LANA-I: An Arabic Conversational Intelligent Tutoring System for Children with ASD

Sumayh Aljameel[1][✉], James O'Shea[2], Keeley Crockett[2], Annabel Latham[2], and Mohammad Kaleem[2]

[1] Department of Computer Science, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia
saljameel@iau.edu.sa
[2] Department of Computing, Math and Digital Technology, Manchester Metropolitan University, Manchester, UK
{j.d.oshea,k.crockett,a.latham,m.kaleem}@mmu.ac.uk

Abstract. Children with Autism Spectrum Disorder (ASD) share certain difficulties but being autistic will affect them in different ways in terms of their level of intellectual ability. Children with high functioning autism or Asperger syndrome are very intelligent academically but they still have difficulties in social and communication skills. Many of these children are taught within mainstream schools but there is a shortage of specialised teachers to deal with their specific needs. One solution is to use a virtual tutor to supplement the education of children with ASD in mainstream schools. This paper describes research to develop a novel Arabic Conversational Intelligent Tutoring System, called LANA-I, for children with ASD that adapts to the Visual, Auditory and Kinaesthetic learning styles model (VAK) to enhance learning. This paper also proposes an evaluation methodology and describes an experimental evaluation of LANA-I. The evaluation was conducted with neurotypical children and indicated promising results with a statistically significant difference between user's scores with and without adapting to learning style. Moreover, the results show that LANA-I is effective as an Arabic Conversational Agent (CA) with the majority of conversations leading to the goal of completing the tutorial and the majority of the correct responses (89%).

Keywords: Autism · Intelligent tutoring system · String similarity · Arabic language

1 Introduction

The number of children being diagnosed with autism spectrum disorder (ASD) is increasing [1]. Children with high functioning Autism (HFA) or Asperger's Syndrome (AS) (i.e. those with higher verbal IQ) are usually offered education in the mainstream schools. However, many mainstream schools are not able to include students with ASD because of the lack of skilled teachers and the poor training and provisions from the responsible institutions [2]. In addition, traditional education using human tutors is a challenge for students with autism, who have difficulties in communication and social

interactions. Many researchers have reported that using a virtual tutor with students with ASD could meet the individual student's needs [3, 4].

Conversational Intelligent Tutoring System (CITS) is a software system which uses natural language interfaces to allow users to learn topics through discussion as they would in the classroom. Many CITS have been developed for different domains. To our knowledge, no academic research exists on Arabic CITS developed specifically for Autistic children. LANA-I [5] is an Arabic CITS, which engages autistic children with a science tutorial where the curriculum material is mapped to VAK model. One challenge in building such a system is the requirement to deal with the Arabic language grammatical features and its morphological nature. The research into Conversational Agent (CA) development techniques revealed that hybrid approach was the most appropriate approach to develop an Arabic CA [6]. The engine of LANA-I is based on the two main CA development strategies, A Pattern Matching (PM) engine and a Short Text Similarity (STS) algorithm that calculate the matching strength of a pattern to the user utterance. The two parts of the engine work together to overcome some of the unique challenges of the Arabic language and to extract responses from resources in a particular domain (Science topic). The main contributions in this paper are:

- A novel architecture for an Arabic CITS using VAK model for an appropriate education scenario.
- The results of designing an experimental methodology to validate the educational tutoring scenario in the Arabic CITS.

In order to evaluate LANA-I, an initial pilot study was conducted on the general population. This study took place in UK with neurotypical children from the target age group (10–12) years whose first language is Arabic. It is important to test LANA-I, with the general population before testing it with Autistic children to avoid any problems and issues that may occur in the tutorial or confusion in the presentation of the tutorial material. The study used the learning gain measurement (defined in Sect. 3) to evaluate the ability of CITS to adapt to a child's learning styles.

This paper is organized as follows: Sect. 2 describes the architecture and methodology of implementing LANA-I. Section 3 explains the experimental methodology and the experiments. Section 4 provides the results and discussion. Finally, Sect. 5 presents the conclusions and future works.

2 LANA-I Architecture and Implementation

LANA-I was developed based on two phases. The first phase involved designing and implementing an Arabic CA, and the second phase focused on development of an educational tutorial on science, mapping the tutorial to the VAK model and introducing the Arabic ITS interface to the CA. In the first phase, a new architecture for developing an Arabic CA was developed using both PM and STS. The key features are summarised as follows:

- Ability to control the conversation through context.
- Ability to personalise the lesson with the user's learning style and provide suitable supporting material to the user.
- A scripting language to provide Arabic dialogue for LANA-I.
- A novel CA engine that manages the response using a combination of the PM technique and the STS technique.
- Managing the response when the context is changed. For example, creating the right response when the user writes something that is not related to the tutorial topic.

The proposed framework for the Arabic CA consists of five components as shown in Fig. 1:

1. Graphical User Interface: Responsible for the communication between the user and CA (in this case the CITS tutor) through a web interface with panels to display supporting material.
2. Controller: The controller manages the conversation between the user and the Arabic CA, as well as cleaning the user utterance by removing diacritics and other illegal characters (e.g. ! £ $).
3. Conversation Agent Manager: Responsible for controlling and directing the conversation through contexts. In addition, the manager ensures that the discussion is directed towards completion of the tutorial.
4. Conversation Agent engine: Responsible for pattern matching and calculating the similarity strength between the user utterance and the scripted patterns.
5. The knowledge base: The knowledge base is responsible for holding the tutorial domain knowledge in a relational database, which includes CA scripts, Log file, and General contexts such as weather, agreement and rude words.
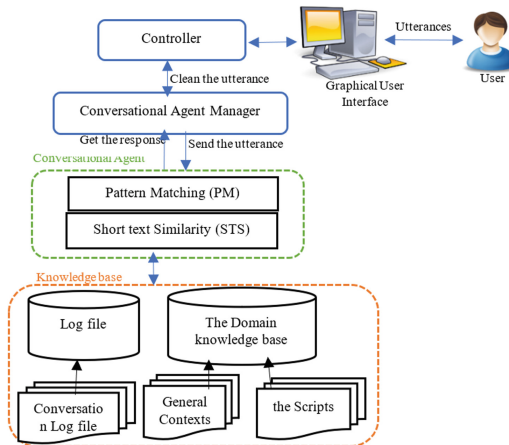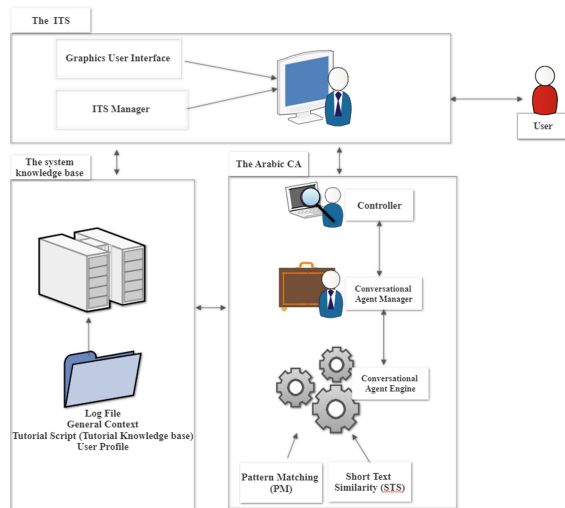

Fig. 1. The LANA-I CA architecture

In the second phase the ITS architecture is designed to adapt to the user's learning style within the Arabic CA. Based on the typical ITS architecture [7], LANA-I ITS architecture consists of four main components as shown in Fig. 2:

1. User Interface Model: responsible for the interaction between the user and the ITS components.
2. Tutor Model: the ITS manager, which is the main component that interacts with the user through the GUI, and personalises the tutorial based on the user's learning style.
3. Student Model: a temporal memory structure, which records the user's responses during the tutoring session and the student's profile such as user ID, user's age, gender, user's learning style, and Pre-test and Post-test scores.
4. Domain Model: the Tutorial Knowledge Base, which contains structured topics that are presented to the user.
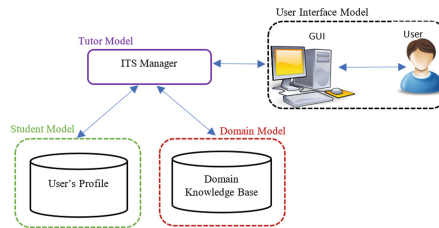
Fig. 2. LANA-I ITS Architecture.

The LANA-I architecture combines the Arabic CA (Fig. 1) and ITS architecture (Fig. 2). The proposed LANA-I CITS architecture, shown in Fig. 3 contains three main components that are described in the following sections:

Fig. 3. LANA-I CITS architecture

2.1   The Knowledge Base

The knowledge base consists of four sub-components: (1) the Tutorial Knowledge base, (2) Arabic general contexts (e.g. weather, and greetings), (3) user's profile, and (4) the log file. The tutorial domain CA was from the Science curriculum for ages 10-12. The material was provided by the Ministry of Education in Saudi Arabia [8]. Script contexts were structured according to the topics in the science book taught in schools in Saudi Arabia, such as the Earth, Moon, Solar System, eclipse, etc. A short interview was conducted to design the knowledge base with three primary school teachers in Saudi Arabia, who teach Science. The aim of this interview was to extract knowledge of lesson design and delivery of tutorials similar to the traditional classroom delivery model. The teachers evaluated the designed tutorial in order to give feedback and approval. The tutorials were implemented within LANA-I using a relational database MySQL to store and retrieve the resources. The LANA-I knowledge base consists of two contexts: the domain, which is the science tutorials, and a general context, illustrated in Fig. 4.
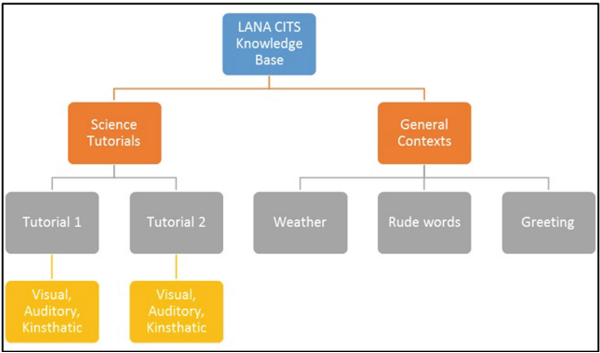

Fig. 4. Knowledge base contexts

In Fig. 4, each context has all related sub contexts mapped to it. The science tutorials hold the tutorials sessions that are related to the science subject such as solar system, earth, and moon. General contexts deal with general conversation that is not related to the domain, such as weather, greeting, rude words, and user leaving words (any words or sentences means that the user will leave the conversation). The general context makes LANA-I seem more aware by responding to the user's utterances, which are not related to the main domain. However, only a few general sub-contexts have been implemented as a proof of concept within the scope of this research.

Once the tutorial was designed and approved by the primary school teachers then the tutorial questions were mapped to the VAK learning style model. Each question in the tutorial is mapped to Visual learning style through pictures and videos, Auditory learning style through sound recording, Kinaesthetic learning style through models and instructions as shown in Table 1.

Table 1. Part of the tutorial and how it is mapped to the VAK model

Question: What cause the four seasons? Answer: Seasons result from the yearly orbit of the Earth around the Sun and the tilt of the Earth's rotational axis relative to the plane of the orbit	V: video A: Audio K: solar system model
Question: How long the earth takes to complete the rotation around the sun? Answer: 365 days, which is one year	V: video A: Audio K: earth-moon model
Question: The moon is the nearest object to the earth but it is different from the earth in many ways. What are the differences between the earth and the moon? Answer: The moon doesn't have atmosphere and water whereas the earth has	V: video A: Audio K: earth-moon model

2.2 Arabic Scripting Language

The three different approaches to develop an Arabic CA and a number of challenges faced by Arabic language were discussed in survey paper [6]. It was concluded that there is a lack of Arabic NLP resources leading to limited capabilities within Arabic CAs. Consequently, this research needs a new scripting language to enable the Arabic CA combined with an ITS to deliver Arabic tutoring sessions using the modern standard Arabic language.

The foundation of LANA-I scripting language is based on the InfoChat scripting language [9]. InfoChat was designed using English scripting language and based on the pattern matching (PM) technique, where the domain is organised into a number of contexts and each context contains rules, each rule in the domain contains a number of patterns and a response that forms the CA output to the user. In the LANA-I CA engine, the similarity strength is calculated through combining the use of the PM, which is based on InfoChat method, and the Cosine algorithm [10] to compute the similarity strength between the utterance and the scripted patterns in order to improve the CA accuracy. The highest matching strength pattern will generate the response back to the user. Based on the scripting methodology reported by Latham [11] the procedures to create the scripts within the Knowledge base are:

1. The methodology for scripting each context is:

   - Create a context table, which has a record with a unique name to represent that context (topic).
   - When the context is invoked, an initialisation rule is fired.
   - All rules and patterns are scripted for the associated context.
   - Test each context to check that the expected rule is fired for a sample of user utterances.

2. The methodology for scripting each rule is:

- Create a rule table, which has a unique rule name to represent the tutorial question.
- For each rule, create patterns that match user utterances. Extract the important words and use the wildcards to replace unimportant words.
- Create patterns from each different utterance phrase. E.g. saying the same things using alternative words.
- Create the CA response to the utterance.
- Add Image, Audio, and instructions to the rule in order to map the learning style.

2.3   Scripting Arabic CA for LANA-I

In LANA-I, the tutorial topics were represented as the contexts and the agent's questions for such topic were represented as the rules, while the pattern represent the user's utterances, which belong to such a rule. The scripting language in LANA-I includes the following features:

- Provide supporting material to the user depending on the user's learning style (Visual: images and videos – Auditory: Sounds – Kinaesthetic: Instructions and objects). The learning style will be determined using a child friendly-customised VAK learning style preferences questionnaire be-fore the tutorial.
- This material is stored in the scripting database and once a rule is fired, the corresponding material is provided to the user through the interface. All images, videos and audios provide the right answer. All of them have the same knowledge that the written text provides.
- The scripting language works with the ITS manager to check the user's learning style and provide the suitable material with the fired rule.

Table 2 shows an example of the scripting language and how the tutorial mapping the VAK model. When the user is visual learner, the rule is fired with the video or image material. In this example (Table 2), the rule has video that will be executed

Table 2. Scripting language (translated)

Context: 1	Context: 1
Context Name: Topic 1 (solar system)	Context Name: Topic 1 (ال نظام ال شم سي)
Rule Number: 1	Rule Number: 1
Rule Name: The earth rotates around the sun. what the result of this rotation?	Rule Name: تدور الأرض حول الشمس كل سنه. ماهي نتيجة هذا الدوران؟
Pattern: * four seasons	Pattern: الفصول الأربعة*
Pattern: * 4 seasons	Pattern: المواسم الأربعة *
Pattern: four seasons *	Pattern: أربع فصول24 ساعه *
Response: *Well done,* the earth rotates around the sun every year and this cause the four seasons.	Response: أحسنت بارك الله فيك. تدور الارض حول الشمس وينتج عنه الفصول الأربعة
Image: yes	Image: yes
Audio: yes	Audio: yes
Instruction: Look at the solar system model and its object	Instruction: أنظر إلى نموذج النظام الشمسي والأجرام المحيطه بها

along with the rule. When the user is an Auditory learner, the rule will be fired with the audio material. The same thing with the kinaesthetic learning style, the rule will be fired with instructions to use the models. Figure 5 shows the models that are used with the Kinaesthetic learning style.
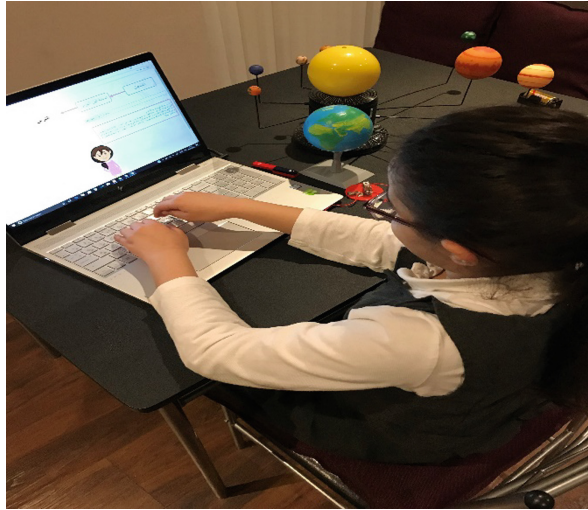
Fig. 5. The solar system model and earth model

2.4 The Arabic Conversational Agent

The second component of LANA-I (Fig. 3) is the Arabic CA. The main components within the Arabic CA architecture (Fig. 1) are: (i) Controller, (ii) Conversational Agent Manager, and (iii) Conversational Agent Engine.

The controller directs and manages the entire conversation by working with several other components, which are ITS manager, GUI and CA manager to achieve the conversation goal. In the beginning, the controller finds the student's learning style by communicating with the ITS manager. Then, the controller receives the user's utterance and provides an utterance checking process based on the following constraints:

- Filter the utterance: The controller filters the utterance to remove any special characters (i.e. $, &, *, !, ?, "", £, (), ^). For example (as shown in English), if the user writes: (Hi, how are you Lana???) the system will convert it to (Hi how are you Lana).
- Check for rude and offensive words: if the utterance contains any of rude or offensive words, the system will warn the user three times before terminating the session.

The controller then checks if the conversation is within the tutorial scenario or not by communicate with the CA. The response will be delivered back to the user with

providing supporting material such as video, picture, audio, or instructions depending on the user's learning style as shown in Fig. 6.

Fig. 6. Conversation between LANA-I and visual learner

The Conversation Manager is responsible for controlling the conversation to make sure that the goal is achieved. The goal is completing the tutorial by checking whether the user stays on the tutorial topic, or the user switches the context i.e. asks about something irrelevant to the tutorial. The Arabic CA implemented in LANA-I adopts a goal-oriented methodology [11].

The Conversational Agent Engine contains a combination of methods of string similarity and PM approaches to determine the similarity between two sets of strings within CA's, while traditional CA's used only a PM approach that involves a strength calculation through different aspects of the user utterance and the scripted pattern such as activation level and number of words, etc. It is important to use string similarity methods to overcome some of the challenges in the Arabic language (described in Sect. 2.3). In the field of CA development, the scripting is the most time consuming part of CA development [12]. The biggest challenge of scripting CAs is the coverage of all possible user utterances [11]. The engine handles the challenge of Arabic scripting by combining the Wildcard PM Function with the string similarity algorithm that calculates similarity strength and overcomes the scripting length challenge. Consequently, strings with minor differences should be recognized as being similar. The LANA-I CA engine has two main components:

- Pattern matching approach (Wildcard PM [9]).
- String similarity algorithm (Cosine algorithm [10])

These components were used to calculate the similarity between the user utterance and the scripted patterns in order to reduce the need to cover all possible utterances when scripting the domain.

Pattern matching is based on similar method to InfoChat [9]. In PM technique, the user utterance will be matched to the stored patterns; these patterns contain wildcard

characters to represent any number of words of characters. An example of that, the wild card symbol (?) matches a single character, where the wild card symbol (*) matches any number of words. An example for the PM approach in LANA-I is illustrated in the following table:

Table 3. Example of matching the utterance with the patterns script

Rule: Although the moon can appear a very bright white, its surface is actually dark. So from where the moon reflects the light?	**Rule:** القمر لايضئ بنفسه فهو انعكاس لضوء آخر . ماهو؟
Pattern 1: the moon reflects * sun **Pattern 2**: * the sunlight **Pattern 3**: * the sunshine	**Pattern 1:** القمر يعكس * الشمس **Pattern 2:** * ضوء الشمس **Pattern 3:** * نور الشمس
Utterance 1: it is the sunlight **Utterance 2:** reflects the moon the sun.	**Utterance 1:** هو ضوء الشمس **Utterance 2:** يعكس القمر الشمس

In Table 3, the utterance#1 will match pattern#2 (* the sunlight), where the wild card symbol (*) will match any number of words. The results from this matching will be 1 whereas the utterance#2 will not match any patterns because of the word order. In this case the matching result will be 0.

The second component of the Arabic CA engine uses the STS algorithm, which for the purpose of this work is the Cosine similarity measure [10]. The similarity between two pieces of text is determined by representing each piece of text in the form of word vector. A word vector is a vector of length N where N is the number of different tokens in the text. The similarity is computed as the angle between the word vectors for the two sentences in vector space.

For two texts t_1 and t_2 the cosine similarity is illustrated in the following equation:

$$\text{SIM}(t_1.t_2) = \frac{\sum_{i=1}^{n} t_1 t_2}{\sqrt{\sum t_{1i}^2} \times \sqrt{\sum t_{2i}^2}} \qquad (1)$$

The cosine similarity result is nonnegative and bounded between [0,1] where 1 indicates that the two texts are identical.

2.5 The Workflow of LANA-I CA Engine

In the beginning, the PM Wildcard will be used to match the user utterance with the patterns stored in the database. If the match is not found, the STS Cosine similarity will be applied. For example, assume the pattern stored in the LANA-I database was (S1), while the user utterance was (S2), as shown in Table 4:

The utterance is not recognised by the PM approach because of the word order or minor differences from the pattern. Therefore, the system applies the Cosine similarity, which is illustrated in the following steps:

Table 4. Example of calculating the similarity

Stored pattern in LANA-I: S1= يختلف القمر عن الأرض بأن ليس له غلاف جوي	S1: The moon is different from the earth that it does not have an atmosphere.
User Utterance: S2= القمر ليس له غلاف جوي لذلك يختلف عن الأرض	S2: The moon does not have atmosphere therefore it is different from the earth.

- Create Matrix[][] where the columns are the unique words from S1 and S2, and the rows are the words sequence of S1 and S2.
- Calculate the similarity between each word, 1 means the two words are identical, 0 means the two words are different.

The result obtained from Cosine similarity when applied to S1 and S2 is equal (0.88), which is greater than the threshold (0.80). This threshold is empirically determined using a small set of sentence pairs as in [13]. When the user's utterance is recognized by the similarity measure, the corresponding response will be generated and delivered to the user.

2.6 LANA-I ITS

The third component in the LANA-I architecture (Fig. 3) contains: The Graphics User Interface (GUI), and the ITS manager.

Fig. 7. LANA-I GUI and its character

The GUI is the point where LANA-I and the user interact with each other. A character called LANA (shown in Fig. 7) is used in LANA-I. This character appears in all the system interfaces to make the conversation more natural and engaging for the users. The LANA character was designed by the author and then evaluated by primary school teachers in Saudi Arabia in order to make the tutorial more engaging.

The ITS manager adapts the tutorial depending on to the user's learning style VAK, which is determined at the beginning of the tutorial through a questionnaire. The

questionnaire was developed on the basis of a widely disseminated version of Smith's visual, auditory, and kinaesthetic styles [15] to be completed by pupil with parents or teacher help. There were three questions focused on Smith's visual, auditory, and kinaesthetic styles (VAK). For each question, the pupils had to respond 'yes' or 'no' to each question. The total questions in the questionnaire are nine and the student's learning style result will be based on the highest score in one area.

2.7   LANA-I Workflow

This section describes the LANA-I workflow from perspectives of teacher and the pupil in order to understand how each activity communicates with others. Figure 8 presents the workflow.

Initially LANA-I starts from the learning style questionnaire, which is taken by the teacher. In the registration screen, the system asks the teacher to enter the pupil ID number (provided by the researcher), age, gender, and the result of the learning style questionnaire, conducted before using the system. There are four options for the learning style: Visual, Auditory, Kinaesthetic, and No-learning style, which used with the control group in the testing stage. After completing this stage, the system shows the pre-test interface, and asks the user to complete the test. The next interface after submitting the test is the CA tutorial, The ITS manager is responsible in this stage for personalising the tutoring session according to the user's learning style by providing the CA components, through the controller, the suitable materials from the Knowledge base component. The ITS manager also saves the user's registration information and the pre-test score in the log file/student's profile. In the CA screen, the user is led through the topic from one question to another until the end of the tutoring session. The user has only one attempt to answer. After completing the tutorial, the user is asked to fill the post-test questions and the score is saved by the ITS manager.

Fig. 8. LANA-I workflow

3   Experimental Methodology

The LANA-I prototype was tested through two main experiments to evaluate the system. The evaluation was based on a set of objective and subjective metrics [16]. The objective metrics were measured through the log file/temporal memory and the pre-test and post-test score. The subjective metrics were measured using the user feedback questionnaire. The aim of capturing the subjective and objective metrics is to test the hypotheses, which relate to the effectiveness of the conversation and using the VAK learning style, end user satisfaction, usability and system robustness. The main hypothesis of the experiments is:

- HA_0: LANA-I using VAK model cannot be adapted the student learning style.
- HA_1: LANA-I using VAK model can be adapted the student learning style.
- HB_0: LANA-I is not an effective Arabic CA.
- HB_1: LANA-I is an effective Arabic CA.

The aim of the first experiment was to test Hypothesis A (LANA-I using VAK model can adapt the tutorial to the student's learning style). The second experiment was conducted to test the Hypothesis B (LANA-I is an effective Arabic CA).

3.1   Participants

The total size of the sample was 24 neurotypical participants within the age group (10–12) years and their first language was Arabic. The participants recruited for the evaluation were residents of the Greater Manchester area within the UK and none of them had previous experience of using LANA-I. All participants' parents received an information sheet about the project and its aims, and a consent form to obtain their permission before conducting the experiment. The participants were divided into two groups. The first group is a control group (n = 12), who used the LANA-I without adapting to the learning style VAK model as basis comparison. The second group is an experimental group (n = 12), who used LANA-I with adapting to the learning style VAK model. From 15 participants, 12 were selected based on their results on the learning style questionnaire in order to divide them into 3 groups of 4.

3.2   Experiment 1: LANA-I Tutoring with and Without VAK Learning Style

Subjective and objective metrics were used to answer the two questions related to Hypothesis A. The experiment is based on the pre-test and post-test scores with and without adapting to the VAK learning Style model. Participants were divided into 2 groups. Each group of participants was asked first to register into the system and complete the pre-test questions. The Control Group used LANA-I without adapting the VAK learning style model. They started the tutorial without the VAK questionnaire, whereas the experimental Group, who used the LANA-I with adapting to VAK learning style model, were asked to fill the VAK learning style questionnaire in order for LANA-I to be adapted to the learning style. After adapting the learning style, the tutorial provided the most suitable material during the session such as videos, images or

instructions and physical resources. The Control Group did the tutorial session based on a text conversation without any additional materials. When the session ended, both groups did the post-test questions in order to measure their learning gain. Finally, both groups completed the user feedback questionnaire.

3.3 Experiment 2: LANA-I CA System Robustness

The data for this experiment was gathered from the LANA-I log file and the user feedback questionnaire whilst participants were completing experiment 1. The subjective and objective metrics were used to answer questions related to Hypothesis B. The data gathered from the log file allows assessment of the performance of LANA-I CA and the algorithms deployed in the architecture. This data will measure success using objective metrics. The data from user feedback questionnaire will be analysed in order to measure success using subjective metrics.

4 Results and Discussion

The learning gain was measured using a pre-test and post-test approach [17–19]. The same test questions were completed before and after the tutoring conversation. The test scores were compared to establish whether there is any improvement as follows:

$$Learning\ gain\ =\ post-test\ score\ -\ pre-test\ score \tag{2}$$

The results from experiment 1, illustrate that there is a statistically significant difference (as shown in Table 5) between the user's test scores with and without adapting to the VAK learning style. The mean rank of the differences between the pre-test and post scores in each case are without VAK (M = 9.50) and with VAK (M = 15.50). The P value of the mean's variation between the two groups is 0.03 (significant at p <= 0.05), that meaning the users performed better when they are using their preferred VAK learning style.

Table 5. Mann-Whitney U test for the differences in learning gain between the control group and the experimental group.

	Group	N	Mean rank	Sum of ranks
Learning gain	Control	12	9.50	114.00
	Experimental	12	15.50	186.00
	Total	24		
Test statistics[a]				
			Learning gain	
Mann-Whitney U			36.000	
Wilcoxon W			114.000	
Z			−2.173	
Asymp. Sig. (2-tailed)			.030	
Exact Sig. [2*(1-tailed Sig.)]			.039[b]	

[a]Grouping Variable: VAK
[b]Not corrected for ties.

The effect size (r) is calculated using the Eq. (3) and found that r = 0.4 indicating a medium to large effect size using [20] criteria of 0.1 = small effect, 0.3 = medium effect, 0.5 = large effect. It can be concluded that there is a statistically significant improvement in the learning gain in the Experimental Group comparing to the Control Group.

$$r = Z \,/\, square\ root\ of\ N\ where\ N = total\ number\ of\ cases \qquad (3)$$

Relative learning gain was also measured in this experiment. Relative learning gain is a measure that calculate the average improvement in test scores as a percentage of the possible improvement [21]. This measure additionally takes into account the opportunity for improvement. Average test score improvements were calculated and compared using the following formula:

$$Relative\ learning\ gain = (PostTest - PreTest)/(TotalScore - PreTest) \qquad (4)$$

For example, if student get 8/10 in pre-test and only improves by 1, this is different to another getting 3/10 in pre-test and only improving by 1 – improvement is 50% in first case but only 14.2% in the second case. Table 6 illustrates the results of the Mann-Whitney U test conducted in order to measure the relative learning gain between Control Group and Experimental Group. It shows that there is a difference in the mean value between Control Group and Experimental Group. The average improvement in test scores in the Experimental Group was improved more than the Control Group. The ranks in each case are: Control Group (M = 9.04) and Experimental Group (M = 15.96). The relative learning gain between the Control Group and the Experimental Group is statistically significant different, p-value less than 0.05 (P = .015). The effect size (r) is calculated using the Eq. (3) and found that r = 0.5 indicating a large effect size using [20] criteria.

Table 6. Mann-Whitney U test for the relative learning gain between control group and experimental group.

	Group	N	Mean rank	Sum of ranks
Relative Learning gain	Control	12	9.04	108.50
	Experimental	12	15.96	191.50
	Total	24		
Test Statistics[a]				
		Relative learning gain		
Mann-Whitney U		30.500		
Wilcoxon W		108.500		
Z		−2.428		
Asymp. Sig. (2-tailed)		.015		
Exact Sig. [2*(1-tailed Sig.)]		.014[b]		

This result highlights that the participants performed better when they have adapted to their preferred VAK learning style. It can therefore be concluded that there is statistically a significant difference between the pre-test and post-test scores with and without adapting to VAK learning style, indicating that the HA_1 can be accepted.

Another test was conducted to find out whether there was a significant difference in the scores for participant's perception of remembering what they have learned from LANA-I. The users who learned with adapting to VAK learning style are much happier with remembering what they have learned from LANA-I (91.7%) than users who learned without adapting to VAK (50%) as shown in Fig. 9. This result indicates that HA_1 can be accepted.

Further tests have been carried out to find out whether there was a significant difference in the scores for participant's satisfaction with adapting the tutorial to the VAK learning style. The data was gathered from the user feedback questionnaire (Question 4: I think I can learn better with adapting to my learning style (Visual – Auditory -Kinaesthetic). The results illustrated that participants are happy with (83.3%), **indicating that HA_1 can be accepted.**

The results in Table 7 reveal that in general LANA-I performed well as an Arabic CA based on the number of corrected and uncorrected responses. The results show from the total conversations there are 281 correct responses (about 89.8% from the total utterances) and 32 incorrect responses, when the algorithm failed to detect the students answer because of spelling mistakes (about 10.2% from the total utterances). The spelling mistakes affected the strength of similarity of the utterances, this cause the incorrect responses. Moreover, from the conversations that did reach the end of the tutorial, the majority (95.8%) of them reached the goal, meaning the user stayed within the tutorial and did not interrupt the conversation before the end of the tutorial. This provides evidence that the users did enjoy using LANA-I as a tutoring system; this is
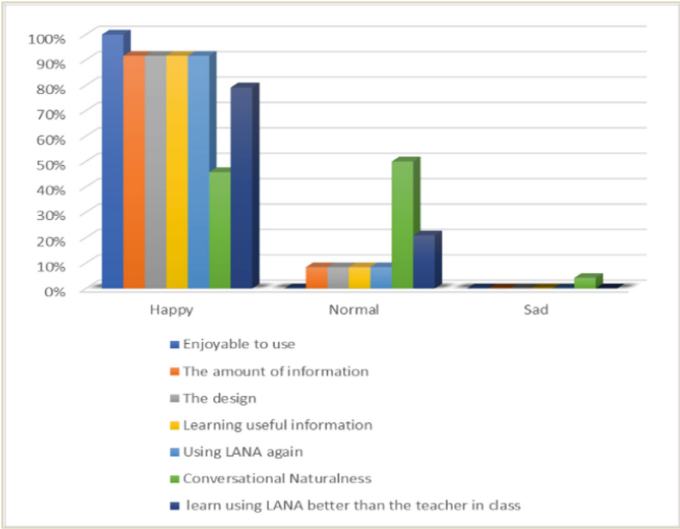

Fig. 9. Frequency analysis LANA-I questionnaire

also reflected in the opinion of the participants in the end user questionnaire who thought that LANA-I is enjoyable (100%) and a vast majority stated they would use the system again (91.6%).

Table 7. Data analysis from the log file

Conversation analysis	
METRIC	
Total number of conversations	24
Total number of utterances in all conversations	313
Average number of words per user utterance	3.1
Average number of utterances per conversation	31
Number of correct response from LANA	89.8%
Number of incorrect response from LANA	10.2%
Number of unrecognized utterances	38%
Number of utterances using Similarity Strength	57.18%
Number of rules fired	170
Average conversation duration (mins)	11.5
Percentage of conversations leading to goal	95.8%

An additional insight provided from these results is that the LANA-I engine is allowing the reduction of scripted patterns. The results illustrated that (57.18%) of all the utterances input by the users were actually different from the scripted patterns and in this case the system used the Short Text Similarity algorithm (Cosine algorithm). The log file shows that 34 unique utterances relating to 15 different rules were correctly recognised and dealt with by the LANA-I algorithm by firing the appropriate rule. Based on these findings Hypothesis HB_1 can be accepted. However, the results have identified some weaknesses in the LANA-I architecture, mainly the number of unrecognised utterances. LANA-I failed to recognise some utterances from the users (38%). Upon further analysis of the log file it was found that some of these unrecognised utterances were due to the gaps in the knowledge base, which can be easily addressed by expanding the patterns in the knowledge base. However, further research is needed to improve the similarity measure in order to reduce the number of patterns in the knowledge base.

5   Conclusion

This paper outlined a novel Arabic CITS called LANA-I, which used the VAK learning style model to enhance the learning of children. Our findings provide evidence for theses novel features:

1. LANA-I can be adapted to the VAK learning style for the tutorial.

2. LANA-I is able to converse in the Arabic language to teach children age (10-12) the science subject.
3. The LANA-I engine used PM with STS similarity algorithm was effective in recognising the user utterance.

The pilot study illustrated some key information with regards to the effectiveness of using the VAK learning style with the tutorial and with regards the functionality and robustness of LANA-I CA. To summarise the main findings of the pilot study are as follows:

- Using VAK learning style with LANA-I improves the participants (neurotypical) learning.
- LANA-I can be adapted to the user's learning style and enhance their learning.
- LANA-I engine managed to reduce the number of scripted patterns by an average of 11% using the STS algorithm comparing to PM approach.
- LANA-I engine managed to respond correctly to the user by an average of 89%
- LANA-I is able to lead the user towards the goal of the tutorial, with 95.8% success of completing the tutorial.

However, the first evaluation highlighted areas of weakness within LANA-I architecture. Further research is required to make components and algorithms within LANA-I more robust and to achieve the main objective, which is developing an Arabic CITS for people with Autism. Additional research is required as follows:

- Further improvement to the knowledge base and the CA engine will be made based on the results of the first pilot study.
- New methodologies will be researched and developed to overcome the spelling variations in the Arabic language, which affect the performance of the similarity algorithm.
- Further enhancement to the similarity algorithm to improve the matching of Arabic text much more efficiently in order to reduce the number of incorrect responses.

These weakness and further enhancements will be addressed by further research and development, to make the system ready for use with autistic pupils.

References

1. The National Autistic Society: Autism (2015). Available: http://www.autism.org.uk/about-autism/introduction/what-is-autism.aspx
2. Alotaibi, A.A.: Knowledge and Use of Applied Behavior Analysis Among Teachers of Students with Autism Spectrum Disorder in Saudi Arabia. Washington State University (2015)
3. Fletcher-Watson, S., Hammond, S., O'Hare, A.M., Petrou, M., Helen, Pain, H.: Click-East: evaluating the impact of an Ipad app on social communicative abilities in young children with autism. In: Presented at the 2013 International Meeting for Autism Research (2013)
4. Bernard-Opitz, V., Sriram, N., Nakhoda-Sapuan, S.: Enhancing social problem solving in children with autism and normal children through computer-assisted instruction. J. Autism Dev. Disord. **31**(4), 377 (2001)

5. Aljameel, S.S., O'Shea, J.D., Crockett, K.A., Latham, A., Kaleem, M.: Development of an Arabic conversational intelligent tutoring system for education of children with ASD. In: Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2017 IEEE International Conference, pp. 24–29. IEEE (2017)

6. Aljameel, S.S., O'Shea, J.D., Crockett, K.A., Latham, A.: Survey of string similarity approaches and the challenging faced by the Arabic language. In: Computer Engineering & Systems (ICCES), 2016 11th International Conference, pp. 241–247. IEEE (2016)

7. Ahuja, N.J., Sille, R.: A critical review of development of intelligent tutoring systems: Retrospect, present and prospect. Inter. J. Comput. Sci. Issues **10**(4), 39–48 (2013)

8. M. o. Education: Ministry of education (2017). Available: https://www.moe.gov.sa/en/Pages/default.aspx

9. Michie, D., Sammut, C.: Infochat scripter's manual. Convagent Ltd., Manchester (2001)

10. Qian, G., Sural, S., Gu, Y., Pramanik, S.: Similarity between Euclidean and cosine angle distance for nearest neighbor queries. In: Proceedings of the 2004 ACM Symposium on Applied Computing, pp. 1232–1237. ACM (2004)

11. Latham, A., Crockett, K., McLean, D., Edmonds, B.: A conversational intelligent tutoring system to automatically predict learning styles. Comput. Educ. **59**(1), 95–109 (2012)

12. O'Shea, J., Bandar, Z., Crockett, K., McLean, D.: A comparative study of two short text semantic similarity measures. In: KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications, pp. 172–181. Springer (2008)

13. Li, Y., McLean, D., Bandar, Z.A., O'Shea, J.D., Crockett, K.: Sentence similarity based on semantic nets and corpus statistics. IEEE Trans. knowl. Data Eng. **18**(8), 1138–1150 (2006)

14. Massaro, D.W., Bosseler, A.: Read my lips: the importance of the face in a computer-animated tutor for vocabulary learning by children with autism. Autism **10**(5), 495–510 (2006)

15. Smith, A.: Accelerated Learning in Practice. A&C Black, London (1998)

16. O'Shea, J., Bandar, Z., Crockett, K.: Systems Engineering and Conversational Agents, vol. 10, pp. 201–232 (2011)

17. Kelly, D., Tangney, B.: Adapting to intelligence profile in an adaptive educational system. Interact. Comput. **18**(3), 385–409 (2006)

18. Person, N.K.: AutoTutor improves deep learning of computer literacy: Is it the dialog or the talking head? In: Artificial Intelligence in Education: Shaping the Future of Learning Through Intelligent Technologies, vol. 97, p. 47 (2003)

19. Lee, J.-L., Orwig, G., Gunter, G., Witta, L.: The effect of cognitive styles on a visually-oriented task in online learning environments. In: Society for Information Technology & Teacher Education International Conference, pp. 3489–3494. Association for the Advancement of Computing in Education (AACE) (2006)

20. Cohen, J.: Statistical Power Analysis for the Behavioral Sciences, 2nd edn. Erlbaum Associates, Hillsdale (1988)

21. Latham, A., Crockett, K., McLean, D.: An adaptation algorithm for an intelligent natural language tutoring system. Comput. Educ. **71**, 97–110 (2014)