Speech Applications in the eWALL Project

Mihai Dogariu, Horia Cucu[*], Andi Buzo, Dragoş Burileanu, Octavian Fratu
Speech and Dialogue Research Laboratory
University Politehnica of Bucharest
Bucharest, Romania
[*]Corresponding author (e-mail: horia.cucu@upb.ro)

***Abstract*—This paper presents how technological progress may come to help smart homes development, namely smart caring-homes. It proposes two different software applications that have social, medical and psychological implications in the lives of people in need of special care as it is often the case for elders or people with physical impairments. These applications focus on processing the speech signal from inside the smart caring-homes and offering various metadata that can be used by care takers to assess a person's health state. The first application tries to count the number of people inside a room based solely on the detected speech signal, while the second application is represented by a cognitive game meant to stimulate the player's memory with a personalized test.**

***Keywords*—smart caring-home; distant speech signal processing; speaker counting; memory test**

I. Introduction

With high-end technology increasing in popularity in our day-to-day lives it has become affordable for people in well-developed countries to resort to modern devices to aid them in household routines and even more. In particular, smart caring-homes offer support under various forms, ranging from being able to tell to the user when it is necessary to perform more exercise, based on the recorded physical activity, to helping the user relax and improve his memory state with the help of certain games. Researchers focused on human needs and bringing comfort closer to people has always been an idea worth investing in, with immediate results.

These applications serve well to people who have been born with physical disabilities, have been left with them as an outcome of accidents or to those whose physical state has degraded over time. Taking into account the effect that aging has on the human body it can be clearly stated that people will encounter a time in life where their body will not be at its prime, having their abilities to see, hear, feel, talk or move affected by the natural aging process. Therefore, it appears that senior citizens would greatly benefit from having at their disposal a system which not only helps them carry out daily tasks with ease but is also part of their life. Such is the system which the eWALL project aims to develop, integrating the electronics in a wall with the underlying infrastructure which is meant to facilitate the communication with all sectors of the health care system so that seniors can feel more comfortable in their own homes [1]. It also addresses cognitive disorders that may come with aging, such as mild dementia, failing memory or poor sense of orientation, all of which would otherwise require special medical assistance.

In the near future it is desired to acquaintance elders with technology and introduce them with similar approaches as the eWALL, in such a manner that it would make their lives easier and help them become more independent, knowing that they do not have to rely on others' assistance to perform tasks of low complexity. The current progress in smart caring-homes allows elders to lead a more comfortable life but medical assistance from a care taker is still needed as the person cannot completely rely on the system to perform every possible task. Perhaps the future will solve some of the problems that caring-homes of this kind still have or offer better solutions than the ones that are already present in these smart environments. Currently, researchers are focusing on responding the most basic needs that elders have, responding to not only physical ones, but also emotional, social and safety related ones as well. It is the current trend to provide the user with information about the weather, monitor the levels of harmful gases in the case he unconsciously left the oven on, his health state with lungs and heart being the point to focus on, how much exercise he has made, how much exercise he still has to make in order to keep a healthy routine, how his social life and cognitive state are evolving by keeping track of the number of people he interacts with and how he is performing in some games that stimulate both memory and mental state. When developing a platform which will interact with elders it is important to make it easy to understand, transparent when discussing about monitoring systems and fit to the individual needs of every user. Therefore, it is important to make the system able to adapt to each user as it is harder to have different people, with different technological backgrounds, learn how a smart environment behaves and how to respond to it, in return. Knowing the target audience is important when taking into account the complexity and the attitude of the language that will be used when communicating with the user, preferring simple sentences and a happy mood, as psychological studies revealed over the years. It is also important that the means in which the smart caring-home acts have to be non-invasive, otherwise making elders reluctant to using it and rendering the system useless.

This paper describes the development of two software applications that can be implemented in a smart caring-home, the first one aimed at assessing the number of visitors in a room by making use of the audio signal recorded in that room and the second one represented by a cognitive game meant to stimulate one's memory, which implies the user's voice interaction with the system. This way, a non-intrusive approach is used to offer relevant information to medical personnel about the elder's social activity and memory state.

978-1-4673-7560-3/15/$31.00 ©2015 IEEE

These two applications will be presented thoroughly in the next sections, each at a time. The reason this system is feasible is that there is an increase in the aging population in the well developed countries which can afford implementing such environment and people are willing to spend money on having their retirement years in a comfortable and independent manner. This certainly offers a boost of confidence and an increase in the quality of life for those in need and also for their families, who can rest assured that their loved ones are being taken care of.

II. Visitors Monitoring Application

A. Diarization system

The first proposed software application that this paper covers and can be part of a smart caring home environment is the Visitors Monitoring Application, an application written in Java, which is responsible for counting the number of persons in a room based solely on the audio signal captured by a microphone inside that room. What this application does is that it continuously records the audio signal from the room and then performs speaker diarization. Speaker diarization is the process of segmenting an audio recording into speaker-homogenous segments [2]. This answers to the question "Who spoke when?", labeling each section of an audio clip containing speech to a certain person. As the identity of the speakers is not known in advance, speaker diarization gives arbitrary IDs to the labels. However, the clustering algorithm assigns the same speaker ID to the speech segments that belong to the same speaker.

The diarization used in this application is based on the platform made available by LIUM [3]. It consists of several steps (see Fig. 1) which include:

1. BIC segmentation. In this step, the speech is segmented based on the BIC (Bayesian Information Criterion) algorithm which takes into account acoustic similarity.

2. BIC clustering. The consecutive intervals are merged if their similarity after BIC exceeds a threshold. Therefore, the speech signal is less granularly segmented.

3. Viterbi re-segmentation. With the segments from Step 2 Gaussian Mixture Models (GMMs) are trained and a Viterbi decoding is performed. The underlying reason for this second segmentation is that GMMs with Viterbi decoding provide a better segmenting and can merge segments which are acoustically similar even though they survived the BIC clustering.

4. Speech/non-speech filtering. With some pre-trained GMM models of speech and non-speech (speech, noise, music, speech+noise, etc.) another decoding is performed and a set of new segments is obtained. The information from this set of segments is combined with the one from Step 3 in order to eliminate the non-speech segments. Even though there might be some expectation about applying this step at the very beginning, the reason for applying it only now is that the segments resulted after the speech/non-speech decoding are very fragmentary. This could lead some speech segments to be left out.

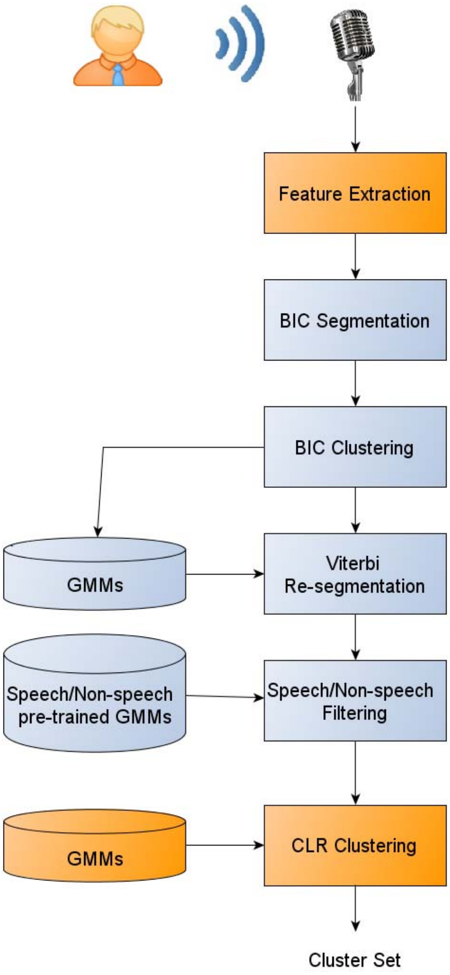


Fig. 1. The block diagram of LIUM diarization system

5. CLR clustering. Cross Likelihood Ratio (CLR) clustering is used as a final step to merge the segments that may belong to the same speaker. The cross entropy metric is used as similarity score in order to determine the segments to be merged.

A complete description of the adapted implementation of the LIUM diarization in our framework is given in [4]. The number of speakers present in the recorded speech signal is then deduced by the number of the distinct clusters of segments. Pre-trained models of known speakers can be used in this flow and can be inserted at step 3 and 5. This will help the analysis of the speakers present in the recording environment. The importance of the accuracy of the estimation is relative to the application. For the targeted application it is not important the exact number of speakers, because the application is not meant to tutor the monitored person, but rather to have a rough understanding on the lifestyle he is carrying.

Therefore, this process can be applied in a smart caring-home to have a better overview of how the social lives of elders are evolving. They can be one of the persons labeled as frequent speakers, being sure that their voice will be encountered in most audio recordings. Family members and close friends can also be labeled as known speakers as they are also expected to be recorded on a frequent basis. Leading to a thorough knowledge about who and when visited the caring-home, important information can be gathered about the elder's social activity, need for communication and even happiness.

As a better quality of life for their owners is the ultimate goal of caring-homes, it is useful to know when that person has not been visited for a long time, has not seen certain family members in a long time or is alone in case of an emergency. For emergency situations it is desired to have more than one system capable of alerting specialized personnel that can quickly respond to alerts. However, in all other situations the Visitors Monitoring Application can provide important metadata that can be interpreted by different specialists and by combining it with information from other systems can lead to relevant information, improving the lives of those in need of caring, which was the motivational drive to develop this application.

B. System configuration

The Visitors Monitoring Application has been developed starting from four basic concepts: system configuration, data acquisition, data processing and processed data storing in a remote database, as described by the flow in Fig. 2. Each of these stages was further developed and brought improvements to the point where they are today, with added functionalities that offer a better manipulation, scalability and customization.

Configuration of the Visitors Monitoring Application is done by modifying a configuration file, under the form of a properties file, as displayed in Fig. 3. Knowing that individual environments can differ from one home to another it is important to have a certain flexibility assured so that the system can be applied under different circumstances. Therefore, the following attributes can be edited: the identifier of the microphone on which the recording will be done, the name that will be assigned to the specific microphone, the audio recording length, in seconds, the remote database's IP and port where the results will be uploaded and whether to store the recorded audio clips on the hard-drive or delete them after they have been processed.

C. Data acquisition

The application retrieves the necessary diarization resources, i.e. Gaussian matrices, mixture models and the universal background model, from some predefined folders, meaning that the user must not alter the folders structure. However, these models can be changed by overwriting the existing files with any new files of the same type. After all resources have been read, the application proceeds into searching for all the available microphones and adding them to a list. Then, an audio format according to which recording will take place is created. This format has been set to 16 kHz sampling rate, 16 bits per sample coding, one channel, with the big endian convention, as this is the most popular format that is used in speech signal processing.

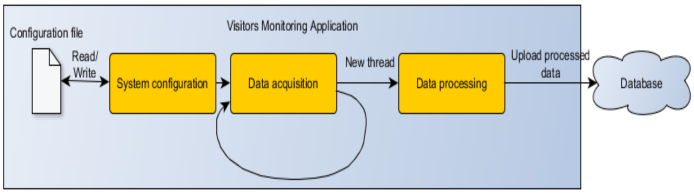
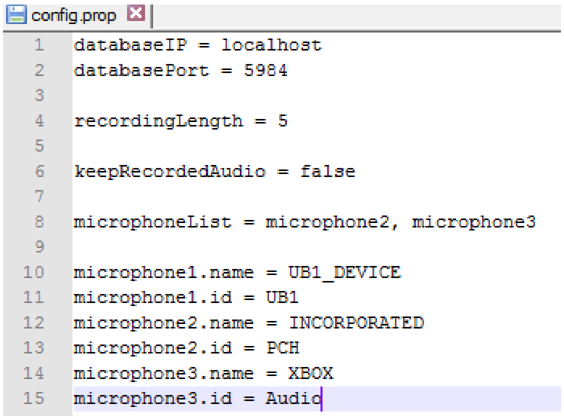


Fig. 2. Visitors Monitoring Application Architecture



```
databaseIP = localhost
databasePort = 5984

recordingLength = 5

keepRecordedAudio = false

microphoneList = microphone2, microphone3

microphone1.name = UB1_DEVICE
microphone1.id = UB1
microphone2.name = INCORPORATED
microphone2.id = PCH
microphone3.name = XBOX
microphone3.id = Audio
```

Fig. 3. Visitors Monitoring Application configuration file example

The next step is to compare the list of available microphones with the microphone list from the configuration file in search for a match. If any match is found then for each of them a recording thread is started, according to the audio format. In other words, the application starts the recording with the specifications from the configuration file on all available matching microphones. After a time frame equal to the "recordingLength" variable has passed, 2 actions take place. The first action is that a new thread is started, on which the recorded audio clip is temporarily saved on the hard-drive and the control is being taken over by the data processing part. The second action is that the system starts a new recording from where the first one left off, thus assuring a continuous recording of the audio signal present in the room, under the same configuration. Basically, it records the audio signal from the available microphones in chunks of length equal to the value of the "recordingLength" variable, expressed in seconds.

D. Data processing

After the audio clip has been temporarily saved on the hard-drive, a diarization process starts for each new recording. The software tool that was used to perform the diarization is the open source Java application developed by the Laboratoire d'Informatique de l'Université du Maine (LIUM - EA 4023), which is available under the GNU General Public License. This has been used under the form of a jar file, included in the main project. During the diarization process there are several steps that the software takes, namely performing segmentation, silence detection, hierarchical clustering, and Viterbi decoding using GMM models trained with EM or MAP, more information being available at the project's official page [5]. Each of these steps outputs an intermediary file which will be used by the next step and so on, reaching the final stage, as it can be seen in Fig. 4. The present fields represent the following:

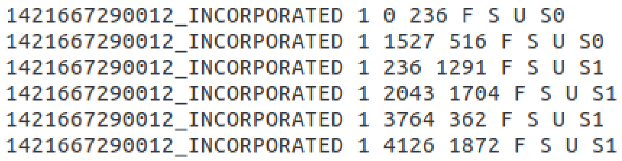
- field 1: 1421667290012_INCORPORATED= the show name
- field 2: 1 the channel number
- field 3: 0 the start of the segment (in features)
- field 4: 236 the length of the segment (in features)
- field 5: F the speaker gender (U=unknown, F=female, M=Male)
- field 6: S the type of band (T=telephone, S=studio)
- field 7: U the type of environment (music, speech only, …)
- field 8: S0 the speaker label

A more human-readable form is also available, to have an easier understanding of the time frames, as it is displayed in Fig. 5. After diarization process ends, depending on the value of the "keepRecordedAudio", the recorded audio clip will be kept on the hard-drive or deleted. This option has been added to conform to privacy and security regulations as it may be unwanted for someone to have everything that was said in his own house stored in some database. Therefore, only metadata will be kept in the database, but that option still exists to be able to run additional tests on the recorded audio without having them automatically deleted. This is the main stage of the application from the speech diarization point of view, with the leading and trailing stages serving an auxiliary part.

E. *Processed data storing*

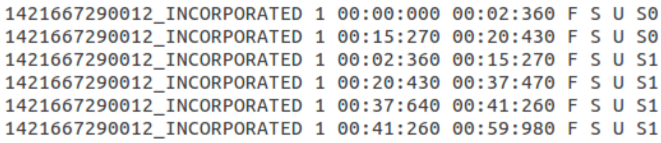
After the diarization output file has been obtained the application continues with storing the most important information about each recording in a database, namely the time when the recording started, the recording's duration, the microphone's name and the number of different speakers that were detected in the room. All this information is stored under the template presented in Fig. 6. For each recorded audio clip there is a corresponding entry with the associated metadata stored inside the CouchDB database, a non-relational database to which the eWALL project consortium agreed upon. The database's IP and port are taken from the configuration file presented in an aforementioned section.

The Visitor Monitoring Application's performances depend on the distance between the speakers and the microphone, the loudness with which one speaks and voice distinctiveness, as there are cases in which not even human hearing can differentiate between two voices. Another problem that this system encounters is when there are at least 2 people talking at the same time, thus making a mixture of voices. This has been treated with a post-processing stage in which very short interventions, i.e. less than 4 seconds, are being ruled out of the speaker counting algorithm. This algorithm has been based on the fact that in a normal conversation each participant would have chunks of speech longer than 4 seconds. Observations made on this outcome suggest that the time frame can be extended even further maintaining good results.



```
1421667290012_INCORPORATED 1 0 236 F S U S0
1421667290012_INCORPORATED 1 1527 516 F S U S0
1421667290012_INCORPORATED 1 236 1291 F S U S1
1421667290012_INCORPORATED 1 2043 1704 F S U S1
1421667290012_INCORPORATED 1 3764 362 F S U S1
1421667290012_INCORPORATED 1 4126 1872 F S U S1
```

Fig. 4. Diarization output file sample



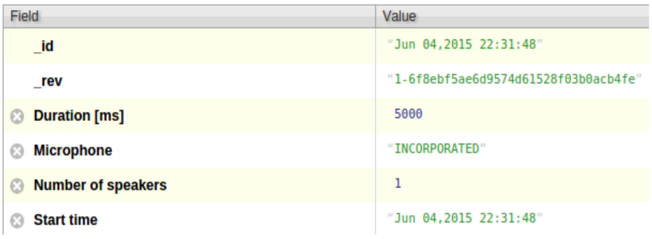
```
1421667290012_INCORPORATED 1 00:00:000 00:02:360 F S U S0
1421667290012_INCORPORATED 1 00:15:270 00:20:430 F S U S0
1421667290012_INCORPORATED 1 00:02:360 00:15:270 F S U S1
1421667290012_INCORPORATED 1 00:20:430 00:37:470 F S U S1
1421667290012_INCORPORATED 1 00:37:640 00:41:260 F S U S1
1421667290012_INCORPORATED 1 00:41:260 00:59:980 F S U S1
```

Fig. 5. Refined diarization output file

III. Memory Quiz Application

The second proposed software application that can be part of a smart caring-home environment is the Memory Quiz Application, an application also written in Java, which is a cognitive game, meant to stimulate memory under the form of a quiz test, using only voice interaction with the system. The voice interaction part is done with the help of an automatic speech recognition (ASR) engine which tries to understand the answer provided by the one who plays the game and interpret it as an answer to a question. Having only a finite set of answers for each question it comes naturally to approach an ASR system based on Finite State Grammars (FSGs), which fits perfectly on the developed architecture, namely choosing one answer out of four possible ones.

All the questions from the test are customizable, meaning that a care taker or family member can create the questions specifically for the elder in question, thus responding to their individual needs in what memory is concerned. They can be questions with a more personal approach, regarding family and friends, or with a more general thematic such as historic or international events, depending on what the operator of the system thinks it is best. Customizability is important because cognitive disorders affect people in different ways, meaning that the memories that are damaged are different from one another, therefore that calls for exercise on different themes to help them improve their condition or at least keep it stable. A certain effort is demanded when creating the proper questions for the test in order to make the output information as accurate as possible but the process of creating the questions itself is intuitive even for non-technical persons. The entire application has been designed to be user-friendly so to offer a high degree of satisfaction with a low degree of complexity in handling it.



Field	Value
_id	"Jun 04,2015 22:31:48"
_rev	"1-6f8ebf5ae6d9574d61528f03b0acb4fe"
⊗ Duration [ms]	5000
⊗ Microphone	"INCORPORATED"
⊗ Number of speakers	1
⊗ Start time	"Jun 04,2015 22:31:48"

Fig. 6. Database entry sample

The Memory Quiz Application's architecture includes two more auxiliary applications, namely an application that creates the questions and uploads them in the remote database and an application that populates the remote database with some sample questions, made for demo purposes. All of them will be detailed in the following sections. The entire process can be considered to consist of two parts, namely adding questions to the database and running the memory test as it can be seen in the application's architecture, presented in Fig. 7.

A. *Adding questions applications*

There are 2 available options for introducing questions in the remote database, questions that will further be used in the memory test. The first consists of introducing individual questions in the database by means of a GUI that allows maximum customization for them. These questions have several specific attributes: the language in which they are written, the topic which they cover, the question itself, 4 different answers, one correct answer out of the 4 and the option to add a picture for each answer, displayable at runtime.The language box is a drop-down list from which the user can choose one of the available languages. At the moment, the supported languages are Dutch, English, French, German, Romanian, Russian and Spanish with the possibility to add even more. After the language has been selected the user proceeds into naming a topic, one that he feels it is best suited for the question he has in mind. The question field will be completed with the question's text body which should preferably be neither ambiguous nor too complex. Just below this field there is an option to add an image for the question, supposed the image would bring more value than the question itself. However, this functionality has not been implemented yet, so for the moment it will just store in the database the image corresponding to the question. The answers to the questions can be either the form of an image or a text field or even both, depending mainly on the aspect of the question. Out of the 4 possible answers the user must select one as the correct answer, by checking the corresponding radio button.

After all attributes have been set, the user clicks the 'Upload' button and the application will verify that the proposed topic and answers are part of the selected language's dictionary. This is mandatory for the voice interaction with the system to take place without causing any errors. If any of the aforementioned words is not part of the selected language's dictionary then it will not be recognized by the automatic speech recognition engine. Once every field is verified the application uploads the question in the remote database, under the form presented in Fig. 8. As the ASR engine works with a FSG when it comes to recognizing the available topics for any language, it is needed to update these FSGs at every new question upload because new topics may appear and they have to be correctly recognized by the ASR system. This updating process is a feature of the application in charge of uploading the questions, taking place dynamically and automatically for every new uploaded question.

Another feature of the application is that it can upload a series of questions at once, in order to populate the remote database for demo purposes.

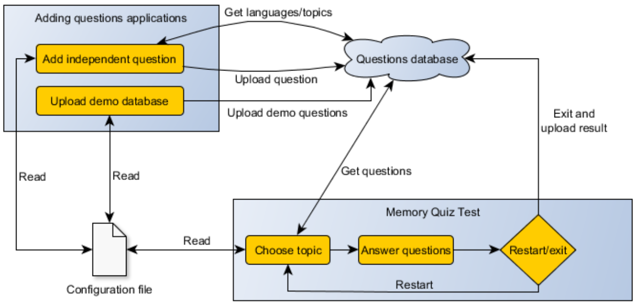
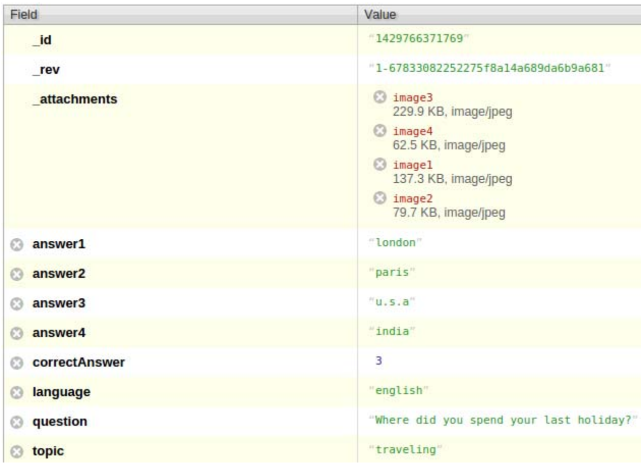


Fig. 7. Memory Quiz Application Architecture



Field	Value
_id	"1429766371769"
_rev	"1-67833082252275f8a14a689da6b9a681"
_attachments	⊗ image3 229.9 KB, image/jpeg ⊗ image4 62.5 KB, image/jpeg ⊗ image1 137.3 KB, image/jpeg ⊗ image2 79.7 KB, image/jpeg
⊗ answer1	"london"
⊗ answer2	"paris"
⊗ answer3	"u.s.a"
⊗ answer4	"india"
⊗ correctAnswer	3
⊗ language	"english"
⊗ question	"Where did you spend your last holiday?"
⊗ topic	"traveling"

Fig. 8. Question sample entry as seen in the database

In the current demonstrator of the application, a set of 18 questions can be uploaded at once, organized as follows: 9 questions in English and 9 in Romanian, for each language there are 3 available topics (i.e. family, friends, nature) and for each topic there are 3 distinct questions. The motivation behind this feature is that when somebody wants to perform a demo or a test on the application he has to first add some questions to the database, which might be time consuming, and only after that perform the required tests. All the attributes concerning the questions are stored in an .xlsx file and the pictures that are to be uploaded in a separate folder. There is no need for further folder management or customization as this serves only the demo purpose, meaning a fast and easy solution for populating database with some questions.

B. *Memory Quiz Test*

The main part of the Memory Quiz Application is running the test and answering the questions that were created using the previous two applications. As a general concept, the user is supposed to give the answer he thinks it is correct by uttering its number, then the system will process the audio signal, evaluate the given answer, give a feedback regarding the correctness of the answer and move on to the next question, if there is another one available. At the end of the test the user is prompted with the overall result and this result is stored in a remote database, which is named by appending the "_results" suffix at the end of the name of the database in which the questions have been stored. The application follows the architecture presented in Fig. 7 and is described, step by step, in the following.

The first step in running the quiz is editing the configuration file according to own preferences, customizing the following parameters, as observed from Fig. 9.

- databaseIP, databasePort, databaseName – the database's IP, port number and name, respectively. These parameters are common for all the available applications in the Memory Quiz package.

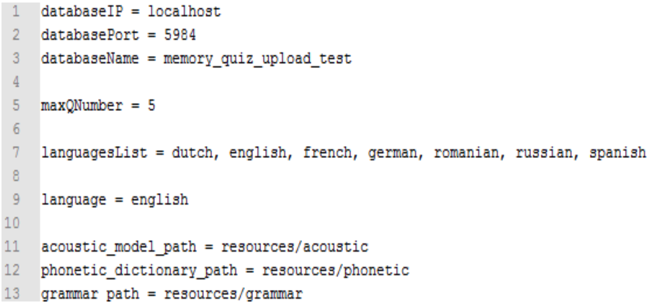
- maxQNumber – the maximum number of questions to display for any topic, which is capped at this parameter's value.

- languagesList – the entire list of languages available for this application. It is common for both the quiz and the independent questions adding applications. Currently, the list is limited to Dutch, English, French, German, Romanian, Russian and Spanish, due to restrictions imposed by the speech recognition engine, but with the possibility to extend it to any language, given that the necessary resources are available.

- language – the selected language for the questions that will be downloaded from the database; can be any of the ones enumerated above.

- acoustic_model_path, phonetic_dictionary_path, grammar_path – the paths to speech recognition resource folders

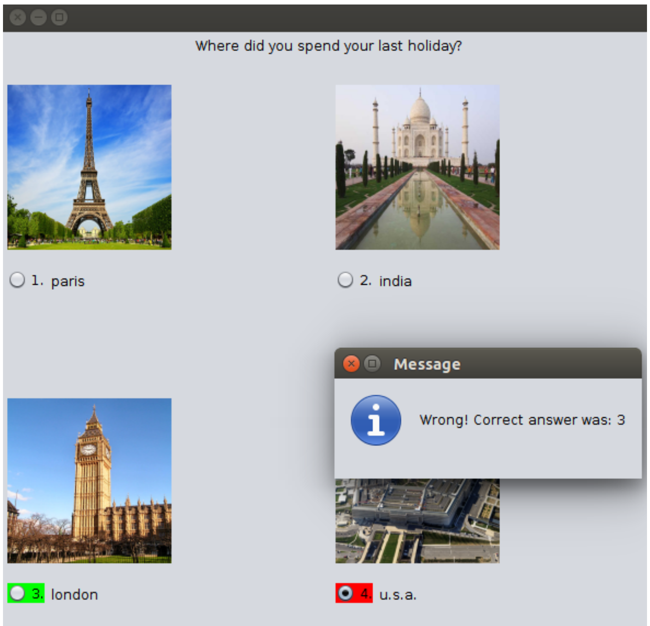
The application will search if a Kinect device is plugged in and available in order to set it as the recording microphone, given the fact that this should be the real-case scenario in which it will run. If no Kinect is available then the default microphone will be selected for the recording process. After that, the application will run on two parallel threads: one thread responsible for the GUI interaction and one Swing Worker thread responsible for controlling the application's logic. The Worker thread will start a speech recognition process, based on the FSG created with the topics that have been checked to be valid in the selected language. If the detected answer is not amongst the words that define the topics then the user will be asked to repeat his answer until a valid topic is selected. Once he does that, the topic recognizer stops and having both the language and the topic selected it proceeds to download all the questions that fit the selected criteria, shuffle them, select only the first '*maxQNumber*' questions and start another recognizer, this time aimed at recognizing the answers to the selected questions. After each recorded valid answer, the GUI thread will turn to green the correct answer and to red the wrong answer, if that is the case, as it can be seen in Fig. 10. At the end of the test, when all questions have been asked and answered, alike, a pop-up will be displayed with the score of the test, after which the score will be uploaded in the results database and the user will be asked whether he wants to replay the game or exit the program.



```
databaseIP = localhost
databasePort = 5984
databaseName = memory_quiz_upload_test

maxQNumber = 5

languagesList = dutch, english, french, german, romanian, russian, spanish

language = english

acoustic_model_path = resources/acoustic
phonetic_dictionary_path = resources/phonetic
grammar path = resources/grammar
```

Fig. 9. Memory Quiz Application configuration file example

As it can be seen in the above figures and explanations, the interaction between the computer and the user is done by sight but, more importantly, by voice. The entire application revolves around offering elders an interactive game that they can play at any time and which will also stimulate their memory. The ASR engine behind the application is the Carnegie Mellon University's (CMU) Sphinx open source software, available for free on their official website (http://cmusphinx.sourceforge.net/), an application also written in Java. They also provide the phonetic dictionaries, acoustic and language models for all the languages available in the application except Romanian, which has been developed internally by the speech and dialogue (SpeeD) research group, at the Faculty of Electronics, Telecommunications and Information Technology from Bucharest. The FSGs used by the answers recognition engine are static at the moment, involving only simple, immutable options that allow one to choose from the numbers '1', '2', '3' and '4' each of them being assigned to its corresponding answer option. This has been preferred to letters (i.e. 'a', 'b', 'c' and 'd') because of the higher spoken distinctiveness between them, thus lower confusion rate. In order to fit every available scenario the FSG has been translated and adapted to all 7 different languages. For the moment it does not offer the option to modify the answers recognizer's FSG dynamically, according to the input values from the application in charge of creating the questions, but it is in plan for the future developments. The main setback regarding this issue is that, at the moment, Sphinx cannot



Where did you spend your last holiday?

○ 1. paris

○ 2. india

○ 3. london

● 4. u.s.a.

Message

Wrong! Correct answer was: 3

Fig. 10. Question during quiz example

change between FSGs on the fly. Instead, it is required to start a completely new recognizer meaning that it must reallocate all the necessary resources which, in turn, leads to unpleasant delays in running the application. Due to this, linking images to the questions and only text to the answers is also disabled as it must make use of the same mechanism that is not yet feasible. It is also worth mentioning that the application is standalone and it only requires access to the Internet and to a remote database in which both the questions and the results of the tests will be stored. From the results database important information can be retrieved, such as in which period the elder performed better on the tests or which are the topics that he is not so good at, thus indicating on what aspects the care takers should focus their attention on.

IV. Conclusions

Smart caring-homes is an industry that has begun developing in the last decade, offering the possibility to implement many applications, aimed at helping the elders. It is a fact that one of technology's greatest goals is to help people lead a more comfortable life, bringing an independent way of living closer to reality for people who have been deprived of this right. The two applications presented in this paper respond to the need of making the care taking process more human independent and transparent to the user. Moreover, bringing human-machine interaction inside the house while keeping a non-invasive approach may be appealing to elders in need of communicating with others. The proposed systems are not meant to replace human supervision but they are supposed to help this effort by offering relevant metadata for more thorough studies. These applications come as proofs of concept, showing that it is feasible to include speech-driven software in smart caring-homes and, at the same time, provide medical information about the person living inside the house. There are still some drawbacks present in the implementation and they are mostly related to adverse environment conditions such as noise, overlapping speech, distance between the microphone and the speaker, speaker orientation towards the microphone, echo, reverberation etc. All these are topics of which speech processing researchers are aware and are continuously investigated in hope of finding the best solution, but it is certain that there is still room for improvement, meaning that the applications will benefit from these developments in the near future. As the smart homes industry is growing it is expected for the interest for speech applications in smart environments to gain more attention, as well.

Acknowledgment

This work has been partially funded by European Commission through the FP7 IP project no. 610658/2013 ”eWALL for Active Long Living - eWALL”, partially by UEFISCDI Romania under grant no. 262EU „eWALL” support project and partially by the PN II Programme "Partnerships in priority areas" of MEN - UEFISCDI, through project no. 32/2014.

References

[1] The eWall Project, http://ewallproject.eu

[2] D. Reynolds, P. Torres-Carrasquillo, “Approaches and applications of audio diarization,” Proc. of ICASSP, Philadelphia, PA (2005) 953-956.

[3] M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin, S. Meignier, “An Open-source State-of-the-art Toolbox for Broadcast News Diarization,“ Interspeech, Lyon (France), 25-29 Aug. 2013.

[4] A. Buzo, H. Cucu, L. Petrică, D. Burileanu, “An Automatic Speech Recognition Solution with Speaker Identification Support“, Proc. of the 10th International Conference on Communications (COMM), Bucharest, 2014, pp. 119-122.

[5] www-lium.univ-lemans.fr/diarization/doku.php/programmes last accesed on 14.07.2015.