Using Games and Smart Devices to Enhance Learning Geography and Music History

Marta Filimon
Faculty of Computer Science, Alexandru Ioan Cuza University
Iasi, Romania *marta.filimon@info.uaic.ro*

Adrian Iftene
Faculty of Computer Science, Alexandru Ioan Cuza University
Iasi, Romania *adiftene@info.uaic.ro*

Diana Trandabăț
Faculty of Computer Science, Alexandru Ioan Cuza University
Iasi, Romania *dtrandabat@info.uaic.ro*

Abstract

This paper presents the implementation of an educational game as a skill for Amazon's software assistant Alexa. The main motivation behind this work comes from the fact that learning through games and smart devices is commonly better received by children. The application we discuss generates test questions by extracting and analyzing information from two major knowledge resources, DBpedia and Wikidata. The questions can be automatically adapted to the knowledge level of the player through Computer Adaptive Testing (CAT). Specifically designed for Alexa as a skill, with intents, slots and sample utterances, users can interact with our application through Alexa voice service or through any smartphone, allowing the game to be played from home, school or any other place with an Internet connection.

Keywords: eLearning, Computer Adaptive Testing (CAT), Knowledge extraction.

1. Introduction

Over the last years, smartphones and smart speakers (like Amazon Echo or Google mini) have become more present in our lives [7]. The number of devices integrating Alexa voice interface from Amazon increases every day, and their type began to diversify. Some services prove to be very useful in the kitchen [13], as proven by the fact that 85% of users have *set an alarm while cooking* using the smart speakers and 45% have *added an object to their shopping cart* [29]. In the automotive domain, Ford has integrated Alexa with the Ford Sync [17] application, which allows users to easily perform operations through voice commands while they drive.

Amazon has provided developers with a service through which they can integrate Alexa voice interface with various applications they develop. Alexa Skill Kit (ASK) is a collection of APIs that help developers quickly and easily create "skills" for Alexa [3, 4]. Skills are voice applications that a user can access through Alexa. A skill (or ability) consists of two parts: the *interface* and the *skill service*. In order to have a functional skill, the two parts must be coded to allow for perfect communication. There are currently around 60,000 skills among Amazon services available from companies like Starbucks, Uber and Capital One, as well as from innovative designers and developers, and their number continues to grow [24]. Part of these skills is used in education [20].

Additionally, it is well known that educational games become ever more frequent today, the majority of them related to general culture and designed to improve knowledge acquisition by a student [25].

In this context, this paper introduces a complex application, offering a novel solution to geography and music history learning. The proposed system creates test questions extracting data from well-known information sources such as DBpedia and Wikidata, constantly adapting tests at the level of the students using Computer Adaptive Testing (CAT). The whole system is using Alexa's interactive voice interface, which can be activated through an Amazon Echo device or from a smartphone.

In the next sections, a series of similar applications are presented, followed in section 3 by the architecture and implementation details for our general culture application. Section 4 discusses the creation of the database of knowledge for generating questions, section 5 presents usability tests and error analyses, while the final section draws several conclusions.

2. Similar Applications

This section presents existing educational applications available as skills for Alexa, in the form of chatbots or other types of voice applications.

2.1 Educational Games

A series of educational applications are available, implemented as skills for Alexa, as discussed in [16]. **The Question of the Day** application [3] proposes a new question every day, extracted from different domains, such as science, art or entertainment. The major difference with our system is the fact that the collection of questions and hints for answers is a closed collection, not dynamically adapted to the knowledge level of the user.

Another educational game is **AmazingWord Master Game** [4], an application introducing a chain game. Starting from a random word, the user has to name a word that starts with the last letter of the initial word, and so on. The difference between this game and our application is that here the user competes with the application; while in Bob each user competes with himself or his friends.

The Tricky Genie [5] is a different kind of application, emphasizing the comprehension of English. Thus, the game forces the user to make a choice between three possible solutions for a challenge presented as a story. The major drawback of this game is that it has a limited number of predefines stories, and the learning does not adapt to the level of English that the user needs to improve.

2.2 Chatbots

Lately, one of the most thrilling innovations in eLearning is the use of chatbot technology, commonly known as a conversation agent that can serve as a helping hand by simplifying teaching methodology [26]. Recent technological advances have opened new gates for innovative and effective solutions to meet the needs of students by developing applications that can serve as a personalized learning resource. Moreover, these automated applications can take up some of the repetitive tasks and can help instructors and teachers save on the time that each student should spend on individual training.

Chatbots work on the principle of interacting with users in a manner close to human interaction. These intelligent robots are often used as virtual assistants (Cortana, Siri, Google Assistant, Amazon Alexa, etc.) [19]. A suggestive example would be *Google Allo* – a smart messaging application, part of Google Assistant, which interacts with the user by sending text messages and answering questions. This application supports both voice and text queries. *Google Allo* is able to answer almost any question, just like when you do a typical search on Google.

In the modern age, chatbots prove to be the most innovative solution in combating the gap between technology and education. Engaging chatbots creates an interactive learning experience for the student - a method similar to the one-to-one method with the teacher. From testing the student's behavior to tracking progress, robots play a vital role in improving a student's abilities. Bots can also play a major role in encouraging a student to work by sending regular reminders and notifications.

2.3 Speech in eLearning

Spoken language is essential in teaching, and eLearning systems have tried to incorporate it in different ways: from assessing children in intercultural contexts [6], to offering spoken explanations to primary school mathematics lessons [1] or activating eLearning system components using speech recognition [14], [28]. A different research direction involved using voice recognition to detect the emotional states of children and to identify moments which can prove to induce stress [12]. The implementation of these solutions are financial expensive and most require that the child is physically present on the site. By difference, our solution can be used by anyone owning an Amazon Echo device or even a smartphone and can be played in any place where Internet access is available.

3. Architecture of our Application

The architecture of our system is depicted in Figure 1. Through our application, users can:

- *answer questions* about general culture in the field of geography or music history;
- *initiate general culture tests* that will end with a score which enables them to advance in the overall player ranking;
- *find the place in the overall ranking* and the number of points they have accumulated;
- *post on Twitter* the accumulated score.

The first step in designing the application was to acquire the user's spoken input through a voice service. Subsequently, templates for the interaction between user and the system have been established, specifying intents, slot types and utterances. The AWS Lambda service was then used to develop Alexa's skill and interactions. The set of questions and answers are stored in a database populated with data extracted from two common information sources. The remaining of the section details the modules used, along with their interactions.

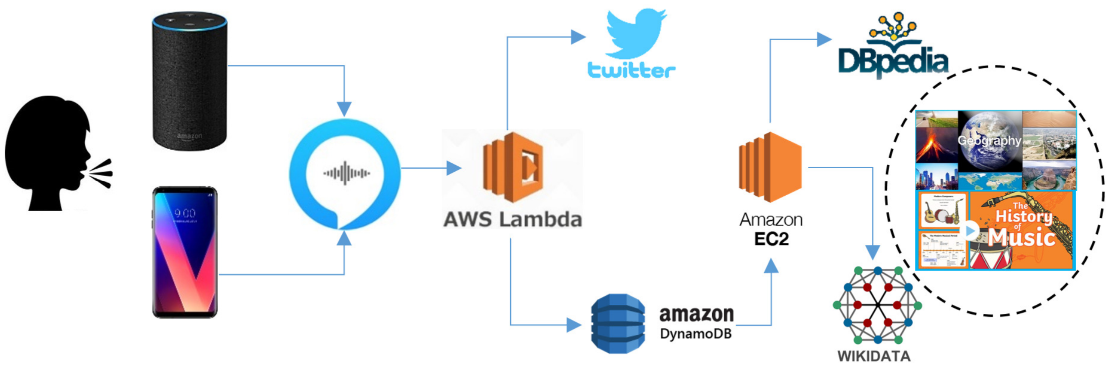

Fig. 1. The Interaction between the Services used to develop the application.

The first step in setting up a skill for Alexa was the activation of an account through the developer's section of Amazon (https://developer.amazon.com). After receiving a unique identifier, various information are collected about the interaction model, integration with services (in our case, the voice service), description, test instructions, etc. In order to set up the interaction model, templates for intents, slot types and utterances have been created.

A JSON object contains a list of commands that Alexa will be able to respond to. The intents are stored with an optional list of slots which characterize the intent, along with their type. Our application accepts the following intents: (1) `AMAZON.CancelIntent` – collects a series of phrases the user can utter to *stop* the game; (2) `BobBeginTest` – includes sentences which *start* a new game; (3) `ComposerTraining` – includes phrases the user can utter when he/she wants to *start a new general culture test* about a *composer* or *operas*; (4) `BobAnswer` - contains input phrases to *answer a general culture question*.

The type of a slot is defined by listing all its possible values. These are phrases containing possible answers, such as:

- `AMAZON.GB_CITY` – names of *cities* that are correct answers for the questions related to *cities*, in the form which is most commonly used in *Great Britain*;
- `AMAZON.Landform` – names of landforms which are correct answers to questions related to *lakes*. In a further stage of the application, other types of *landforms* will be added (*mountains*, *hills*, *valleys,* etc.);
- `AMAZON.Date` – format for a *date* which can be a correct answer for the creation of an *opera*, for instance;
- `OperasNames` – names of *operas* which appear as correct answers to questions related to this type (*Marriage of Figaro, Symphony number five*, etc.).
- `Composers` – names of *composers* which appear as correct answers to questions related to this type (*Amadeus Wolfgang Mozart, Johann Sebastian Bach*, etc.).

A collection of sample utterances can be added for the interaction model and are used to specify each possible input that comes from the user and precede them with the intent it relates to.

The AWS Lambda service was used as it provides support for Alexa skills and facilitates calls to other AWS services through lambda functions (https://aws.amazon.com/lambda/). The application is written in NodeJS, as it is recommended for small server-side applications where non-blocking operations are very important. For creating the lambda function, a blueprint of the running environment is needed, along with a template for the function under development. The trigger function was also configured, as the voice interaction through Alexa Skill Kit. After creating the function, an ID is provided which ensures that data from the user is sent to the precise lambda function. In order to have a functional skill, the code needs to be deployed, which we performed through the AWS Command Line Interface (CLI).

4. Question Pool

Current research discovered that complex test questions, as opposed to simple, factoid questions, are better suited for assessing reading comprehension [8], [18]. Most of the existing approaches used to generate questions are focused on generating questions from a single sentence [11], [21, 22] and only a few approaches use semantic analysis of text [2], [9] or semantic resources to create test question.

4.1 Question Generation

In the previous section, we presented the application which uses a database of information to generate questions. In order to populate the database, two popular knowledge sources, DBpedia [10] and Wikidata [29, 31] were used. Data from Wikipedia was previously used in similar approaches [23]. Access to the two knowledge bases was made through a separate application designed to extract and filter data to only store the information that the game needs.

DBpedia or Wikidata can be accessed through SPARQL endpoint. Initially, the Skill Service component handles requests made to the knowledge bases. Although at first glance communication seems fast, with the increase in the number of extracted instances and with the creation of operations similar to those of relational databases, the latency has increased considerably, reaching more than a second. The effect of increasing latency could easily be noticed by a user of the Bob application. The interaction between Alexa and the user no longer gives the impression of a natural conversation. To reduce latency, it was necessary to store the information needed to construct a question into a database. We have chosen DynamoDB because it is a NoSQL database. To further reduce network latency, we intend to use in a further development stage the cross location replication of the data principle.

4.2 Collecting Data

The data for question generation is extracted jointly from Wikidata and DBpedia. The first step deals with selecting a topic for the test questions, from the user or from a collection of concepts. Once an instance is selected, a query is made through the SPARQL endpoint provided by *Wikidata*. Depending on the amount and type of data returned in the previous step, if more information is needed, a second *query* will be made to the SPARQL endpoint provided by *DBpedia*. The use of both knowledge bases was imposed because, as the comparative study between DBpedia, Freebase, OpenCyc, Wikidata and YAGO [15] suggests, DBpedia tends to lack consistently structured data.

Ambiguities may arise when combining the two source data since they have different methods of identifying instances. In DBpedia, the *name* attribute is used, but sometimes it differs from the *name* attribute in Wikidata, which also provides a *name of the instance* tag. To mitigate the effects of inconsistencies, other predicates that specify the name in other forms, such as `dbo:longName` were considered. Then a query is build that matches the *name* given by Wikidata to the one provided by DBpedia.

For each answer, several similar response candidates are generated to ensure it is not trivial for the player to identify the correct answer. For example, if the correct answer should be the *Caspian Sea Lake*, located in *Asia*, for another candidate answer we will not consider lakes located in other continents. For example, for questions related to *cities*, *locations* from the same *country* as the correct answer are selected, along with *cities* randomly selected, to ensure that we have at least four possible variants to choose from.

In the case of the answer for *Cities* in the same *country* with the correct answer will be selected according to the following criterion: *that they should be among those with a large population, i.e. in the top 12 cities of the population*. In this step, one of the difficulties encountered was the fact that not all *cities* had as object the `P31` (*instance of*) predicate, `Q515` (*city ID*). After analyzing the objects for the `P31` predicate, we noticed that the instances which are cities are related to the `Q15284` instance (*the municipality ID*), meaning that:

1. There is an object for which the value of the `P31` predicate has the subject of the predicate `P279` (*subclass of*) equal to `Q15284`, which in natural language means that: *the selected subject is instance of a subclass for the concept of a municipality*.
2. The selected topic is an instance of a subclass that is, in turn, the subclass for the concept of a *municipality*.

4.3 Manipulating Intents

The Bob application can receive requests from users to ask questions about *cities*, *countries*, *lakes*, *operas*, *composers* or a *random* category. These are translated by the Voice Service into `BobAskCity`, `BobAskCountry`, `BobAskLake`, `BobAskOpera`, `BobAskComposer`, and `BobRandomQuestion` respectively. We exemplify below how the `BobAskComposer` intent is manipulated, the procedure is similar for all intents.

When the `BobAskComposer` intent is received by the Service Skill component, the `handleComposerIntent` function that is part of the composer module will be called. Following this call, the user will be asked a question about a *composer* for which Bob finds information in the database. The `handleComposerIntent` function forms a configuration object containing: (1) the name of the table containing information about the category to which the question belongs (in this case composers), (2) the object that corresponds to the function that builds the question. Finally, it calls the `ask_answer.handleAsk` function, which, regardless of the category of the question, will randomly choose a record from the database of composers. The `ask_config.question` function will then create the question. Before providing the user with the question, the correct answer will be added to the session, along with other information, so that they can be later retrieved.

The request received from Voice Service contains the request key, which is a JSON object containing information about the last interaction of the client with the Alexa voice interface. Within this object, a summary of the request is encoded, specifying the type of intent and the value of the slots, if any. To evaluate the user's response, the answer is extracted from the JSON response object and compared to the session data representing the correct answer.

If there is a match, the answer is considered correct and the user is notified and asked if he wants to find out more about the answer, if more information is available in the database. The user will answer this question with *Yes* or *No*, which will further generate an `AMAZON.YesIntent`, respectively `AMAZON.NoIntent`. These two intents can be generated at any time. For example, a user can say yes in response to a question or to an operation he asks Alexa. In order to distinguish between the moments when the user is asking for more information from the database, the *MoreInfo* field was added to the session, taking the true value when it is legal for the next intent to be `AMAZON.YesIntent` or `AMAZON.NoIntent`.

4.4 Testing

The application can receive requests from users to start a new general culture test. This is translated by the Voice Service component into a `BobBeginTest` intent. The first question in a test is randomly chosen from the database. To mark the fact that the user finds itself into a test, the test key containing an object is added to the session. This, in turn, has two key-value elements representing the number of remaining questions and the number of questions that the user has correctly answered. The only difference between the intent of `BobRandomQuestion` and `BobBeginTest` is the session component.

The answer is transmitted to the device that is integrated with Alexa via a callback that receives as a parameter an object representing the session and an object containing the answer that Alexa will provide to the user along with other information. In order not to duplicate the code, we chose to send the `BobRandomQuestion` intent function, a fake callback that adds the test-specific items to the session, and then calls the real callback, which transmits the response to the user. The answer to the questions in the test is similar to the answers to a question. The only differences are the changes that need to be made to the elements in the session, which are specific to a test, and the fact that one question can be attempted to respond only once.

4.5 Choosing the Questions and Answers for a Test

After completing a test, each player will receive 4 or 10 points depending on the level of geography or music history he/she possesses. We will consider that there are two levels: beginner and master, beginners receiving 4 points at the end of the game, and masters 10 respectively. To objectively determine the level of a player, we used an algorithm that dynamically adapts the questions during a test (a process called Computer Adaptive Testing or CAT) [30]. The objective of better classifying a player's level of knowledge will be based on answers already given to questions, a priori probability of each question, and a priori likelihood of classifying the population into levels of knowledge. In order to determine the question which best classifies the player we used the following relationships:

- $P(m_k)$ the probability that a player will have masterliness m_k, in our case m_k can be *beginner* or *master*;
- $P(z_i|m_k)$ the probability that the answer will be z_i, which can be right or wrong, given the masterliness m_k;
- Let be Z the response vector a player gives in a test, $P(m_k|Z) = \frac{P(Z|m_k)P(m_k)}{P(Z)} = \frac{P(Z|m_k)P(m_k)}{\sum_{n=1}^{K} P(Z|m_n)P(m_n)}$ is the probability that a player has the level m_k given that he answered questions with Z, and K is the number of masterliness that in our case is equal with 2;
- Suppose answers to questions are independent, which implies that $P(Z|m_k) = \prod_{i=1}^{N} P(z_i|m_k)$, where N represents the number of questions, and m_k is a level of masterliness;
- $P(z_i = v) = \sum_{j=1}^{K} P(z_i = v|m_j)P(m_j)$ the probability that answer at question i to be v which in this case may be *correct* or *wrong*;
- Probability of responding v to a question i the player has masterliness m_i, $P(m_i) = \frac{P(z_i=v|m_i)P(m_i)}{\sum_{j=1}^{K} P(z_i=v|m_j)P(m_j)}$;
- Entropy expected after answering the question $i, H(S_i) = P(z_i = 1)H(S_i|z_i = 1) + P(z_i = 0)H(S_i|z_i = 0)$.

In order to choose the most appropriate question, 3 randomly extracted questions in each category will be selected from the ones with the maximum of information gain.

The Bob application can get from users the request to post their score on Twitter [27]. This will be translated by the Voice Service component into a `BobPost` intent. In order to implement this functionality, the skill had to be configured first in order to link to the user's Twitter account. For this, the application needs access to the Twitter user's `accessToken` through which it can access, modify, or add information to the user's account. When a `BobPost` type intent is received by the application, the score of a user, together with his location, will be extracted from the database. A user's ID can be extracted from the JSON object that the Voice Service component forms in order to handle customer requests. Once we have obtained the user data through the Node.js package, Twitter will post its score and place. To do this, we had to integrate an endpoint that implements the OAuth 2.0 protocol, for which we have defined an authorization URL in the Alexa Skill Developer Portal in the Configuration section.

5. Evaluation

We performed usability tests and collected end-users opinions about our application (both from groups and from individuals), to see what can be improved or changed in the future in the application.

5.1 Usability Testing

In order to evaluate the application, we invited ten participants, 4 girls and 6 boys (from 12 years old to 20 years old, from primary classes to high school classes) and organized different playing sessions. We have recorded their experience with the game while they were performing various tasks. During the tests, we took into account their social interaction activity, having 5 participants with high activity, 3 with moderate and 2 with reduced.

Methodology: The conducted usability test consisted of an introduction (where we present the Bob application), tasks description (where we ask participants to use as many options from Bob app as they can), a short interview and a post-test questionnaire. We instructed the participants to think out loud and express their thoughts during the test. After the task series that we communicated verbally to the participants, we gathered their assessment of the overall experience using the QUIS (The Questionnaire for User Interaction Satisfaction) scale. The tasks that users performed covered the main options of the applications and each session took around 8-10 minutes. In some cases, users have executed the steps without having seen them executed in advance by someone, just as there were cases where they performed the tasks after someone else went previously through the steps, presenting them.

Individual opinion

During all experiments, before showing our applications to groups, we asked some of the participants to use Bob application. We have recorded their experience with the application while they were performing the usual tasks.

Participants: We collaborated for evaluation with four participants. Their selection was random, the group is formed of 2 girls and 2 boys. 1 of them didn't have previous experience with voice applications. They received the application and their interactions with Bob were assessed for the duration of a day.

Results: From our observation during the test sessions, the individuals had the best experience while performing the tasks after someone else. They found quickly the application options and they were able to use them appropriately. The execution of tasks wasn't very fast, and because in the test room was quiet they have not been blocked very often by the application. The participants were asked to rate their experience with a note from 1 to 9, where 1 stands for a confusing/frustrating experience and 9 for a clear/pleasant experience. The user responses to the post-test questionnaire show that the "overall feeling experience" is most appreciated by participants (8.5), followed by "selection commands", "questionnaires", "post to Twitter" (all with scores around 8).

Group opinion

After our initial interaction with individuals, we presented our applications to mixed groups with or without members which participate in the first part with individual evaluation. Again, we have recorded their experience with the Bob application while they were performing the same tasks as individuals.

Participants: We collaborated for evaluation with ten participants, from which four participated in the individual evaluation. Their selection was random, the group is formed of 4 girls and 6 boys. All have previous experience with games from tablets or smartphones, and only 3 of them didn't have previous experience with voice applications. They received Bob application and their interactions with the application were during one day (after the day with individual evaluation).

Results: From our observation during the test sessions, the participants had the best experience while performing the tasks after previously observing someone else doing the tasks. If at the beginning they follow very strictly the steps of the tasks, after the second or third round of questions they performed the tasks with the goal to obtain the fastest results (in order to receive more points). They found quickly the application options and they were able to use them properly. In many cases they talk too fast and they need to repeat the commands or answers in order to be understood by Bob. Similar to individual experiments, the participants were asked to rate their experience with a score from 1 to 9, where 1 stands for confusing/frustrating experience, and 9 for clear/pleasant experience. Again, the user responses to the post-test questionnaire show that the "overall feeling experience" is most appreciated by participants (around 7.5) followed by rest of the options which are between 6 and 7.

Groups vs. individuals' evaluation

The group evaluation requires children to communicate, either verbal or non-verbal, i.e. through gestures. When the participants interact, they are able to finish the level faster than one individual, because they discuss and find faster the requested answer.

Amazon Echo vs. Smartphone Evaluation
The tests presented above were performed with both the Amazon Echo and smartphones. If in the first case the experiments were carried out inside the rooms, in the second case tests were done inside and outside. The participants reported that it is much more convenient to use the smartphone that can be used very easily outside, but they also signaled that outside noise dramatically decreases the power of recognition of commands given within the Bob application. When using the application outside, participants had to speak louder than usual, often to ask questions again and there were situations when they did not understand what the Bob application was saying.

Remarks
When comparing to individuals, groups generally didn't follow very strictly the steps of the tasks, and they tried to find shortcuts to finish the assigned tasks sooner. Also, they blocked more often the application, mainly because they talk a lot and Bob does not understand what he has to do. If, in the beginning, students from primary school were impressed by applications, after explanations they were very happy to use the Bob application even they didn't obtain very good results. For they the CAT propose usually the beginner level. Also, in group evaluation where is involved communication, they were very verbose and very happy to solve together tasks.

Recordings analysis clearly showed that:

- The Bob application was very attractive and was appreciated positive both by individuals and by groups;
- Even in the few cases the voice interaction was a bit confusing, after explanations and after given examples, things became more clear;
- The device presents minor performance and stability issues, especially when they work in groups and when they are very verbose;
- The social interaction between participants motivated them to learn the presented notions and try the tests again;
- All participants agree that the lessons can be more attractive with Bob application and it can help them to learn the new content of the lessons;
- Difficulties had been encountered because the questions were not targeting the native country of the participants;
- Also, evaluations based on games reduce the stress of the children and can provide a fast way for professors to see which the level of an entire class is.

5.2 Error Analysis

Most problems occurred related to understanding and evaluating answers to questions that Bob is addressing, which affects the test side. These components of the system are most error-prone because the user is in the position to say complicated words that may not be correctly understood by Bob, especially if the user in a non-native speaker. Unwanted behavior occurs because the user response is not clear enough and fails to be associated with an existing intent, or the associated intent is not expected. These problems are based on the fact that the voice service component, when a user speaks a word-by-word, will compare them with the items in the replica list that the system accepts and identifies the one that seems closest to the one uttered by the user. Problems arise when the user did not pronounce correctly certain words or spoke too quickly, or if his interaction with Alexa was interfered with by other sounds or an unexpected event of the user (for example, coughing or sneeze).

If the user interaction with Bob fails to be assigned to any intent, it will be ignored. Naturally, the user will repeatedly say the most likely answer, which may give the impression that the application has been blocked. If the interaction was assigned to another intent, which is unlikely, then the application will give an unexpected response. For example, if during a test the correct answer sounds very similar to "Post on Twitter", the application will tell the user that the score has been posted on Twitter. One of the consequences of this behavior is the interruption of the test. At this time, the problems with processing the input from the user cannot be solved by the Bob application because this component is provided by Alexa. In the future, in order to improve the understanding of the user's pronunciation, we intend to make a learning module in which Bob learns minimal information about certain geographic elements, and his task is to repeat the name of that element.

The participants' complaints were related to the fact that the speed of speaking and speaker volume could not be configured during a test. Another issue is related to the flexibility to choose the domain of the questions during a test. Although they knew they could only use geography and the history of the music, they wanted to receive information from other fields not related to the two (mathematics, literature, etc.). Also, the participants with a high level of knowledge asked for the possibility to choose, at the beginning of a test, the level of difficulty, in order to avoid the simple questions in the first part of the test, while the system is guessing the user's level.

6. Conclusions

The purpose of this paper is to present a general culture game, tailored for the geography and the music domains, enhanced with voice interaction, as well as to document each step, in order to facilitate other possible skill development. With the integration of Alexa Voice Interface into more and smarter devices, including from automotive or home appliances domains, our application can be used as entertainment when we wait in traffic or while we cook.

Over the next period, the Amazon site will abound in skills developed for Alexa, and we tend to believe that many of them will be dedicated to education. Our main contributions in this paper come from the fact that:

- we are proposing a solution to create an ability to help users learn geography or music history with;
- a new method of obtaining evaluation questions based on the extraction of information from DBpedia and Wikidata resources and;
- a new method of adapting these tests to user level based on CAT (Computer Adaptive Testing).

To improve the application, new categories for the questions, apart from the one already included of geography or music history domains, will be added. Statistics can be used to predict the users' educational path, as well as the tendency to answer correctly or wrongly specific sets of questions, depending on their knowledge. More levels of difficulty can also be included, as well as other social networks to post the score. An important advantage over similar approaches comes from the fact that our solution does not require specialized equipment, can be installed locally and used anywhere through Amazon Echo or through a smartphone.

Future work will focus on two main directions. The first is to use the game in the learning and assessment processes of students who learn geography or music history. The second is to expand the knowledge database to related fields such as history, biology, literature, astronomy as well as other more distant fields such as art, sports, cinema, music, etc. Regardless of the domain that is going to be added, the programmer should do the following steps:

- Decide the predicates of the Wikidata records that are going to take part in a question;
- Find the relation between the IDs of the records in Wikidata and the one used in DBpedia in order to have a higher probability that important information is not missing;
- In the skill service part, the logic of forming the question for the new domain should be added, and the configuration file which handles the test creation should be modified accordingly.

Acknowledgements

This work was supported by a grant of the Romanian Ministry of Research and Innovation, PCCDI – UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0818 / 73PCCDI, within PNCDI III. It was also partially supported by POC-A1-A1.2.3-G-2015 program, as part of the PrivateSky project (P_40_371/13/01.09.2016).

References

1. Ahmad, A. R., Halawani, S., Boucetta, S.: Using Speech Recognition in Learning Primary School Mathematics via Explain, Instruct and Facilitate Techniques. In Journal of Software Engineering and Applications, 07(04), pp. 233-255 (2014)
2. Aldabe, I., Lopez, d. L. M., Maritxalar, M., Martinez, E., Uria, L.: ArikIturri: An Automatic Question Generator Based on Corpora and NLP Techniques. In Proceedings of Intelligent Tutoring Systems, 8th International Conference, ITS 2006, Jhongli, Taiwan, June 26-30, pp. 584-594 (2006)
3. Alexa Skills Store. Question of the Day (2019), https://www.alexaskillstore.com/Question-of-the-Day/B01N6QUAXX. Accessed March 17, 2019
4. Alexa Store. Amazon Word Master Game (2019), https://www.alexaskillstore.com/Amazing-Word-Master-Game/B017OBSCOS. Accessed March 17, 2019
5. Alexa Store. Tricky Genie (2019), https://www.alexaskillstore.com/Tricky-Genie/B01IKR3OAC. Accessed March 17, 2019
6. Álvarez, C.: Dialogue in the classroom: the ideal method for values education in multicultural contexts. In 6th International Conference on Intercultural Education "Education and Health: From a transcultural perspective". Procedia - Social and Behavioral Sciences 132, pp. 336–342. DOI: 10.1016/j.sbspro.2014.04.319 (2014)
7. Amazon. Echo & Alexa (2019), https://www.amazon.com/Amazon-Echo-Bluetooth-Speaker-with-WiFi-Alexa/dp/B00X4WHP5E. Accessed March 17, 2019
8. Amruta, U., Ashwini, G.: A Survey on Automatic Question Paper Generation System. International Advanced Research Journal in Science, Engineering and Technology (IARJSET). National Conference on Innovative Applications and Research in Computer Science and Engineering (NCIARCSE-2017), 4(4), January 2017, 18-20. DOI: 10.17148/IARJSET/NCIARCSE.2017.06 (2017)
9. Araki, J., Rajagopal, D., Sankaranarayanan, S., Holm, S., Yamakawa, Y., Mitamura T.: Generating Questions and Multiple-Choice Answers using Semantic Analysis of Texts. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, December 11-17, pp. 1125-1136 (2016)
10. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In Aberer K. et al. (eds) The Semantic Web. Lecture Notes in Computer Science, 4825. Springer, Berlin, Heidelberg, pp. 722-735 (2007)
11. Becker, L., Basu, S., Vanderwende, L.: Mind the gap: Learning to choose gaps for question generation. In Proceedings of NAACL-HLT 2012, pp. 742–751 (2012)
12. Chen, K., Yue, G., Yu, R., Shen, Y., Zhu, A.: Research on Speech Emotion Recognition System in E-Learning. In Conference Computational Science - ICCS 2007, 7th International Conference, Beijing, China, May 27-30, Part III. (2007)
13. Cnet. 11 ways to use Alexa in the kitchen (2019), https://www.cnet.com/ how-to/amazon-echo-ways-to-use-alexa-in-the-kitchen/. Accessed March 17, 2019
14. Eichner, M., Göcks, M., Hoffmann, R., Wolff, M.: Speech enabled services in a web-based e-Learning environment. In Proceedings of the IASTED International Conference, WEB-based education. February 16-18, Innsbruck Austria (2004)
15. Farber, M., Ell, B., Menne, C., Rettinge, A.: A Comparative Survey of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. In Semantic Web 1, pp. 1-26 (2015)
16. Filimon, M., Iftene, A., Trandabăț, D.: Bob - A General Culture Game with Voice Interaction, In Proceedings of 23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Budapest, Hungary (2019)
17. Ford. Sync (2019), https://www.ford.com/technology/sync/. Accessed March 17, 2019

18. Hamaker, C.: The effect of adjunct questions on prose learning. Review of Educational Research, 56(2), pp. 212–242 (1986)
19. Heater, B.: Comparing Alexa, Google Assistant, Cortana and Siri smart speakers. TechCrunch. 8 October 2017. https://techcrunch.com/2017/10/08/comparing-alexa-google-assistant-cortana-and-siri-smart-speakers/ accesed in March 2019, (2017)
20. Lewis, V. 6 Ways Amazon Alexa can help with Homework. Perkins: School for the blind eLearning (2018), http://www.perkinselearning.org/technology/blog/6-ways-amazon-alexa-can-help-homework. Accessed March 17, 2019
21. Lindberg, D., Popowich, F., Nesbit, J., Winne, P.: Generating natural language questions to support learning on-line. In Proceedings of the 14th European Workshop on Natural Language Generation, pp. 105–114 (2013)
22. Mazidi, K., Nielsen, R.: Linguistic considerations in automatic question generation. In Proceedings of ACL 2014, pp. 321–326 (2014)
23. Palmero, A. A.: Extending Linked Open Data resources exploiting Wikipedia as source of information. PhD Thesis. Dipartimento di Informatica. Scuola di Dottorato in Informatica – XXV ciclo. Universita degli Studi di Milano (2014)
24. Perez, S.: The number of Alexa skills in the U.S. more than doubled in 2018 (2019), https://techcrunch.com/2019/01/02/the-number-of-alexa-skills-in-the-u-s-more-than-doubled-in-2018/. Accessed March 17, 2019
25. Shapiro, J.: 9 Facts about Video Games and the People who play them (From Pew's New Study). Forbes (2015), https://www.forbes.com/sites/jordanshapiro/2015/12/18/ 9-facts-about-video-games-and-the-people-who-play-them-from-pews-new-study/ #5bd739f63e35. Accessed March 17, 2019
26. Spilka, D.: 4 Ways For Using Chatbots For eLearning. eLearning Industry. August 18, 2017. https://elearningindustry.com/chatbots-for-elearning-4-ways-using accesed in March 17 2019, (2017)
27. Twitter, https://twitter.com/twitter. Accessed March 17, 2019
28. Ţucă, L. P., Iftene, A.: Speech recognition in education. Voice Geometry Painter Application, In the 9th Conference on Speech Technology and Human-Computer Dialogue, IEEE, July 6-9, 2017, Bucharest, Romania, pp. 1-8 (2017)
29. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM 57, pp. 78-85 (2014)
30. Weiss, J. D., Kingsbury, G. G.: Application of Computerized Adaptive Testing to Educational Problems. Journal of Educational Measurement, 21, pp. 361–375 (1984)
31. Wikidata, https://www.wikidata.org/wiki/Wikidata:Main_Page. Accessed March 17, 2019