From VoiceXML to multimodal mobile Apps: development of practical conversational interfaces

David Griol and José Manuel Molina

Computer Science Department, Carlos III University of Madrid, Avda. de la Universidad, 30. Leganés (Spain), 28911
{david.griol, josemanuel.molina}@uc3m.es

KEYWORD

Conversational interfaces; VoiceXML; Mobile devices; Android

ABSTRACT

Speech Technologies and Language Processing have made possible the development of a number of new applications which are based on conversational interfaces. In this paper, we describe two approaches to bridge the gap between the academic and industrial perspectives in order to develop conversational interfaces using an academic paradigm for dialog management while employing the industrial standards. The advances in these technologies have made possible to extend the initial applications of conversational interfaces from only spoken interaction (for instance, by means of VoiceXML-based systems) to multimodal services by means of mobile devices (for instance, using the facilities provided by the Android OS). Our proposal has been evaluated with the successful development of different spoken and multimodal conversational interfaces.

1. Introduction

Recent advances in conversational interfaces has been propelled by the convergence of three enabling technologies. First, the Web emerged as a universal communications channel. Web-based conversational interfaces are scalable enterprise systems that leverage the Internet to simultaneously deliver dialog services to large populations of users. Second, the development of mobile technologies and intelligent devices, such as smartphones and tablets, have made it possible to deploy a large number of sensors and to integrate them into conversational interfaces that provide multimodal interaction capabilities (i.e., use of different modalities for the input and/or output of the system) and allow their access in almost every place and at any time. Third, computational linguistics, the field of artificial intelligence that focuses on natural language software, has significantly increased speech recognition, natural language understanding and speech synthesis capabilities (McTear et al., 2016).

These advances have extended the initial application domains of conversational interfaces to complex information retrieval and question answering applications (Metze et al., 2014), e-commerce systems (Tsai, 2005), surveys applications (Stent et al., 2006), in-car systems (Hofmann et al., 2014), remote control of devices and robots in smart environments (Minker et al., 2010), e-learning and tutoring systems (Kopp et al., 2012), communication within vehicles (Misu et al., 2015), Ambient Assisted Living systems (Bickmore et al., 2010), recommendation systems (Reschke et al., 2013), or virtual companions (Horchak et al., 2014).

In this paper, we describe two approaches to bridge the gap between the academic and industrial perspectives in order to develop conversational interfaces using an academic paradigm for dialog management while employing the industrial standards. Our first approach to integrate statistical methodologies in industry applications combines the flexibility of statistical dialog management with the facilities that the VoiceXML language offers (Rouillard, 2007; González-Ferreras et al., 2006), thus introducing statistical methodologies for the development of commercial (and not strictly academic) conversational interfaces. Our technique employs a statistical model based on neural networks that takes into account the history of the dialog up to the current dialog state in order to predict the next system response (Griol et al., 2008). To learn the dialog model we propose the use of dialog simulation techniques. Our approach for acquiring a dialog corpus is based on the interaction of a user simulator and a dialog manager simulator (Griol et al., 2013). In addition, the system prompts and the grammars for ASR are implemented in VoiceXML compliant formats, for example, JSGF[1] or SRGS[2].

Multimodal conversational systems offer the user combinations of input and output modalities for interacting with mobile devices, taking advantage of the naturalness of speech (Pieraccini, 2012). Different vendors offer APIs for the development of applications that use speech as a possible input and output modality, but developers have to design ad-hoc solutions to implement the interaction management. Speech access is then a solution to the shrinking size of mobile devices (both keyboards to provide information and displays to see the results). Besides, speech interfaces facilitate the access to multiagent systems (Corchado et al., 2008), especially in environments where this access is not possible using traditional input interfaces (e.g., keyboard and mouse). It also facilitates information access for people with visual or motor disabilities.

Our second approach is focused on the development of multimodal conversational agents for mobile devices operating with the Android OS (McTear and Callejas, 2013). Our proposal integrates the Google Speech API to include the speech recognition functionality in a multimodal conversational agent. The development of multimodal systems involves user inputs through two or more combined modes, which usually complement spoken interaction by also adding the possibility of textual and tactile inputs provided using physical or virtual keyboards and the screen. In our contribution, we also model the context of the interaction as an additional valuable information source to be considered in the fusion process. We propose the acquisition of external context by means of the use of sensors currently supported by Android devices. The Android sensor framework (*android.hardware* package) allows to access these sensors and acquire raw sensor data.

The remainder of the paper is organized as follows. Section 2 describes the main modules of a conversational interface, the main approaches to develop them, and the main challenges that different experts have envisioned as future research guidelines. Section 3 describes our first approach to develop conversational interfaces using the VoiceXML language. A practical example of our proposal to implement interactive voice portals that provide municipal information is described. Section 4 describes our proposal focused on the development of multimodal conversational interfaces for mobile devices operating with the Android OS. We describe a practical App that facilitates the interaction by means of speech or using the screen and virtual keyboard. The services that are provided by the App include accessing the latest local and international news, the weather forecast for the coming days for the current place, the results of different lottery contests and events, the movie listings and upcoming movies. Finally, in Section 5 we present the conclusions and outline guidelines for future work.

2. Related work

A conversational agent is a software that accepts natural language as an input and produces natural language as an output engaging in a conversation with the user (McTear et al., 2016; Pieraccini, 2012). To successfully manage the interaction with the users, conversational agents usually carry out five main tasks: automatic speech recognition (ASR), natural language understanding (NLU), dialog management (DM), natural language generation (NLG) and text-to-speech synthesis (TTS).

Speech recognition (Rabiner and Juang, 1993; Baker et al., 2009) is the process of obtaining the text string corresponding to an acoustic input . It is a very complex task as there is much variability in the input characteristics, which can differ depending on the linguistics of the utterance, the speaker, the interaction context and the transmission channel. Linguistic variability involves differences in phonetic, syntactic and semantic components

[1] https://www.w3.org/TR/jsgf/

[2] https://www.w3.org/TR/speech-grammar/

that affect the voice signal. Inter-speaker variability refers to the big difference between speakers regarding their speaking style, voice, age, sex or nationality.

Once the conversational agent has recognized what the user uttered, it is necessary to understand what he said. Natural language processing is the process of obtaining the semantic of a text string (Minker, 1998; Baker et al., 2009). It generally involves morphological, lexical, syntactical, semantic, discourse and pragmatical knowledge. Lexical and morphological knowledge allow dividing the words in their constituents distinguishing lexemes and morphemes. Syntactic analysis yields a hierarchical structure of the sentences, while semantic analysis extracts the meaning of a complex syntactic structure from the meaning of its constituents. In the pragmatic and discourse processing stage, the sentences are interpreted in the context of the whole dialog.

There is not a universally agreed upon definition of the tasks that a dialog manager has to carry. Traum and Larsson (Traum and Larsson, 2003) state that dialog managing involves four main tasks: i) updating the dialog context, ii) providing a context for interpretations, iii) coordinating other modules and iv) deciding the information to convey and when to do it. Thus, the dialog manager has to deal with different sources of information such as the NLU results, database queries results, application domain knowledge, knowledge about the users and the previous dialog history (Griol et al., 2014; Bohus and Rudnicky, 2003).

Natural language generation is the process of obtaining texts in natural language from a non-linguistic representation. The simplest approach consists in using predefined text messages (e.g. error messages and warnings). Finally, a text-to-speech synthesizer is used to generate the voice signal that will be transmitted to the user.

Human beings have always been interested in being able to communicate with artificial companions. In fact, one of the main challenges of AI since his early days has been to achieve the man-machine communication through natural language. At the beginning of the XX century J.Q. Stewart built a machine that could generate vocalic sounds electrically; and during the 30s, the first electric systems covering all sounds were built. The first one was the VOCODER, an speech analyzer and synthesizer developed in Bell Laboratories that could be operated by a keyboard. At the same time appeared the first systems with very basic natural language processing capabilities for machine translation applications.

During the 40s, the first computers were developed and some prominent scientists like Alan Turing pointed out their potential for applications demanding “intelligence”. This was the starting point that fostered the research initiatives that in the 60s yielded the first conversational agents. For example Weizenbaum’s ELIZA (Weizenbaum, 1966), which was based on keyword spotting and predefined templates.

Benefiting from the incessant improvements in the areas of speech recognition, natural language processing and speech synthesis, the first research initiatives related to spoken dialog systems appeared in the 80s. To some extent the origin of this research area is linked to two seminal projects: the DARPA Spoken Language Systems in the USA (DARPA, 1992) and the Esprit SUNDIAL in Europe (Peckham, 1993).

Among the most important research projects in the 90s with multi-domain capabilities, stands out the DARPA Communicator. This government-funded project aimed at the development of cutting-the-edge speech technologies, which could employ as an input not only speech but also other modalities. Currently experts have proposed higher level objectives to develop dialog systems, such as providing the system with advanced reasoning, problem solving capabilities, adaptiveness, proactiveness, affective intelligence, multimodality and multilinguality (Dybkjaer and Minker, 2008). As can be observed, these new objectives are referred to the agent as a whole.

Throughout the last years, some experts have dared to envision what the future research guidelines in the application of multimodal conversational interfaces would be. These objectives have gradually changed towards ever more complex goals, such as providing the system with advanced reasoning, problem solving capabilities, adaptiveness, proactiveness, affective intelligence, and multilinguality. All these concepts are not mutually exclusive, as for example the system's intelligence can also be involved in the degree to which it can adapt to new situations, and this adaptiveness can result in better portability for use in different environments.

Proactiveness is necessary for computers to stop being considered a tool and becoming real conversational partners. Proactive systems have the capability of engaging in a conversation with the user even when he has not explicitly requested the system's intervention. This is a key aspect in the development of ubiquitous computing architectures in which the system is embedded in the user's environment, and thus the user is not aware that he is interacting with a computer, but rather he perceives he is interacting with the environment. To achieve this goal, it is necessary to provide the systems with problem-solving capabilities and context-awareness.

Adaptivity may also refer to other aspects in speech applications. There are different levels in which the system can adapt to the user. The simplest one is through personal profiles in which the users have static choices to customize the interaction. Systems can also adapt to the users' environment, for example ambient intelligence applications such as the ubiquitous proactive systems described. A more sophisticated approach is to adapt to the user's knowledge and expertise. This is especially important in educative systems to adapt the system taking into account the specific evolution of each of the students, the previous uses of the system, and the errors that they have made during the previous interactions.

There is also an increasing interest in the development of multimodal conversational systems that dynamically adapt their conversational behaviors to the users' affective state. The empathetic educative agent can thus indeed contribute to a more positive perception of the interaction.

Portability is currently addressed from very different perspectives, the three main ones being domain, language and technological independence. Ideally, systems should be able to work over different educative application domains, or at least be easily adaptable between them. Current studies on domain independence center on how to merge lexical, syntactic and semantic structures from different contexts and how to develop dialog managers that deal with different domains.

Finally, technological independence deals with the possibility of using multimodal systems with different hardware configurations. Computer processing power will continue to increase, with lower costs for both processor and memory components. The systems that support even the most sophisticated multimodal applications will move from centralized architectures to distributed configurations and thus must be able to work with different underlying technologies.

3. VoiceXML-based conversational systems: Interactive voice portals to provide municipal information

In this section we describe a voice portal that integrates different technologies such as the VoiceXML standard, databases, web and speech servers, and several programming languages (SQL, PHP, HTML), which make it more dynamic and flexible and increase its quality, efficiency, and adaptation to the users' specific preferences and needs. The functionalities of the system are to consult information about the City Council (Government Team, Councils, etc.) and the city (history, geographic and demographic data, access to the city, yellow pages, movie show times, news, events, weather, etc.), carry out several steps and procedures (check lists and personal files, book municipal facilities or make an appointment), complete surveys, access the citizen's mailbox to leave messages for suggestions and complaints, and be transferred to the City Council to be attended by a teleoperator.

The *Home module* implements the first dialog that is provided to the user. The options that the system provides to users are divided into five modules that takes into account the type of interaction and data that is facilitated: information, procedures and formalities, surveys, mailbox of the citizen, and human operator. Therefore, this module facilitates the access to the rest of functionalities provided by the portal.

The *Information module* provides specific information about the city. This information has been divided into six categories and classified so that users can easily access each functionality provided by this module:

- *City Council*: This module provides specific information related to the local government, local governing entities, teams and areas.
- *City*: By means of this module it is possible to access information related to the city (history, access, and yellow pages with bars, cafes, restaurants, shops, hostels, hotels and entertainment guides for cinemas and theaters).
- *Thematic Areas*: This module integrates the rest of information that users can consult by means of the voice portal. This information has been divided into 15 areas, which respectively provide general information, competences and contact details. Additional functionalities can be perfectly incorporated by simply adding more static information in the corresponding database of the application.
- *News*: This section provides current news of the municipality, including the date, title and corresponding description of each one of them.
- *Events*: This section reproduces the list of events of the municipality, providing its specific area, title, date, location, and detailed description.
- *Weather information*: Users can get the current weather information from the municipality and a forecast for the next two days.

The *Procedures and Formalities* module provides information about procedures and steps, such as verifying the correct incorporation into specific lists, checking the status of dossiers and documents, booking a municipal facility, or making an appointment related to a specific municipal service. The *Surveys* module allows the City Council to easily and quickly know citizens' opinion about specific issues and queries. Surveys in the application have been designed to be completely anonymous, so the voice portal does not require the users' identification. Once users have finished answering a survey, they are given the options to listen to the answers they have provided. An example of a dialog for a user asking about a specific procedure is shown below (dialog translation from Spanish to English).

```
S: Here you can access and complete procedures and formalities.
   To verify whether you are included in a list or not, say list or press 1. To check the
   status of your dossier, say dossier or press 2. To make a reservation of a sport
   facility, say reservation or press 3. If you want to schedule an appointment, say
   appointment or press 4.
U: Dossier.
S: Please tell us your ID saying or dialing the 8 digits one by one.
U: 47452060.
S: You have‘provided 47452060. The status of your dossier is as follows: Your certificate
   of registration has been
   requested.
S: What do you want to do now?
   To return to the main menu of the voice portal, say main or press 1. To access or
   complete other procedures, say procedures or press 2. If you want to exit the application,
   say exit or press 3.
U: Exit.
S: You have chosen to exit the application. Thanks for using our voice portal. See you soon!
```

The *Citizen's mailbox* implements the functionality of recording a user' speech message and store it for further processing. Thus, citizens can provide their requests, complaints, claims or comments at anytime and anywhere. The Citizen's mailbox is then managed by a specific Office of the City Hall. In addition, if users provide their contact information (telephone, mobile phone or email), this Office would contact them to provide a personalized response to their request. Finally, the *Teleoperator* module transfers the user's call to a human operator.

3.1 Architecture of the application

The voice portal has been developed following the client-server paradigm with the architecture described in Figure 1.

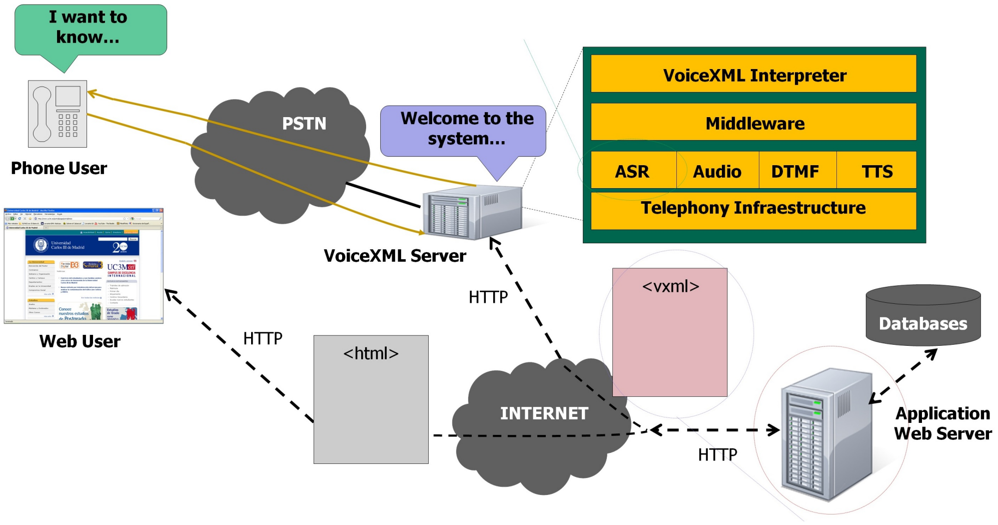


Figure 1: Architecture designed to develop interactive voice portals

The architecture is based on two main components: an IVR (Interactive Voice Response) server and a set of web servers. The IVR provides users with pages following the VoiceXML standard[3], the ASR and TTS interfaces, and VoIP and telephony technologies. Different web servers connected to the IVR via Internet provide dialog management facilities, grammars and system prompts, and the access to the information and different web services.

Regarding the VoiceXML server, it receives the users' calls and interprets the documents to provide the required services. The interpreter, in our case the Voxeo Evolution platform[4], also requests the required resources for the application, defines the logic of the services and stores users' session state to interact accordingly. To carry out these actions, the VoiceXML interpreter includes different systems to deal with users' calls, manage the communication with the servers and access the required resources, play audio files, convert text to speech, collect user data, perform voice recording, and manage sessions and events.

There are currently many VoiceXML language interpreters. One of the most important ones, given the number of functionalities provided, is Voxeo Evolution[5]. Voxeo allows creating VoiceXML applications and access them by means of a local phone number and/or a Skype number. Voxeo also allows to track calls in real time, as well as automatically create log files. These files are very useful for debugging and optimizing the

[3] http://www.w3.org/TR/voicexml21/
[4] http://evolution.voxeo.com/
[5] http://evolution.voxeo.com/

application. In addition, the Voxeo platform provides a fast and efficient support system, which includes forums, support tickets and very complete documentation. Finally, Voxeo also provides the VoiceXML interpreter and the ASR and TTS components required for the voice portal. Our system uses the Prophecy 9 Multi-Language VXML implementation, which has allowed to develop the application for its use in Spanish.

Regarding the Web servers, PHP and VXML files are used to implement each service provided by the voice portal, in addition to access MySQL databases containing the specific information. The different functionalities and corresponding files allow users to complete more than one action in each call.

3.2 Providing a personalized service

The information provided by the voice portal can be classified into static and dynamic information. Static information has been collected from web pages, stored and classified in the databases of the application. Each time users request this information, the system accesses the database and returns this information encapsulated into a VoiceXML file. Examples of this type of information include the history of the city, access information, or contact information of hotels and main offices in the city. Dynamic information includes local news and events, weather information, surveys, and entertainment guides for cinemas and theaters. This information is automatically updated in the application by means of a PHP-based procedure that access the required web pages, carries out a syntax processing of this information, and stores the updated information the database. Each time the user requires this type of information, the system only has to access the database and return it.

All the application dialogs use voice grammars and DTMF, which means that users can access menus by speech or using the phone keys, making the application more accessible. Grammars are encoded following the XML standard format defined by the W3C and, therefore, supported by any VoiceXML platform. In addition, this format allows greater flexibility in terms of grammar structure and debugging. Static grammars deal with information that does not vary over time, including a small number of options to choose from. These grammars are coded in the same file where they are used. Dynamic grammars include information that varies with time and often deal with large amounts of data. These grammars are automatically created using PHP files to manage their contents (creation, obtain contents, modify and update information).

One of the main aspects in the development of the voice portal is the introduction of different functionalities that allow the adaptation of the system taking into account the current state of the dialog as long as the characteristics of each user. On the one hand, we captured the different VoiceXML events considering different messages for the main events: *noinput* (the user does not answer in a certain time interval or it was not sensed by the recognizer), *nomatch* (the input did not match the recognition grammar or was misrecognized) and *help* (the user explicitly asks for help).

Additionally, VoiceXML provides the *property* element to establish the value of a property that affects the behavior of the platform. These properties may be defined for the whole application, for the document, or for a certain element in a form or menu. For the implementation of the voice portal we have tuned the following properties: *Confidencelevel*, *Sensitivity*, *Documentfetchhint* y *Grammarfetchhint*. The property *Confidencelevel* allows to adjust the accuracy of recognition in order to be accepted. The *Sensitivity* allows to adjust the sensitivity of the recognizer. The properties *Documentfetchhint* and *Grammarfetchhint* allow to adjust the usage of the cache to make searches either safer or faster. In our voice portal all these properties are adjusted dynamically depending on the analysis of the generated events and the history of the dialog.

On the other hand, the voice portal adapts to specific characteristics of the users. It can be used in different languages (Spanish, English, French, German and Italian), as the speech recognition is tuned using the property *xml:lang* and the prompts have been stored in the different languages using different encodings in the database. Also the voice portal stores the telephone numbers from which the users access the system in order to compute which are the most frequent queries and predict the user preferences which can be directly accessed by the user in the next calls in order to safe time and provide a better user experience.

4. Multimodal conversational interfaces for Android mobile devices

Our second proposal is focused on the development of multimodal conversational agents for mobile devices operating with the Android OS (McTear and Callejas, 2013). The Google Speech API is integrated to include the speech recognition functionality in a multimodal conversational agent. The development of multimodal systems involves user inputs through two or more combined modes, which usually complement spoken interaction by also adding the possibility of textual and tactile inputs provided using physical or virtual keyboards and the screen. In our contribution, we also model the context of the interaction as an additional valuable information source to be considered in the fusion process. We propose the acquisition of external context by means of the use of sensors currently supported by Android devices. The Android sensor framework (*android.hardware* package) allows to access these sensors and acquire raw sensor data.

Using the Google Speech API (package *android.speech*), speech recognition can be carried out by means on a *RecognizerIntent*, or by creating an instance of *SpeechRecognizer*. The former starts the intent and process its results to complete the recognition, providing feedback to the user to inform that the ASR is ready or there were errors during the recognition process. The latter provides developers with different notifications of recognition related events, thus allowing a more fine-grained processing of the speech recognition process. In both cases, the results are presented in the form of an N-best list with confidence scores.

The dialog manager of the system is based on a statistical methodology (Griol et al., 2014). The visual structure of the user interface (UI) is defined by means of layouts, which are defined by declaring UI elements in XML or instantiating layouts elements at runtime. Finally, we propose the use of the Google TTS API to include the text-to-speech functionality. The *android.speech.tts* package includes the classes and interfaces required to integrate text-to-speech synthesis in an Android application.

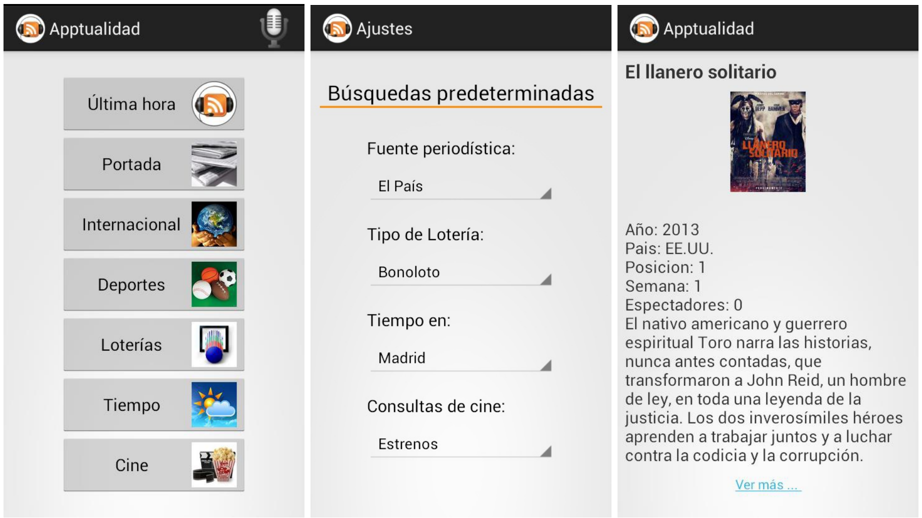
The text-to-speech functionality has been available on Android devices since Android 1.6 (API Level 4). To listen a sample of the included TTS speech synthesizer, once located in the settings menu of the device, the option Settings of Speech Synthesis must be selected in the menu Speech Input and Output. This menu allows selecting the TTS engine, language, and speed used to read a text (from very low to very fast). The *android.speech.tts* package includes the classes and interfaces required to integrate text-to-speech synthesis in an Android application. They allow the initialization of the TTS engine, a callback to return speech data synthesized by a TTS engine, and control the events related to completing and starting the synthesis of an utterance, among other functionalities.

4.1 A multimodal entertainment App

We have developed a practical multimodal entertainment App for Android-based mobile devices. Users can interact with the developed application by means of their speech or using the screen and virtual keyboard. The App allows to access the latest local and international news, the weather forecast for the coming days and current place, the results of different lottery contests and events, and the movie listings and upcoming movies. The information is provided in Spanish. Users can also personalize the information that is provided by the App by means of specifying their preferences when accessing the different services.

In order to provide the functionalities described, the system engages in a dialog with the user to retrieve different pieces of information that are complemented with the context-awareness capabilities of the system. This way, the system response is adapted taking into account the specific preferences and suggestions selected by the users, as well as to the context in which the interaction takes place. The statistical models for the user's intention recognizer and dialog management modules were learned using a corpus acquired by means of an automatic dialog generation technique previously developed (Griol et al., 2011).

Figure 2 shows the main screen of the application, the screen that users can employ in order to personalize the services provided by the App, and an example of the information provided for a specific movie.



Búsquedas predeterminadas

Fuente periodística:
El País

Tipo de Lotería:
Bonoloto

Tiempo en:
Madrid

Consultas de cine:
Estrenos

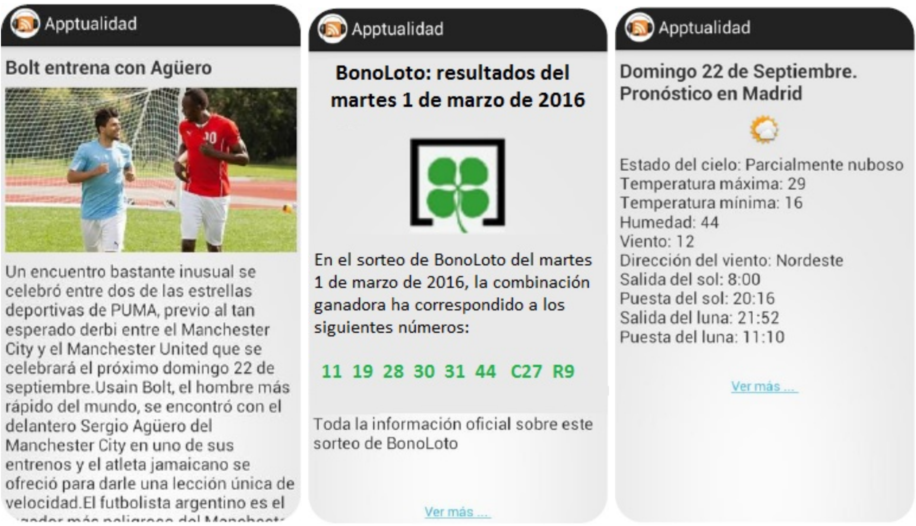
El llanero solitario

Año: 2013
Pais: EE.UU.
Posicion: 1
Semana: 1
Espectadores: 0
El nativo americano y guerrero espiritual Toro narra las historias, nunca antes contadas, que transformaron a John Reid, un hombre de ley, en toda una leyenda de la justicia. Los dos inverosímiles héroes aprenden a trabajar juntos y a luchar contra la codicia y la corrupción.

Ver más ...

Figure 2: Set of functionalities provided by the developed App (main screen, personalized user profiles, and movies listing)

Figure 3 shows different examples corresponding to the access of the latest news, the results of a specific lottery contest for a given day, and the weather forecast corresponding to the date provided by the user.



Bolt entrena con Agüero

Un encuentro bastante inusual se celebró entre dos de las estrellas deportivas de PUMA, previo al tan esperado derbi entre el Manchester City y el Manchester United que se celebrará el próximo domingo 22 de septiembre.Usain Bolt, el hombre más rápido del mundo, se encontró con el delantero Sergio Agüero del Manchester City en uno de sus entrenos y el atleta jamaicano se ofreció para darle una lección única de velocidad.El futbolista argentino es el

BonoLoto: resultados del martes 1 de marzo de 2016

En el sorteo de BonoLoto del martes 1 de marzo de 2016, la combinación ganadora ha correspondido a los siguientes números:

11 19 28 30 31 44 C27 R9

Toda la información oficial sobre este sorteo de BonoLoto

Ver más ...

Domingo 22 de Septiembre. Pronóstico en Madrid

Estado del cielo: Parcialmente nuboso
Temperatura máxima: 29
Temperatura mínima: 16
Humedad: 44
Viento: 12
Dirección del viento: Nordeste
Salida del sol: 8:00
Puesta del sol: 20:16
Salida del luna: 21:52
Puesta del luna: 11:10

Ver más ...

Figure 3: Set of functionalities provided by the developed App (latest news, lottery contests, and weather forecast)

The developed multimodal App also uses Google Maps, Google Directions and Google Places. Google Maps Android API makes it possible to show an interactive map in response to a certain query. It is possible to add markers or zoom to a particular area, also to include images such as icons, highlighted areas and routes. Google Directions is a service that computes routes to reach a certain spot walking, on public transport or bicycle, and it is possible to specify the origin and destination as well as certain intermediate spots. Google Places shows detailed information about sites corresponding to number of categories currently including 80 million commerces and other interesting sites. Each of them include information verified by the owners and moderated contributors. The application also employs the *android.speech* libraries described in the previous section.

5. Conclusions

In this paper, we propose two techniques for developing conversational agents using well-known standards and operative systems like VoiceXML or Android, and also including a statistical dialog manager automatically learned from a dialog corpus. The main objective of our work is to reduce the gap between academic and industry perspectives and take the best of both methodologies. On the one hand, the effort that is required for the definition of optimal dialog strategies is reduced. On the other, VoiceXML and Android-based implementations makes it possible to benefit from the advantages of using the different devices and platforms that are already available to simplify the development of conversational agents. The paper also describes two systems developed using the described techniques and respectively providing spoken or multimodal access to users' adapted information.

We have described a practical application of the combination of conversational agents and hand-held Android mobile devices to develop context-aware multimodal applications. The developed Android conversational agent uses geographical context and user profiles to provide adapted entertainment information and services to its users. To develop this system we have defined the complete requirements for the task and developed the different modules, and the necessary information sources to be incorporated in the user profiles.

We are currently undergoing the next phases in the deployment of the application. We want to include additional functionalities to facilitate the location of points of interest related to the provided user preferences, and to also consider additional information sources related to the users' emotional state and personality for a more detailed adaptation of the services that are provided. With the results of these activities, we will optimize the system, and make it available in Google Play.

6. References

Baker, J., Deng, L., Glass, J., Khudanpur, S., Lee, C., Morgan, N., and O'Shaughnessy, D., 2009. Developments and directions in speech recognition and understanding. *IEEE Signal Processing Magazine*, 26(3):75–80.

Bickmore, T., Puskar, K., Schlenk, E., Pfeifer, L., and Sereika, S., 2010. Maintaining reality: Relational agents for antipsychotic medication adherence. *Interacting with Computers*, 22:276–288.

Bohus, D. and Rudnicky, A., 2003. RavenClaw: Dialog management using hierarchical task decomposition and an expectation agenda. In *Proc. of 8th European Conference on Speech Communication and Technology (Eurospeech'03)*, pages 597–600. Geneva, Switzerland.

Corchado, J., Tapia, D., and Bajo, J., 2008. A multi-agent architecture for distributed services and applications. *Computational Intelligence*, 24(2):77–107.

DARPA, 1992. Speech and Natural Language Workshop. In *Book of Proceedings*. San Mateo.

Dybkjaer, L. and Minker, W., 2008. *Recent Trends in Discourse and Dialogue*. Springer.

González-Ferreras, C., Escudero, D., and Cardeñoso, V., 2006. From HTML to VoiceXML: A First Approach. *LNCS*, 2448:266–279.

Griol, D., Callejas, Z., López-Cózar, R., and Riccardi, G., 2014. A domain-independent statistical methodology for dialog management in spoken dialog systems. *Computer, Speech and Language*, 28(3):743–768.

Griol, D., Carbó, J., and Molina, J., 2013. A statistical simulation technique to develop and evaluate conversational agents. *AI Communication*, 26(4):355–371.

Griol, D., Hurtado, L., Segarra, E., and Sanchis, E., 2008. A Statistical Approach to Spoken Dialog Systems Design and Evaluation. *Speech Communication*, 50(8-9):666–682.

Griol, D., Sánchez-Pi, N., Carbó, J., and Molina, J., 2011. An Agent-Based Dialog Simulation Technique to Develop and Evaluate Conversational Agents. *Advances in Intelligent and Soft Computing (PAAMS'11)*, 88:255–264.

Hofmann, H., Silberstein, A., Ehrlich, U., Berton, A., Muller, C., and Mahr, A., 2014. *Natural Interaction with Robots, Knowbots and Smartphones: Putting Spoken Dialog Systems into Practice*, chapter Development of Speech-Based In-Car HMI Concepts for Information Exchange Internet Apps, pages 15–28. Springer.

Horchak, O., Giger, J.-C., Cabral, M., and Pochwatko, G., 2014. From demonstration to theory in embodied language comprehension: A review. *Cognitive Systems Research*, 29-30:66–85.

Kopp, K., Britt, M., Millis, K., and Graesser, A., 2012. Improving the efficiency of dialogue in tutoring. *Learning and Instruction*, 22(5):320–330.

McTear, M. and Callejas, Z., 2013. *Voice Application Development for Android*. Packt Publishing.

McTear, M. F., Callejas, Z., and Griol, D., 2016. *The Conversational Interface*. Springer.

Metze, F., Anguera, X., Barnard, E., Davel, M., and Gravier, G., 2014. Language independent search in MediaEval's Spoken Web Search task. *Computer, Speech and Language*, 28(5):1066–1082.

Minker, W., 1998. Stochastic versus rule-based speech understanding for information retrieval. *Speech Communication*, 25(4):223–247.

Minker, W., Heinroth, T., Strauss, P., and Zaykovskiy, D., 2010. *Human-Centric Interfaces for Ambient Intelligence*, chapter Spoken Dialogue Systems for Intelligent Environments, pages 453–478. Elsevier.

Misu, T., Raux, A., Gupta, R., and Lane, I., 2015. Situated language understanding for a spoken dialog system within vehicles. *Computer Speech and Language*, 34:186–200.

Peckham, J., 1993. A new generation of spoken dialogue systems: results and lessons from the SUNDIAL project. In *Proc. of 3rd European Conference on Speech Communication and Technology (Eurospeech'93)*, pages 33–42. Berlin, Germany.

Pieraccini, R., 2012. *The Voice in the Machine: Building computers that understand speech*. MIT Press.

Rabiner, L. and Juang, B., 1993. *Fundamentals of Speech Recognition*. Prentice Hal.

Reschke, K., Vogel, A., and Jurafsky, D., 2013. Generating Recommendation Dialogs by Extracting Information from User Reviews. In *Proc. of ACL'13*, pages 499–504.

Rouillard, J., 2007. Web services and speech-based applications around VoiceXML. *Journal of Networks*, 2(1):27–35.

Stent, A., Stenchikova, S., and Marge, M., 2006. Reinforcement learning of dialogue strategies with hierarchical abstract machines. In *Proc. of SLT'06*, pages 210–213.

Traum, D. and Larsson, S., 2003. *The Information State Approach to Dialogue Management*, chapter Current and New Directions in Discourse and Dialogue, pages 325–353. Kluwer.

Tsai, M., 2005. The VoiceXML dialog system for the e-commerce ordering service. In *Proc. of CSCWD'05*, pages 95–100.

Weizenbaum, J., 1966. ELIZA - A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9:36–45.