [18]. In our case, where we are concerned with ILTS, choosing incorrect analysis of the learner's input might lead to incorrect diagnosis of errors or issuing incorrect feedback [18].

III. EFFORTS IN DEVELOPING ILTS

ILTSs are systems that focus on the natural language of the learner and have the ability to effectively increase the efficiency by automating the learning processes. Incorporating

NLP techniques into ILTS will provide appropriate methods to evaluate students' answers by identifying the text-input errors and provide an immediate feedback to them. In this section, we discuss different ILTSs that have been implemented in different domains in different languages.

A. Development of Intelligent Language Tutoring Systems

1. Auto-Tutor

Auto-Tutor is a computer tutor that mimics the course patterns and educational tactics of a real human tutor via a dialogue with the learner using natural language. Auto-Tutor has been implemented by Tutoring Research group at the University of Memphis for teaching college students a computer literacy course [10], [21], [22], [35]. Auto-Tutor has been developed incrementally [21], [22], [35] where the latter has a 3D interactive interface and has been implemented using visual basic .NET and C#.

Auto-Tutor asks the leaner a couple of challenging questions that requires a paragraph of correct answers. Each question in the Auto-Tutor is associated with a particular set of *expectations* (ideal answers) and *misconceptions* (wrong answers) that are stored in a *curriculum script*. Auto-Tutor has been programed to be able to correct the *misconceptions* and provide an appropriate feedback to the learner. In order for Auto-Tutor to be capable of evaluating the learners' answers and match it with the *expectations* and *misconceptions*, LSA has been utilized as the pattern-matching method [35]. However, lexico-syntactic technique [11] is also used in order to overcome the limitations of LSA (as mentioned in section 2.2.1) via taking syntactic information into account while evaluating learners' answers.

2. Web-based Intelligent Language Tutoring Systems

In [23], *German Tutor* was presented as an ILTS that has been constructed to form the grammar practice for delivering a course in German via a web-based environment. Intelligence has been brought into the system through the German grammar and a parser that parses the learner's input. The student model is involved for maintaining all the learners' profiles and provides students with adaptive feedback that is suited to their expertise along with some proposed exercises. Efficiency has been attained through the robust design of the system, as shown in Fig.1, where the web-server interacts with the client, through Java and CGI scripts, while intelligent and adaptive mechanisms were built on the separate server side where the answer is processed.

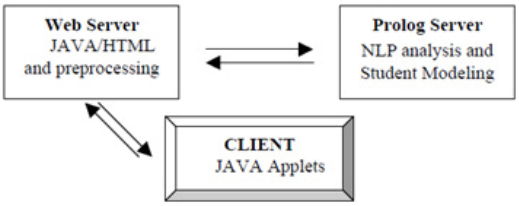

Fig.1: German Tutor Architecture [23].

The system has been evaluated through testing it with 19 students within 1 hour class. 84 % of the students reported that the system was very robust by providing them the immediate feedback and freer grammar practice.

3. Authoring Systems

Authoring systems are a kind of ILTS where the author/expert manually represents his/her knowledge in a natural language (i.e. s/he can add knowledge, detect and fix errors by her/himself without any kind of automation in achieving this task) [24], [12]. In order to reach the full functionality of ILTSs, different techniques have been proposed, see Table 2:

TABLE II. TECHNIQUES, APPROACHES AND LIMITATIONS OF AUTHORING SYSTEMS.

Technique used	Approach	Limitation
Concept-Grid + xPST	The approach requires the use of a lattice-style table-driven interface in order to construct templates that have two sets (required/correct concepts versus incorrect concepts). The proposed techniques provide just in-time feedback to the leaner based on his/her response. [12]	Dialogue between the leaner and the system was not provided; the system only evaluates the learners' responses. Another limitation is that, the approach focused intensively on words and numerical analysis instead of logic and grammar.
Natural-K	The approach requires exchanging the knowledge representation language of *Pyrenees* [25] (An ITS which teaches problem-solving strategies explicitly) with natural language. Natural-K is used to re-author the knowledge base of Pyrenees [24].	Dialogue was not provided between the learner and the system.

4. Extracting Learning Concepts from Educational text in an ITS

In [26], ITS was developed which spontaneously identifies the concepts to be learned by students from the learning contents for mathematics subject. The proposed system utilizes the statistical language model (which extracts the linguistics characteristics of a natural language) along with the conceptual map modeling in order to construct list of the learning concepts. The

learning domain consists of 10 corpora which contains two different subjects in mathematics. The statistical language model intends to approximate a probability distribution over the words that try to express how frequently a word appears in a text accurately.

The system has been evaluated through testing it for five different chapters from the two mentioned subjects. F-measure shows that the system's performance has achieved 90% before post-processing and 94% after post-processing.

5. Automated Language Tutoring Framework

In [27], a framework integrated in an ILTS is suggested which is based on *string search algorithm* for only extracting word-patterns from 'questions bank'. The proposed framework makes it easy for non-native speakers of the language to get enough language efficiency. The implementation of this framework tries to overcome some limitations from previous literatures such as: REAP [39] which is an ITS that does not allow a free-responses from the user. The limitations have been overwhelmed by incorporating some features such as *coherent* and *in-time feedback*, incorporated with *multimedia capabilities* that enable interactive voice responses and analyzing the text input from the learner. This is accomplished by utilizing auto-generated scripts that is created with the use of an authoring tool which in turn has been implemented in Java. ANTLR parser was used in order to generate parse trees based on the grammatical structure of the learner's response. This facilitates handling of semantic, syntactic, and contextual errors in responses. Diagnosing the errors during feedback along with analyzing the input-text from the learner makes the system more robust as comparing with previous tutors. The system achieves 93.89% as accuracy in detecting errors and its capability to handle many responses.

6. Beetle II System

Beetle II system is a tutorial dialogue system that has been implemented to accept input from the learner without restrictions and to provide experimentation with multiple tutorial planning and dialogue tactics. Learners interact with Beetle II system through an interactive chat interface via a keyboard. A natural language dialogue 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. Beetle II has been develped to ask the learners to illustrate their answers in order to give them a detailed feedback. It 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 [28].

7. CIRCSIM-Tutor

CIRCSIM-Tutor [29] is an ILTS in the area of cardiovascular physiology that employs a natural language dialogue with the learner via utilizing a collection of tutoring tactics that simulate the human-to-human tutoring. CIRCSIM-Tutor employs natural language for both input and output. The tactic that allows him to tackle several syntactic constructions and lexical items including: sentence segmentation and spell checking. The aim of building this system was to produce a text that is both linguistically and pedagogically reasonable. CIRCSIM-Tutor contains the following components: the text generator, the input understander, the planner, the problem solver, the student model, the screen manager and the knowledge base. The system follows the following four tutoring tactics (see Table 3), one for each plan:

TABLE III. THE FOUR TUTORING TACTICS FOR CIRCSIM-TUTOR.

Plan	Tactics
Tutor	Ask the student a set of questions
Give answer	Ask the student to demonstrate his/her answer.
Hint	Remind ("Remember that. . . ")
Acknowledge	4 possible cases (see below)

The four possible cases includes: ask the next question, assess the student's response, recognize the student's answer, if the answer is incorrect, either to give a hint or the correct answer [29]. CIRCSIM-Tutor proved that it overcomes all the hints' weaknesses and limitations that have been implemented in a system by [30] where the same failed teaching strategies were reused with another student.

B. Development of Arabic-based Intelligent Language Tutoring Systems

Many researchers have been contributed for the development of Arabic language in NLP. The outcomes of such researches have been utilized in implementing an Arabic ILTSs [34], [36].

In [31], a prototype system that uses expert systems technology is developed which represents Arabic syntax in Prolog as production rules. The proposed system has been tested in educational environment since it has the capability to detect some syntactic errors in Arabic. In [32] another expert systems-based system was proposed that could be utilized as a tool in constructing the rules of Arabic grammar which in turn could help in building Arabic NLP systems. In [33], the first Arabic grammar checker, called Arabic GramCheck, was proposed that its performance outperforms the Arabic

grammar checker embedded with Microsoft word version. Arabic GramCheck has identified main issues and challenges when developing a grammar checker for Arabic. Arabic GramCheck is capable to diagnose sophisticated grammatical errors and suggests feedback for them. The feedback is not meant for Arabic learner but rather for normal Arabic word processors users.

In [34], the first real Arabic ILTS was developed, so-called an Arabic Intelligent computer-assisted language learning (Arabic ICALL), in order to teach Arabic for primary schools students using advanced NLP techniques (morphological and syntax analyzers). The linguistic analyzers and the error analyser were proposed in order to provide sufficient feedback to students at Egyptian schools. The feedback system derives feedback by comparing the answer of the student along with the ideal answer that is produced by the system. The system makes it easy for the students to correct the input sentence by themselves through allowing them to locate the error made by them (i.e. the system assists the students to make use of their own mistakes).

Various Arabic ICALL systems were lacking to have deep error analysis, sophisticated error handling, and fast response; the reasons that enable [17] to develop an efficient error analyzer and error handler as an essential part towards enhancing Arabic ICALL systems. Suggested NLP tools include *morphological analyser*, *syntax analyser* and an *error analyser* in order afford feedback to students. The developed system offers an opportunity for online learners to correct the sentences independently which enables the learner to determine where his/her error is.

In [18], the authors addressed the important issue of morphological disambiguation of corrected interpretations of mistaken Arabic verbs that have written by beginners to intermediate Second Language Learners. An Arabic ILTS has been developed that is able to analyze both well- and ill-formed students' answers through analyzing each input from the student and then generate all the possible analysis. All the generated analysis will be send to the disambiguation system in order to predict the correct analysis according to morphological features. The corrected analysis will be used to identify the student's mistakes that can be raised from his/her input, and then Arabic ILTS will diagnose these mistakes (i.e. specifically identifying the error source and provide an immediate specific feedback).

As a continuation of the research carried out by [18], an Analysis and Feedback of Erroneous Arabic Verbs system was introduced by [36], which generalizes the findings from the Arabic ICALL system. As error diagnosis is not possible with current Arabic morphological analyzers, Constraint Relaxation and Edit Distance techniques are successfully employed to provide error-specific diagnosis and adaptive feedback to Arabic learners. The capabilities of these techniques in diagnosing errors related to Arabic weak verbs, which are formed using complex morphological rules, was demonstrated. Experimental results were satisfactory and the performance achieved was 74.34% in terms of the recall rate.

There has been a considerable progress in the field of Arabic ILTS thanks to the efforts made by researchers in Arabic speaking world. Though, the research community has witnessed a good progress in designing and implementing the Arabic ILTS, there are several issues which remain either unaddressed or partially solved due to the lack of linguistic resources and tools which create a critical obstacle when it comes to Arabic error diagnosis in particular.

IV. Discussion and Conclusion

Intelligent Tutoring Systems (ITSs) played a significant role in simulating the one-to-one tutoring with students. NLP focuses on how computers interpret and process the human natural language (either text or speech) in order to perform helpful applications. Two different issues have been discussed within this survey under NLP which includes: *Latent Semantic Analysis* (measuring the similarity between any two fragments of text), and *Entailment issues* (deciding whether the meaning of two pieces of a text is entailed to each other).

ILTSs by incorporating NLP techniques enable them to have the capability to evaluate and process the learner's input text, and provide pedagogical feedback. To this end, Arabic ILTSs are still in their initial stages compared to the work done on other languages, such as English and Germany, which we can benefit from the extensive research done in this field. Arabic share many of its characteristics with Semitic and morphologically rich languages. We have illustrated aspects of Arabic language in this context along with some difficulties that could encounter the development of NLP systems in general and ILTS in particular.

Using ILTS in the educational field makes it easy to discover the leaner's error and provide an immediate feedback for him/her along with allocating the error source (which is a major goal in the design of ILTS). We attempt in this paper to demonstrate the development of ILTSs generally, and a dedicated section has been specifically particularized for the development of Arabic ILTSs that still has a gap in the development of ILTS.

V. Future Prospects

In this section, we suggest some research areas which have not been covered in the current research domain and require more attention from researchers:

- Almost all of current studies in ILTSs are text-based dialogues which were focusing on analyzing the input-text from the learner. Further research could work on implementing an ILTS that interacts with voice-based dialogues.
- Concept Grid as an authoring tool has been focused on evaluating students' answers and to provide an immediate feedback. Further research could extend these capabilities by incorporating a dialogue between the system and the learner.
- A good research opportunity could be raised from this survey which is building an Arabic web-based ILTS for teaching Arabic language for non-native speakers in order to reach a wider range of people over the internet.

Almost all of the ILTSs that incorporated a dialogue with learner have been done in English as compared to other languages. It is useful to incorporate other difficult to learn languages, such as Arabic, in order to explore more in the languages' features while dealing with the learner.

References

[1] D'Mello, S., & Graesser, A. (2013). Design of Dialog-Based Intelligent Tutoring Systems to Simulate Human-to-Human Tutoring. In *Where Humans Meet Machines* (pp. 233-269). Springer New York.

[2] Yazdani, M. (1986). Intelligent tutoring systems survey. *Artificial Intelligence Review*, *1*(1), pp. 43-52.

[3] Nwana, H. S. (1990). Intelligent tutoring systems: an overview. *Artificial Intelligence Review*, *4*(4), 251-277.

[4] Martens, A., & Uhrmacher, A. M. (2004). Modeling of Tutoring Processes in Intelligent Tutoring Systems. In *KI 2004: Advances in Artificial Intelligence* (pp. 396-409). Springer Berlin Heidelberg.

[5] El-Khouly, M. M., Far, B. H., & Koono, Z. (2000). Expert tutoring system for teaching computer programming languages. *Expert Systems with Applications*, *18*(1), pp. 27-32.

[6] Chowdhury, G. G. (2003). Natural language processing. *Annual review of information science and technology*, *37*(1), 51-89.

[7] Yue, X., Di, G., Yu, Y., Wang, W., & Shi, H. (2012). Analysis of the combination of natural language processing and search engine technology. *Procedia Engineering*, *29*, 1636-1639.

[8] Hofmann, T. (1999, July). Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence* (pp. 289-296). Morgan Kaufmann Publishers Inc.

[9] Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, *25*(2-3), pp. 259-284.

[10] Graesser, A. C., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Tutoring Research Group, T. R. G., & Person, N. (2000). Using latent semantic analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments*, *8*(2), pp. 129-147.

[11] Rus, V., & Graesser, A. C. (2006, July). Deeper natural language processing for evaluating student answers in intelligent tutoring systems. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE* (Vol. 21, No. 2, p. 1495). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

[12] Devasani, S., Blessing, S. B., & Gilbert, S. (2011, January). Lattice-based approach to building templates for natural language understanding in intelligent tutoring systems. In *Artificial Intelligence in Education*, pp. 47-54. Springer Berlin Heidelberg.

[13] Monz, C., & de Rijke, M. (2001). Light-weight entailment checking for computational semantics. In *Proc. of the third workshop on inference in computational semantics (ICoS-3)*.

[14] Moldovan, D., & Rus, V. (2001, July). Logic form transformation of wordnet and its applicability to question answering. In *ACL* (Vol. 1, pp. 402-409).

[15] Pazienza, M. T., Pennacchiotti, M., & Zanzotto, F. M. (2005, April). Textual entailment as syntactic graph distance: a rule based and a SVM based approach. In *Proceedings of the Pattern Analysis, Statistical Modelling, and Computational Learning (PASCAL) Challenges Workshop on Recognising Textual Entailment*.

[16] Farghaly, A., & Shaalan, K. (2009). Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, *8*(4), 14.

[17] Shaalan, K., Talhami, H. (2006) Arabic Error Feedback in an Online Arabic Learning System, Advances in Natural Language Processing, *Research in Computing Science (RCS) Journal*, National Polytechnic Institute of Mexico, ISSN 1665-9899, 18: 203-212.

[18] Shaalan, K., Magdy, M., & Samy, D. (2010, May). Towards Resolving Morphological Ambiguity in Arabic Intelligent Language Tutoring Framework. In *Workshop on Supporting eLearning with Language Resources and Semantic Data* (p. 12).

[19] Shaalan, K. (2013). A Survey of Arabic Named Entity Recognition and Classification, Computational Linguistics, MIT Press.

[20] Othman, E., Shaalan, K., & Rafea, A. (2004, September). Towards resolving ambiguity in understanding arabic sentence. In *International Conference on Arabic Language Resources and Tools, NEMLAR, Egypt*.

[21] Graesser, A. C., Person, N., Harter, D., & Tutoring Research Group. (2001a). Teaching tactics and dialog in AutoTutor. *International Journal of Artificial Intelligence in Education*, *12*(3), pp. 257-279.

[22] Graesser, A. C., VanLehn, K., Rosé, C. P., Jordan, P. W., & Harter, D. (2001b). Intelligent tutoring systems with conversational dialogue. *AI magazine*, *22*(4), 39.

[23] Heift, T., & Nicholson, D. (2000, January). Theoretical and practical considerations for web-based intelligent language tutoring systems. In *Intelligent Tutoring Systems*, pp. 354-362. Springer Berlin Heidelberg.

[24] Jung, S. Y., & VanLehn, K. (2010, January). Developing an intelligent tutoring system using natural language for knowledge representation. In *Intelligent Tutoring Systems*, pp. 355-358. Springer Berlin Heidelberg.

[25] VanLehn, K., Bhembe, D., Chi, M., Lynch, C., Schulze, K., Shelby, R., ... & Wintersgill, M. (2004, January). Implicit versus explicit learning of strategies in a non-procedural cognitive skill. In *Intelligent Tutoring Systems*, pp. 521-530. Springer Berlin Heidelberg.

[26] Günel, K., & Aşlıyan, R. (2010). Extracting learning concepts from educational texts in intelligent tutoring systems automatically. *Expert Systems with Applications*, *37*(7), pp. 5017-5022.

[27] Thirugnanasambantham, C. (2011, December). An Intelligent System Framework for an Automated Language Tutoring Tool. In *Informatics and Computational Intelligence (ICI), 2011 First International Conference on*, pp. 122-125. IEEE.

[28] Dzikovska, M. O., Bental, D., Moore, 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*, pp. 620-625. Springer Berlin Heidelberg.

[29] Evens, M. W., Chang, R. C., Lee, Y. H., Shim, L. S., Woo, C. W., Zhang, Y., ... & Rovick, A. A. (1997, March). CIRCSIM-Tutor: An intelligent tutoring system using natural language dialogue. In *Proceedings of the fifth conference on Applied natural language processing: Descriptions of system demonstrations and videos*, pp. 13-14. Association for Computational Linguistics.

[30] Woo, C. W. (1992). *A Multi-Level Dynamic Instructional Planner for an Intelligent Physiology Tutoring System*. ILLINOIS INST OF TECH CHICAGO DEPT OF COMPUTER SCIENCE.

[31] Hegazi, N., Ali, G., Abed E.M., Hamada S. (1989). Arabic Expert System for Syntax Education, *In the Proceedings of the Second Conference on Arabic Computational Linguistics*, pp. 596-614 Nov., Kuwait.

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

[33] Shaalan, K. (2005). Arabic GramCheck: A grammar checker for Arabic. *Software: Practice and Experience*, *35*(7), pp. 643-665.

[34] Shaalan K. (2005). An Intelligent Computer Assisted Language Learning System for Arabic Learners, Computer Assisted Language Learning: An International Journal, Taylor & Francis Group Ltd., 18(1 & 2): 81-108.

[35] Graesser, A. C., Chipman, P., Haynes, B. C., & Olney, A. (2005). AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *Education, IEEE Transactions on*, *48*(4), 612-618.

[36] Shaalan, K., Magdy, M., Fahmy, A. (2013). Analysis and Feedback of Erroneous Arabic Verbs, Journal of Natural Language Engineering (JNLE), Cambridge University Press, UK. DOI: 10.1017/S1351324913000223.

[37] Abdel Monem, A., Shaalan, K., Rafea, A., Baraka, H. (2008). Generating Arabic Text in Multilingual Speech-to-Speech Machine Translation Framework, Machine Translation, Springer, Netherlands, 20(4): 205-258.

[38] Shaalan, K., Abo Bakr, H., Ziedan, I. (2009). A Hybrid Approach for Building Arabic Diacritizer, In the Proceedings of the 12th European Chapter of the Association for Computational Linguistics (EACL 2009) Workshop on Computational Approaches to Semitic Languages, Association for Computational Linguistics, PP. 27-35, 31 March, Athens, Greece.

[39] Heilman, M., & Eskenazi, M. (2006). Language learning: Challenges for intelligent tutoring systems. In *Proceedings of the workshop of intelligent tutoring systems for ill-defined tutoring systems. Eight international conference on intelligent tutoring systems* (pp. 20-28).

[40] Shaalan K., Rafea, A., Abdel Monem, A. & Baraka, H. (2004). Machine Translation of English Noun Phrases into Arabic, *The International Journal of Computer Processing of Oriental Languages (IJCPOL)*, World Scientific Publishing Company, 17(2):121-134.