Design and Evaluation of a Conversational Agent for the Touristic Domain

Andreea I. Niculescu, Kheng Hui Yeo, Luis F. D'Haro, Seokhwan Kim, Ridong Jiang, Rafael E. Banchs
Department of Human Language Technology, Institute for Infocomm Research, Singapore 138632
E-mail: {andreea-n, yeokh, luisdhe, kims, rjiang, rembanchs}@i2r.a-star.edu.sg

***Abstract*— This paper focuses on the design and evaluation of SARA, a conversational agent for the touristic domain featuring a high number of different, unique characteristics: spoken dialogue interaction, dialogue orchestration, context dependent information, an animated avatar and support for different kind of dialogue types, i.e. chat, specific and general question-answering, task oriented dialogues. The agent has currently two implementations: as web client and as mobile phone application. The paper describes the modules and resources required for running the agent on both interfaces, as well as the evaluation results obtained from two assessment studies concerning the interaction design of these two agent interfaces. The feedback gathered from the studies will enable us to improve the applications in terms of service, performance and usability.**

I. INTRODUCTION

Nowadays there is an increasing interest of companies from different economic sectors on using conversational agents. The reasons behind it rely on the capabilities these agents have to maintain users engaged in conversations for a prolonged time.

In the artificial intelligence field, a conversational agent refers to a system that allows natural dialogue interactions with humans through one or more modalities, e.g. text, speech, visual content, graphics, gestures, etc. Conversational agents are able to provide facial expressions, emotions, turn-tacking, visualization of structured content, emphasis, orientation, contextualization, personalization, etc., enriching the user experience with the system. Another important feature of conversational agents is their capability of switching between different types of verbal interactions: for example, within the same interaction session an agent is able to execute command, answer questions or simply chat with a user. The use of animated avatars (or embodied agents) gives the agent the possibility to respond more "humanly" by performing movements that reflect attention, task processing or error handling. Finally, conversational agents are equipped with flexible input/output modalities: depending on environment conditions (e.g. noisy or low illuminated place, etc.), amount or type of the information to be provided (e.g. using charts instead of tables, summarizing, playing videos, etc.), user´s profile and preferences the agent can use different input/output modalities to communicate with the user.

In the literature there are several examples of conversational agents used for a wide range of applications and domains, such as health-care [1], weather forecast [2], travel organization [3], entertainment [4], tutoring [5] etc. Probably one of the most popular applications is the Apple´s conversational agent Siri [6]. Siri is able to provide information for multiple tasks and domains including making appointments, sending text messages, performing phone calls, querying the stock market, getting weather information, making reservations or searching on the web.

Our application focuses on the touristic domain of Singapore. The reason for choosing this domain is that tourism is currently considered one of the fastest growing industries in the world, generating an estimated eleven percent of the total global domestic product [7]. This proves to be true also in our geographical context: according to the Singapore Tourism Board the number of tourists coming to the country is yearly increasing [8]. In 2013, the total number of tourists hit a new record: 15.5 million people came to visit the island which represents the highest tourist rate in the past decade.

The increasing amount of visitors requires additional resources in terms of accommodation, board, transportation and touristic guidance and information. Traditionally, tourists rely on using static information, such as book guides, printed maps, and informative flyer material to locate points of interest. This way of getting to know new places is considered to be useful, although the source of information might be outdated. With the prevalent usage of smartphones and tables the ITC industry has started to look for technological solutions for travelers who want to orient faster in new environments. As such, several tools were released on the marked offering interactive maps, automatic reservations and check-ins, personalized recommendations, travel guide assistants, etc. The advantage of these tools as compared with more traditional options relies on the reduced production and distribution costs, easy access to continuously updated information, as well as additional embedded services providing geo-positional and contextual information, social media sharing, in-site merchandising etc.

In this context, SARA was built as a response to the growing demand for personal touristic assistance and offers a comfortable solution for those who want to explore the city by themselves and have no touristic guide around. Currently, there are several other local touristic applications available at Google Play Android application store, such as *Singapore City Guide, Your Singapore Guide*, *Singapore Guide, Your Singapore Navigation*, etc., that provide information for tourists. However, these applications have a lower degree of interactivity in the sense that they support neither speech input/output modalities nor question & answering (QA) style

of interaction as SARA does, presenting the information as any other descriptive internet web page.

On the other hand, most of the applications concerning touristic guidance offer services in terms of either navigation or exploration. Applications meant for navigation include *Siri*, *Google Maps navigation*, *Sygic*, etc. while applications offering exploration services, i.e. presenting descriptive information for the user to read include *TripAdvisor, Wikihood*, *Triposo*, *Your Singapore Guide*, etc. Certainly these applications provide highly valuable information and help. However, from a user point of view it makes more sense to have navigation and exploration options available at the same time, i.e. without having to switch between applications [9]. Therefore, there is a need to create tools that can handle both of these categories.

Additionally, we believe that users need an application that does not distract them from walking on the street or looking around when sightseeing. Here is to mention that most of the current applications offering navigation and exploration services use visual information displayed on the mobile phone screen. To address this problem many natural language systems were developed. Examples include pedestrian navigation systems [9][10][11][12][13][14][15], city guidance systems [16][17] and QA systems [18].

Our system SARA complements the work mentioned above bringing navigation and exploration in a single application that combines visual information with speech to create a more natural way of interaction. Currently, we are working towards multi-lingual support which represents an additional feature that distinguishes SARA from other similar applications.

This paper is organized as follows: section II presents the system design and architecture; section III describes the agent interfaces, i.e. the web client and the mobile phone app; section IV shows the results of two evaluation studies carried out to assess the interfaces. Finally, the paper ends with conclusion and future work in section V.

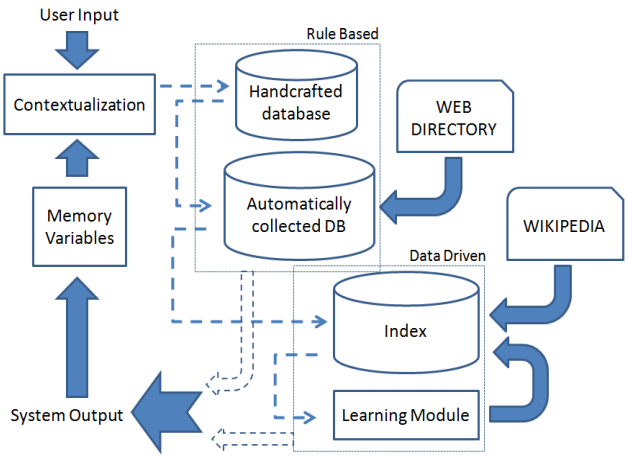

Fig. 1. Overall system architecture.

II. System Design and Architecture

The overall system architecture of SARA is depicted in figure 1. The system consists of three main subsystems operating in cascade: a **rule-based** subsystem for accessing information from two different databases, a **data-driven** subsystem for accessing information from an index and a **learning** module.

A. Databases

The rule-based subsystem parses the pre-processed user input to identify possible references to specific entities that are contained in the databases. Once the entities are resolved, a rule-based natural language understanding (NLU) module resolves the information request intention, which is represented in the form of a SQL query. The query is afterwards submitted to the databases in a predefined order of precedence.

The first database is a manually created resource that contains information about the most relevant tourism spots in Singapore. A total of 75 locations of interest are represented in this database. It includes museums, historical buildings, temples, and some of the most popular touristic attractions.

The second database was automatically collected by crawling some available web directories about Singapore. This database contains more than 8000 entries and it includes locations of interest, such as shopping centers, hotels, restaurants, transportation hubs, stores, etc. This database is accessed only when the first database fails to serve the information request. In case the second database also fails to provide an answer, the user input is passed to data-driven subsystem.

The data-driven subsystem is implemented by an example-based dialogue management [19] and deals with relatively general-purpose queries. This subsystem is equipped with an index consisting of multiples question-answer pairs. Each index pair is matched with a given input query to compute the contextual similarity between them. If an index pair has a higher similarity score compared to other examples, its answer part is considered as first candidate for the system response.

For implementing an example-based dialogue manager in our conversational agent we used Lucene[1], a special library for developing search engines. Lucene provides several useful features for storing dialogue examples, preprocessing utterances, computing similarities and retrieving the results by interpreting various types of queries. Each document instance in the index has the following three fields: a question, its answer, and the source of this information. All these field values are preprocessed with Lucene's built-in Standard Analyzer for tokenizing, lowercasing and removing stop-words. During the execution phase, at each turn the system prepares a query for matching the user's input utterance to a question and answer (QA) pair of every example in the index. Then, a set of example candidates can be retrieved by Lucene. To obtain a more accurate confidence score for each candidate we added a re-scoring process based on a linear combination of the cosine similarities to the QA pair of each

[1] http://lucene.apache.org/

candidate. Finally, the values for answer and source fields of the example having the highest score are chosen as the result.

The success of this example-based approach depends on the quantity and quality of the index examples. Since building these resources manually from scratch is time consuming, labor intensive and expensive, we propose to use knowledge extracted from Wikipedia. This knowledge can be retrieved without significant effort, i.e. using semi-automatic methods.

Wikipedia contains a large number of articles consisting of unstructured texts and structured info-boxes. Each of these info-boxes is a set of tuples summarizing key attributes described in the corresponding article – see figure 2. The QA pair examples were generated based on this structured information.

Fig. 2. An example of info-box attribute to be matched with the templates for generating QA pairs

A total of 3155 articles related to Singapore were collected from Wikipedia database dump as of February 2013. Among them, 1308 articles include info-boxes. From this dataset, we sorted out the 50 most frequently occurred attributes and prepared a set of templates for generating QA pairs corresponding to each attribute type.

TAB 1 TEMPLATES FOR QA PAIRS GENERATION ABOUT 'BIRTH_DATE' ATTRIBUTE

When was [ENTRY] born?
[ENTRY] was born [PREP] [VAL].
What is [ENTRY]'s date of birth?
[ENTRY]'s date of birth is [PREP] [VAL].
What is [ENTRY]'s birthday?
[ENTRY]'s birthday is [PREP] [VAL].
When is [ENTRY]'s birthday?
[ENTRY]'s birthday is [PREP] [VAL].

Table 1 shows the QA templates prepared for the '*birth_date*' attribute. In these templates, '[ENTRY]', '[VAL]' and '[PREP]' represent the main title of a given article, the value for '*birth_date*' attribute in info-box, and the proper preposition for the value in sentence generation, respectively.

After preparing the templates for every selected attribute type, the QA pairs were generated by matching the corresponding attributes and their values in the info-boxes over the Wikipedia collection.

For example, we can retrieve the birthday of Lee Hsien Loong - an entry name in Wikipedia- who was born on '10th February 1952' from the '*birth_date*' attribute of the info-box – see figure 2. This information can be matched with the templates in table 1 and then the following four QA pairs can be generated:

Q: When was Lee Hsien Loong born?
A: Lee Hsien Loong was born on 10th February 1952.
Q: What is Lee Hsien Loong's date of birth?
A Lee Hsien Loong's date of birth is 10th February 1952.
Q: What is Lee Hsien Loong's birthday?
A: Lee Hsien Loong's birthday is 10th February 1952.
Q: When is Lee Hsien Loong's birthday?
A: Lee Hsien Loong's birthday is 10th February 1952.

Finally, 7164 QA pairs were generated by matching 1308 info-boxes with 112 templates. These QA were used to build a Lucene index for the example-based dialogue management of SARA.

The third subsystem provides the conversational agent with learning capabilities. It implements a simple sub-dialogue in which the system informs the user it doesn't know the answer and subsequently asks the user to provide information on that particular topic. If the user provides a response the new information is stored in the index after being manually validated by a system operator. In this way, the system 'learns' from the user about a new topic.

In addition to the three main subsystems described above, there is another contextualization module, which stores the user-system interaction information state by means of a memory variables stack. These variables include the recent topics and entities mentioned by the user, such as locations, organizations, persons, etc. The variables are used for grounding and contextualizing each new user input augmenting the overall system information about the interaction. In this way, the system is able to resolve incomplete user inputs such as "how old is *she*?", "how can I get *there*?", "tell me more about the *first one*", and so on.

B. Server Side Components

All user queries are directed to a spoken dialogue service engine named APOLLO, a domain independent configurable event-driven spoken dialogue framework. The framework is able to integrate heterogeneous dialogue components (such as speech recognizer, natural language understanding, speech synthesizer, etc.) through message-oriented middleware and a number of adaptors in a plug-and-play fashion. The dialogue system is empowered by a multi-purpose XML-based dialogue engine which is capable to pipeline information flow construction, programmable event mediation, multi-topic dialogue modeling and different types of knowledge representation. The proposed platform provides a generic framework for quick and easy construction of robust, efficient and flexible spoken dialogue applications ranging from simple state-based dialogue prototype to complex frame-based, mixed-initiative, multi-topic spoken dialogue applications [21].

The dialogue framework comes with a pool of pluggable modules: audio recording, voice activity detection, speech recognition, language understanding, language generation, speech synthesis, database access, socket communication, rule engine to web search, etc. All these modules are reusable and interchangeable with any other compatible modules using the same plug-ins.

The supported backend knowledge in this framework includes structured database, text document, AIML (Artificial Intelligence Markup Language) [22], XML script, production rules, as well as internet web pages.

Figure 3 shows the backend knowledge and its related component plug-ins. In addition, a web-socket plug-in was developed for the communication with various dialogue clients, e.g. web browsers or mobile applications. Different from conventional client/server architecture, the system leverages the latest web technology for low latency and persistent communication. The communication protocol consists of various JSON objects.

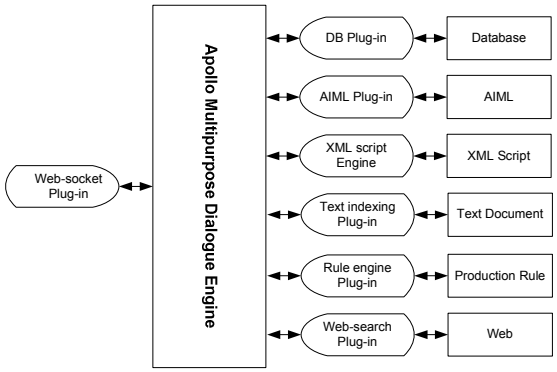

Fig. 3 APOLLO dialogue components and knowledge representation.

Since tourist domain requires many attributes, e.g. name, general description, postal address, opening hours etc. we chose to use relational databases, as they offer an efficient way of information representation. A database plug-in it is used to retrieve data base information. For every user query, the NLU module tries to translate it into a database query [23]. The following example shows the XML script for a query through the database plug-in concerning ticket prices for the Singapore flyer:

```
<querydb param="select TickPrice, GeneralURL from
sightseeingspot where entryName=\'Singapore Flyer\'"
return="_dbRes"/>
```

The database result of the query is used to generate the system response.

For general queries about Singapore (e.g. “How many people live in Singapore?”), the answers are usually quite straightforward, as they don’t contain extra attributes. Such information for example is represented as text document and retrieved by text indexing plug-ins.

In this framework the AIML knowledge base is also supported through an AIML engine: information about the conversational agent herself (e.g. “what is your name?”, “what can you do?”) can be represented in AIML language. However, more advanced knowledge representation can be supported in the form of XML script. This knowledge representation can be defined in the XML script as different object-oriented objects. The representation takes the advantages of both object-oriented technology (e.g., encapsulation, inheritance, etc.), as well as XML (e.g. platform independent, extensibility, structured representation of data and methods in human readable language, etc.).

The XML-based object-oriented model addresses some challenges for spoken dialogue applications in the following areas: multi-topic representation, mixed-initiative implementation, reactive planning and reasoning through hierarchical decomposition of objects. On the other side, the language generation, the user input verification, as well as several help techniques can be implemented with less effort by using the inherent features of the proposed model.

In addition, production rules and dynamic web search are supported through a rule engine plug-in and a web-search engine. In dialogue cases where reasoning capabilities are required the rule engine can be engaged. The system capability of reasoning depends on provided facts and implemented production rules. For those queries that the system fails to find an answer in the databases, the web search engine can perform an online search. In such cases the system response will be based on the search results.

The management and coordination of all above engines and dialogue logic are controlled by the APOLLO dialogue framework. The framework is fully driven by XML script. Apart from XML, the framework supports also Python scripting through a Python plug-in. The leverage on Python script makes the dialogue framework more flexible and efficient in terms of new dialogue development.

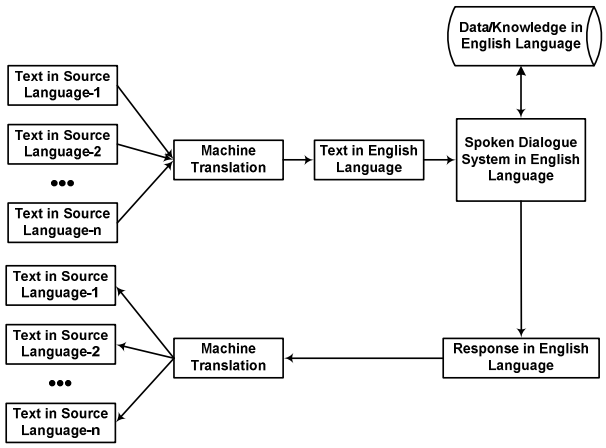

Fig.4 Schematic diagram of a multilingual dialogue system

Currently, the core knowledge sources collected for SARA are in English. In order to reuse the data collection and the trained statistical models we added machine translation plug-ins to enable SARA for other languages as well. In the machine translation process, the text in the source language is translated into the target language - in this case English.

Following on, the natural language understanding, dialogue management, database/knowledge base access and natural language generation are all performed in the target language. Next, the automatic machine translation is carried out from the target language to the source language. Finally, the speech synthesis is performed in the source language. The schematic diagram of this process is illustrated in figure 4.

At the moment, our machine translations modules support only Chinese and Malay. These modules were developed by the Human Language Technology department of the Institute for Infocomm Research (A-Star) Singapore[2] and use two-way machine translation.

III. User Interface

Currently, SARA can be accessed using two different interfaces: a web-based client interface and an Android mobile phone application.

A. *Web-based client Interface*

The web-based client interface shown in figure 5 has several components: the avatar, the text input field, the response area and the webpage display.

Fig. 5 Web-based client interface.

The avatar, as seen on the top left of the picture, is used to provide a spoken response to the user. It reads out the text returned by APOLLO. Next to the avatar a text field can be seen which allows the user to submit queries to the system. Speech input is also supported as an alternative option to text. This option is signalized by the microphone icon located on the right corner of the text box.

Once the system returns a response, the user is able to visualize the question and the answer prompted by the system in the black response area shown on the top of the picture.

The rest of the screen is reserved for a webpage display which offers additional information related to the user query. In figure 5, the user is presented with the Singapore Flyer website that is referenced in the question. This webpage is retrieved from the database and can be configured in the server-side scripts.

B. *Mobile-based client Interface*

The mobile client application is composed by four main screens, each one responsible for supporting different specific informational needs such as maps, web, scan and pictures. The navigation to each of these screens is controlled by the server-based dialogue manager. However, the screen can be also directly accessed by the user from the home screen/dashboard – see figure 6.

Fig 6. Dashboard/home screen of the mobile client app.

As seen from figure 6, the top third of the interface displays an avatar and a text bubble, which are used in combination to provide natural language responses in the form of both text and speech. Although typically, both text and speech contain the same information, the system is able to deliver different informational pieces to the avatar and the text bubble. This is particularly useful when long sentences are provided; in this case the avatar's speech can be maintained short and concise. As the user is able to turn off the speech mode, the information passed by the system to the text bubble is always equal or more informative than the one passed to the avatar.

As input method, the user can use text by typing into the editable text box or speech by clicking on the microphone icon. Clicking the microphone icon will open the Android's native speech recognition engine which will capture the user speech input and transform it into text.

The four main application screens are located on the lower part of the application. In the following paragraphs we describe the main functionalities of these four screens.

Map screen. This screen provides access to a third party map service[3], which allows displaying information related to: (1) the user's current location (as far as the user has enabled the native GPS tracking functionality of her/his smart-phone); (2) locations of interest that have resulted from the user query (in this case the server will instruct the app to automatically navigate to the map screen and display the resulting locations of interest);

[2] http://www.i2r.a-star.edu.sg/

[3] https://developers.google.com/maps/

(3) routes between the user's location and the location of interest if the user requests for transport information or directions. The locations of interest displayed in the map are clickable objects, which, if clicked, will trigger the request "*Tell me more about this location*".

Web screen. This screen represents a built-in browser that allows the display of web-pages or any other type of html content. Most of the entities registered in the system's databases and index have an associated URL that is returned by the server along with the text response. The server will guide the application to navigate to a webpage whenever the user makes a request explicitly ("Show me the website of the zoo") or implicitely ("Where can I find more information about this venue?"). Alternatively, the user can navigate to the webpage directly from the dashboard/home screen.

Scan feature. This screen integrates a standard QR-code decoder. The system implements two different types of interaction within this functionality: external codes and internal (system-specific) codes. For external codes, the app will behave as a regular QR-code decoder: if the code contains a URL, the app will navigate to the web screen and will display the specified webpage; if the code contains a text string, the string will be displayed in the text bubble and the avatar will read it out. Internal codes, which are distinguished by a specific URL signature and a numeric code, will be processed by the application as normal user requests. As such, they will will be sent to the server, which will interpret the code and respond accordingly. This functionality can be also used to trigger specific system behaviors. For any commercial decoder, the internal QR-code will provide a link to a website.

Picture gallery. This screen is used to display a collection of pictures locally stored on the mobile device. These pictures are provided by the server to accompany the information provided to a user request. The user can specifically a direct request for pictures, e.g. "Pleases show me pictures of Sentosa", "Can I see the Merlion?" etc. Once the app receives the pictures, they will remain in the local storage until deleted by the user. Usually, the system provides pictures to each user query and the user can access them by navigating to the gallery from the dashboard/home screen.

Finally, the mobile client app includes additional capabilities, such as placing phone calls or sending SMS messages for venues of interests. The phone numbers are available in either the databases or the index.

IV. SYSTEM EVALUATION

To evaluate the interfaces of our conversational agent two types of assessment methods were used: a qualitative heuristic evaluation for the mobile app and a quantitative user study for the web client application.

HEURISTIC EVALUATION

A. Evaluation design

Heuristic evaluation is a qualitative inspection method involving usability specialists going through the interface and performing several tasks judging whether the user interface follows established usability principles (heuristics).

Four experts participated in our heuristic evaluation. They received the set of tasks to perform, such as to exchange greetings, ask for locations of interest, restaurant recommendation, pictures, general questions about weather, currency, opening hours, etc., as well as to request the application to call a particular location of interest. An overview of the tasks is presented in table 2.

TAB 2. LIST OF REQUESTED TASKS TO THE EXPERTS

Task type	Task content
General question/Informal	Exchange greetings and ask SARA what she can do
General question/ Formal	Ask SARA about locations of interest
Focused question/ Web	Pick up one location and ask SARA for more detailed information about it
Focused question/ Map	Ask where "Little India" located
Focused question/ Recommendation	Ask for restaurant recommendation in the area of "Little India"
Focused question/ Pictures	Ask SARA to show of Chinatown
General question/ Formal	Ask SARA for some general questions about local currency, weather, current president of Singapore
Focused question/ Web & Map &Pictures	Ask SARA about Singapore Flyer-information, pictures, location, how to go there, opening hours etc.
Focused question/Request for phone call	Ask SARA to call the Singapore Flyer for you – since you want to make a dinner reservation at their restaurant
Scan functionality	Scan a QR code

Additionally, the experts were free to play around with the application and try it out as long as they wished. To complete the evaluation they used the heuristics created by Nielsen [24] (see table 3), as well as their own professional experience.

TAB 3. HEURISTICS BY J. NIELSEN [24]

Heuristic	Explanation
Visibility of system status	The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
Match between system and the real world	The system should speak the users' language. Follow real-world conventions, making information appear in a natural and logical order.
User control and freedom	Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
Consistency and standards	Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
Error prevention	Even better than good error messages is a careful design which prevents a problem from occurring. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
Recognition rather than recall	Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable.
Flexibility and efficiency of use	Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
Aesthetic and minimalist design	Dialogues should not contain information which is irrelevant or rarely needed.
Help users recognize, diagnose and recover from errors	Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
Help and documentation	Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation.

The experts were asked to use both input options, i.e. speech and text. Each evaluation session took about an hour. The interaction was recorded and the experts could make their comments while performing the tasks.

B. Evaluation results

The evaluation using heuristics brought us a huge amount of information concerning usability problems and interaction design failures, as well as many improvements suggestions.

Regarding the **visibility of the system status** one of the main concerns was that the mobile app tends to disconnect easily from the server presenting the user with a frozen error page. The page contains a refresh button which by tapping did not seem to work, even though the interface was trying in the background to reconnect to the server. The experts warned that users would likely get confused by not knowing what is happening with the application.

Another - less critical issue - addressed the problem of switching off the avatar's speech: in this case, experts recommend highlighting the avatar's speech bubble in color to signalize the avatar is answering the question.

Also, system functionalities, such as the information provision in the form of maps, web info and picture, as well as the scanner could be confusing for new users. For example, one expert was wondering whether the system can answer questions about directions if currently in picture mode. Another expert was confused on how to use the scanner and was asking whether it goes automatically or he had to press a button to start the scanning process. Such information could be provided in a **help and documentation** section that users could be presented with when they install the application for the first time.

Concerning the **match between system and real-world conventions** the system should take into account the fact that users expect that information regarding direction to be presented similar to google maps – currently, the application only points out the path to be followed, but does not offer any additional information on how to reach a location by bus, metro etc.

Also, the system should allow users to deepen a topic of interest by enabling further information exchange, i.e. continuation of the dialogue as it is usual in conversations - for example, the system should be able to inform users not only about restaurant directions or opening hours, but also about menu information, review summaries and dish recommendations. A good feature that experts praised was the system capability to send sms and place phone calls on request, a functionality which reminded of a good receptionist service.

The **user control and freedom** is limited as no interruptions are allowed while the avatar is talking or a web site is loaded: users don't have the option of an undo button to stop or re-start a query when the system takes too long to respond. One of the experts also commented that the web browser functionalities are limited to the current information query which is frustrating for someone used with a full browser capability.

In terms of **consistency and standards** experts pointed out these issues:

1. lack of consistency concerning the roll-on menu which does not include the scan option - see figure 7b)
2. lack of consistency concerning the usage of home buttons and the roll-on menu – these buttons and menu are included in all other screens, i.e. map, web and pictures. However, the scan screen does not have this option – see figure 7a).
3. some buttons are no recognizable as buttons (< Back) and share similarities with simple labels (Scan QR Code)- see figure 7a). This can cause confusion among users.

Another, less critical consistency problem addresses the avatar facial expressions: the avatar should display a sad face when she has no answer and smiles when she finds the right information. Otherwise, it lacks consistency between avatar's behavior on one side and the presented content, on the other side.

According to three experts, new users would wonder about the way they should talk to the system, i.e. whether they can make full sentences or use only keywords. Users might not know that they have to press the arrow button to enter their text/speech input. Also, in case when the system has no answer or gives a wrong response, users would be confused about the reasons: was it a speech recognition problem? Or maybe the system does not have that particular information? Or rather the natural language understanding/or generation modules are failing? A typical user would probably keep repeating the same question in a slightly modified version, i.e. using rather keywords than full sentences. Additionally, he would start speaking louder and at slower pace. This, of course, may or may not improve the system result depending on what caused the answer failure. Therefore, it is wise to explain users how to talk/interact with the application when they use it for the very first time.

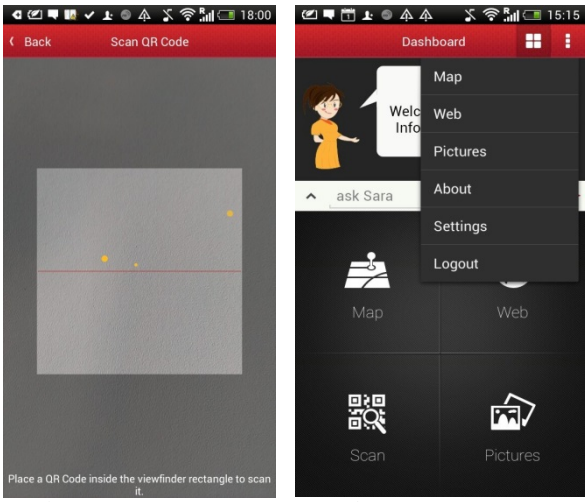

Fig 7. a) QR scanner b) Home screen

As **error prevention** measure it is advisable to enlarge the input field and allow users to see their whole query - in the current format users often experience visualization difficulties. A too small input field can lead users to oversee mistakes and submit a wrongly recognized input that would further cause system to give a wrong answer. Experts commented that SARA would be more suitable for tables and iPads since these devices are usually bigger and consequently offer more space for input fields.

Additionally, the voice input button (microphone icon) and the arrow key to submit the query are too close to each other, as shown in figure 6. The same is valid for the home button and roll-on menu, also placed very close to each other. The button closeness can be a source of input mistakes.

Another source of errors is the lack of a spell checker for text input that could reduce the probability of inputting misspelled words.

A good observation concerning error prevention refers to the fact that users might press by mistake the log out button and consequently get logged out without wishing to do so. Here is important that the system asks users for confirmation.

Regarding **aesthetic and minimalistic design** experts recommend to redesign the browser window which does not have a proper navigation, apart a from back and a forward button. Also, an additional control button meant to hide the browser navigation should be removed, as it clutters the interface and takes too much space.

When opening the interface users are presented with a start screen where they have to press a button in order to reach the login screen. Experts recommended to reduce the application start and login process to a single screen.

Another improvement suggestion refers to the redundancy of the roll-on menu for the home screen: here the roll-on menu repeats the same information (Map, Web, Picture) presented on the dash board screen (see figure 7b). These three menu options should not be mentioned when users are navigating on the home screen/dashboard.

The optional presence of the avatar was considered a good design choice as it allows users to customize the interface.

A good observation pointed out by almost all experts concerns the heuristic **recognition rather than recall:** in the current version the system does not support any kind of history to preserve previously entered questions and results. However, this information is available in the backend. Such functionality would be very useful also on the client-side as users could store the answers and retrieve them later.

In terms of **flexibility and efficiency** experts agreed that the system should be able to handle both keywords and full sentences input. Additionally, it should be trained to understand synonyms.

In the version of our mobile app there are no strategies implemented to **help users to recognize, diagnose and recover from mistakes.** Therefore, experts suggested the following:

1. in cases when confidence scores are low, present users with a list of options; this would show the system willingness to cooperate in finding a solution. Also, it would prevent users for getting frustrated by a repeated avatar response: "Sorry, I don't know what that is!"
2. when the system cannot connect to the server or it crashes it should inform the users about the cause, log the error and send it to the server.
3. if the user repeats the same questions twice, it should be an indication that the previous answer was wrong; the system should react accordingly and change the confidence scores

Experts also suggested other improvements for SARA such as changing the avatar's TTS with a more pleasant voice, improving the pronunciation for Singaporean entity names, and offering information based on user profiles. The back-end data base should be extended with more information about cafés, typical local food recommendations, daily currency, stock market information and weather forecast. Also, the system should be extended with the following functionalities: pedestrian navigation, GPS based sightseeing and restaurant recommendations, autonomous restaurant reservations, as well as personalized greetings.

TAB 4. USER SCENARIOS FOR SINGAPORE TOURISTIC AGENT EVALUATION

User Scenario	Questions
1. Tourist places (e.g. Marina Bay, Sentosa or Orchard Road which are ones of the most important touristic and shopping areas in Singapore)	Find a convenient metro station to reach the place
	Find a suitable place for having lunch
	Ask for places of interest in the surrounding area
	Select one place of interest and ask questions about it (including location)
	Find and select transportation options to go there
	Ask for places of interest nearby
	Select one place of interest and ask questions about it (including location)
	Find a suitable place for having dinner
	Ask for shopping centers in the area and select one to go
	Ask for a hotel conveniently located near the shopping center you selected
	Find a suitable place for having dinner
2. Moving to other touristic areas and places in Singapore	Ask for directions on how to go to the zoo
	Ask for information related to the zoo
	Find a suitable place for having lunch in some central Singapore area
	Find directions on how to reach that area
	Ask information about museums in Singapore
3. General info about Singapore	Ask about the type of weather in Singapore
	Ask information about the languages spoken in Singapore
	Find out information about the currency and exchange rates in Singapore
	Find out when was Singapore founded and who was the founder

USER EVALUATION

A. *Evaluation design*

To evaluate our web client we performed a user evaluation based on the following scenario cases – see table 4.

For the first scenario, participants were asked to select some of the proposed questions and fully complete the last two scenarios from the table.

In total, 10 users from our research lab participated in the study. After completing the scenarios they complete a questionnaire concerning the reliability, usability, and functionality of the system. Additionally, they could make comments and give feed-back on their experience with the application.

The provided questionnaire followed the recommendations of the ISO/IEC 9126 quality standard model [9]. For each question the evaluators had to choose between the following five categorical scores: strongly agree (SA), agree (A), neutral (N), disagree (D), or strongly disagree (SD).

B. *Evaluation results and analysis*

For each scenario we computed the objective completion rate, as the percentage of the specific objectives completed with respect to the total number of objectives in the task. Figure 8 summarizes the objective completion rates for the first scenario (Marina Bay, Sentosa and Orchard Road), the second scenario (Moving to other areas) and the third scenario (General Information).

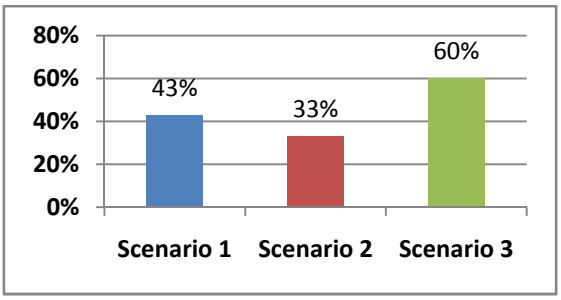

Figure 8. Objective completion rates for the considered scenarios

As seen in figure 8, scenario 3 (General Information) achieved the best result with a completion rate of 60% followed by a 42% for scenario 1. In contrast, scenario 2 (Moving to other touristic places) achieved the worst performance achieving a completion rate of 33%. Here, the most common errors reported by the users – and confirmed later by inspecting the session logs - were related with the proper identification of venues and venue directions. It is important to mention that the success of scenario 3 is due to a major coverage of general information extracted from Wikipedia and less variability in the questions formulated by the users and those used in the templates (see table 1). In contrast, for the other two scenarios, there are two possible reasons for the lower results: a) the evaluators felt freer and more willing to do these scenario since the locations are well know places for them so they use more spontaneous queries which the system was not able to understand, and b) due to the dialogue nature of the selected questions (e.g. ask for a shopping centre in the area and select one to go), therefore the system is required to handle one or more turns before providing the answer and any kind of error in the system will forbid the user to get the correct answer.

Figure 9 show the results of the subjective evaluation on user experience related with reliability, usability, and functionality of the system. In this case, for usability there were a total number of 11 questions, for reliability 5 questions (2 of them optional for cases were the system failed), and for functionality other 5 questions. As we can see, for all the three components the users agree or strongly agree in between 40-60% which means that the system is good. Regarding the answers were the evaluators disagree or strongly disagree (30-40%) most of the low scores came from the scenarios where venues and direction were not properly identified by the system.

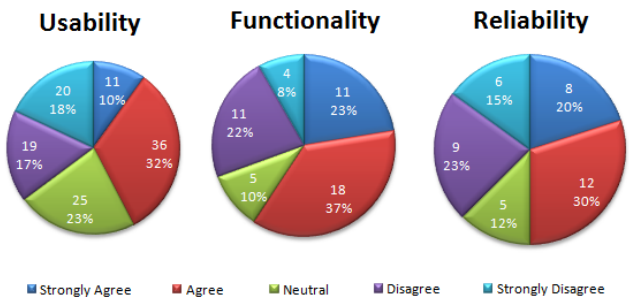

Figure 9. Results of subjective evaluation

V. CONCLUSION AND FUTURE WORK

In this paper we presented SARA, a conversational agent for the city of Singapore. SARA combines general QAs about Singapore's touristic spots and task-oriented dialogues for the same domain. The system is currently implemented as Android application and as web-based system.

The paper described in details the system's modules, plug-in components, data bases and interfaces, as well as the results of our first round of evaluations. The evaluation results showed that SARA needs several improvements before it can reach the level of a successful commercial application. The information collected during the evaluation provided valuable feed-back on service, performance and usability.

Our next goal is to work towards the improvements suggested by our test participants and design experts, enlarge the data base coverage with additional touristic information, and integrate it with our digital receptionist and restaurant recommendation.

Additionally, we are planning to take advantage of the multilingual capabilities of the Apollo framework by allowing the users to use Chinese and Malay language, and to create an iOS and Windows mobile version of the application.

Finally, we plan to conduct a second round of evaluations engaging more users to interact with our system. In particular, we will be targeting potential system users, such as tourists, visitors, new comers etc.

REFERENCES

[1] Beveridge, M. and Fox, J. Automatic generation of spoken dialogue from medical plans and ontologies. Biomedical Informatics, 39 (5)

[2] Maragoudakis, M. MeteoBayes: Effective plan recognition in a weather dialogue system. IEEE Intelligent Systems, 22(1), p. 66-77.

[3] Huang, C., Xu, P., & Zhang, X. Zhao, S., Huang, T., & Xu, B. LODESTAR: A Mandarin spoken dialogue system for travel information retrieval. Eurospeech, 1999. pp. 1159-1162.

[4] Ibrahim, A., & Johansson, P. Multimodal dialogue systems for interactive TV applications. 4th Conference on Multimodal Interfaces, p. 117-122. 2002.

[5] Johnson, W. L., & Valente, A. Tactical language and culture training systems: Using Artificial Intelligence to teach foreign languages and cultures. Proc. IAAI, (pp. 1632-1639), 2008.

[6] Bellegarda, J. Spoken Language Understanding for Natural Interaction: The Siri Experience. In Natural Interaction with Robots, Knowbots and Smartphones: Putting Spoken Dialog Systems into Practice, p 3-14. 2014.

[7] Kabassi, K. Personalizing recommendations for tourists. Telematics and Informatics, 27(1):51-66, February 2010

[8] Singapore Tourist Board, http://www.stb.gov.sg/

[9] Janarthanam, S., Lemon, O., Bartie, P., Dalmas, T., Dickinson, A., Liu, X., Mackaness, W. and Webber, B. Evaluating a city exploration dialogue system combining question-answering and pedestrian navigation. In Proc. ACL 2013.

[10] Malaka R. and A. Zipf. Deep Map – challenging IT research in the framework of a tourist information system. In Information and Communication Technologies in Tourism 2000, pages 15–27. Springer, 2000

[11] Raubal, M. and S. Winter. Enriching way finding instructions with local landmarks. In Second International Conference GIScience. Springer, USA, 2002

[12] Dale, R., Geldof, S. and Prost, J. CORAL: Using Natural Language Generation for Navigational Assistance. In Proceedings of ACSC2003, Australia, 2003

[13] Bartie, P. and W. Mackaness. D3.1.2 - The Space-Book City Model. Technical report, The SPACEBOOKProject (FP7/2011-2014 grant agreement no.270019), 2013

[14] Shroder, C.J., Mackaness, W. and Gittings, B. Giving the Right Route Directions: The Requirements for Pedestrian Navigation Systems. Transactions in GIS, pages 419–438, 2011

[15] Dethlefs, N. and H. Cuayahuitl. Hierarchical Reinforcement Learning and Hidden Markov Models for Task-Oriented Natural Language Generation. In Proc. of ACL, 2011

[16] Ko, J., Murase, F., Mitamura, T., Nyberg, E., Tateishi, M., Akahori, I. and Hataoka, N. CAMMIA: A Context-Aware Spoken Dialog System for Mobile Environments. In IEEE ASRU Workshop, 2005

[17] Kashioka, H., Misu, T., Mizukami, E., Shiga, Y., Kayama, K., Hori, C. and Kawai, H. Multimodal Dialog System for Kyoto Sightseeing Guide. In: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, 2011

[18] Webb, N. and B. Webber. Special Issue on Interactive Question Answering: Introduction. Natural Language Engineering, 15(1):1–8, 2009

[19] Lee, C., Jung, S., Kim, S., Lee, G., Example-based dialog modeling for practical multi-domain dialog system, Speech Communication. 51(5), 466–484, 2009.

[20] Hatcher, E., Gospodnetic, O., and McCandless, M., Lucene in action, 2004.

[21] R. Jiang, Y. K. Tan, D. K. Limbu, and H. Li, "Component pluggable dialogue framework and its application to social robots", in *Proc. Int'l Workshop on Spoken Language Dialog Systems*, 2012.

[22] http://en.wikipedia.org/wiki/AIML

[23] Enikuomehin A.O., Okwufulueze D.O. "An Algorithm for Solving Natural Language Query Execution Problems on Relational Databases", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 10, 2012.

[24] Nielsen, J. (1994a). Enhancing the explanatory power of usability heuristics. *Proc. ACM CHI'94 Conf.*(Boston, MA, April 24-28), 152-158.