Development and Evaluation of a Spoken Dialog System to Access a Newspaper Web Site

César González-Ferreras, Valentín Cardeñoso-Payo

Departamento de Informática.
Universidad de Valladolid, Spain.
{cesargf,valen}@infor.uva.es

Abstract

This paper presents a system that provides access to a newspaper web site. The system is based on an interaction model, that uses browse and search strategies, and on an information model, which is made up of a tree and indexes. A frame-based approach is used to control the dialog flow. VoiceXML is used as a language to describe dialog turns. The systems works for Spanish language.

To measure the usability of the system we evaluated the performance using objective measures and user satisfaction using SASSI questionnaire. The results of the evaluation show a task success rate of 92% and a WER of 18.09%. Overall user satisfaction is positive: users perceive the system as useful and easy to use. We conclude the paper with a discussion of the most relevant issues about the development and evaluation of the system.

1. Introduction

In the last decades there has been a lot of work in building spoken dialog systems to allow people communicate with machines using speech. Speech has some advantages over traditional graphic user interfaces because it is more natural and has more expressive power. People is used to communicate everyday using speech. Moreover, speech is the most suitable modality for some environments (e.g. while driving a car) and for some kind of users (e.g. disabled people).

Nowadays web contents are an important source of information and web browsers the standard way of accessing them. Furthermore, mobile phones allow access to the web anytime and everywhere. However they have small displays that difficult the interaction. Using speech in that scenario will allow a more usable interaction. Speech could be used alone in a spoken dialog system or combined with the graphical user interface in a multimodal dialog system.

Recently, spoken dialog systems interact with users in a natural and flexible way while achieving high rates of performance, [1, 2]. They allow access to structured information over the telephone line. Their success has led to a mainstream use of spoken dialog systems. VoiceXML has become the standard for developing commercial Interactive Voice Response Applications. Finally, researchers are developing frameworks and toolkits to assist in the development of such spoken dialog systems.

There is a growing interest in providing speech access to web contents. The main difference with traditional spoken dialog systems is that textual information lacks the required structure. The limitations of the speech channel are a problem too, because it is not possible to send much information over it. Different approaches have been proposed: extending a web browser using speech, [3]; automatically infer some structure from HTML code, [4]; semi-automatic conversion from HTML to VoiceXML using manual annotation, [5]; automatic generation of dialogue systems for specific web content, [6]; use speech as the input to an information retrieval engine, [7].

This paper presents the development and evaluation of a spoken dialog system which provides access to a newspaper web site. The system is based on an information model, which structures the information, and on an interaction model, which describes how the interaction is carried out. First, the information model is built automatically from web contents. Next, the system uses that model to interact with the user using browse and search strategies. If the user has a specific information need, search can be used to access it directly. Browse can also be used to see which information is available.

The implementation of the system uses a frame-based approach to control the dialog flow. The system uses VoiceXML as language to describe the interaction. This allows us to separate the dialog control from the details of the speech technology. At a given point of the interaction, the dialog manager selects the next action to be done and a new VoiceXML page is generated. Different templates are used to build such pages. Our system works for Spanish language.

We carried out an evaluation of the system to measure its usability. System performance is measured, obtaining a task success rate of 92% and a WER of 18.09%. User satisfaction is measured, obtaining positive results. Users said the system is useful and easy to use, although the interaction is repetitive and boring.

The structure of the paper is as follows: section 2 explains the system in detail; section 3 describes the results of the evaluation; in section 4 we discuss about the system and its usability; section 5 presents conclusions and future work.

2. System overview

Two models are used to describe the system: interaction model and information model. A frame-based approach is used to control the dialog flow. VoiceXML is used as language to describe each turn of the dialog.

2.1. Interaction model

To access news the user can use two different strategies: browse and search. Browse allows the user to see what information is available before selecting a news. However, if the user has an specific information need, a query can be used to access it directly.

This work has been partially supported by *MCyT* under project number TIC2003-08382-C05-03.

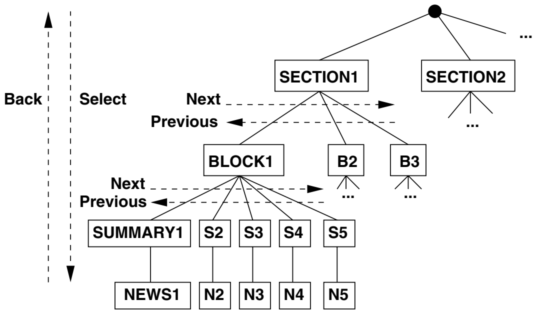


Figure 1: *Browse tree.*

2.1.1. Browse

First, the user selects a section of the newspaper. Next, the system presents the headlines of that section. Then, the user selects one and access the summary of that news. Finally, the user can access the body of the news to get more information.

News in each section are grouped in blocks of 5, to avoid presenting all the news at once. In each block, the user can select one news or go to the following block.

2.1.2. Search

In search strategy, the user selects a section of the newspaper and makes a query. The system prompts the headlines of the news that match that query and the user selects one. The system presents the summary of that news and the user can select to access the body to get more information.

2.2. Information model

The information model contains all the domain information. It is built automatically from the contents of the web site of a local newspaper[1]. In order to allow browse and search strategies, the information model is made up of two elements: a browse tree and search indexes. Before building the information model all the HTML pages are converted into XML using Tidy[2] and XSLT pages.

2.2.1. Tree

Figure 1 shows the structure of the tree that supports the browse strategy. News are organized in 15 sections. Each section has several news, grouped in blocks of five. Each news is structured at three different levels of detail: headline, summary and body. Accessing the information using browse is like moving along the nodes of the tree.

2.2.2. Indexes

An index is a data structure that allows searching information. For each term it has all the documents in which that term appears. To build the index, the vector space model is used, [8]. It represents each document by a vector in the document space. Each dimension of the space corresponds to a term in the document collection (a stemming algorithm can be used to reduce the dimensionality of the space). Given a document, there are several methods to compute the value of each vector coordinate. We used the one called *term frequency-inverse document frequency (tf-idf)*. The following formula is used to compute the weight (w) of each term in the document, where tf is the number of times the term occurs in the document; *df* is the number of documents in which that term appears; and N is the number of documents in the collection:

$$w = (1 + log(tf)) * log\frac{N}{df} \tag{1}$$

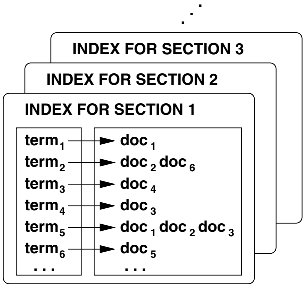
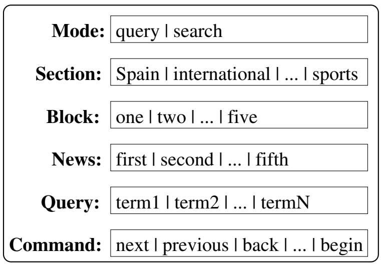


Figure 2: *Search indexes.*



Mode:	query | search
Section:	Spain | international | ... | sports
Block:	one | two | ... | five
News:	first | second | ... | fifth
Query:	term1 | term2 | ... | termN
Command:	next | previous | back | ... | begin

Figure 3: *Frame of information.*

One index is built for each section of the newspaper (figure 2). First, each news story is converted into a vector: all the terms are extracted and a stemmer[3] is used. Next, the weight of each term is calculated using tf-idf. Finally, the index is built using the 25 most relevant components of each news.

To use tf-idf weighting scheme, a document collection is needed. We have collected news stories from that web site during more than a year (71,141 news). With all those stories, we have built dictionaries which give us the document frequency of each term (in how many documents of the collection it appears). A different dictionary is built for each section of the newspaper, in order to obtain more accurate results.

2.3. System architecture

A frame-based approach is used for dialogue management. The user has to provide some items of information to access news. The slot elements can be seen in figure 3. The main advantage of using a frame-based approach is flexibility, because the interaction is not constrained by a predefined control flow.

The architecture of the system can be seen in figure 4. The interaction begins with a welcome VoiceXML page. Then, the user utterance is processed by the speech recognizer. The result is sent to the parser, which extracts the information items and builds a frame. Next, the dialog manager updates the interaction history and selects a template to dynamically build the next VoiceXML page. The VoiceXML generator uses that template, the interaction history and the information model to build the

[1] http://www.nortecastilla.es

[2] http://www.w3.org/People/Raggett/tidy/

[3] http://snowball.tartarus.org

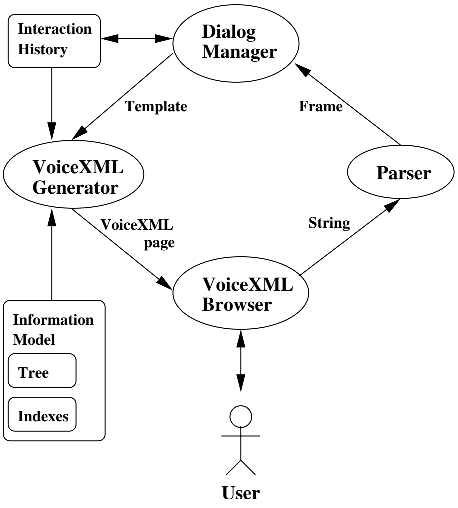


Figure 4: *Architecture of the system.*

next VoiceXML page. The VoiceXML browser interprets that page, prompts a message to the user using TTS and gets the user input using ASR. The result of ASR is given to the parser and the loop continues until the end of the dialog.

The interaction history has two different informations: the information items given by the user and the information about past system and user turns.

To describe the interaction with the user, VoiceXML language is used. Each VoiceXML page specifies the next system message and what the user can say. To build such VoiceXML pages 6 different templates are used. Each of them describes how to build a VoiceXML page using the information model and the interaction history.

To describe the user input ABNF grammars are used, as the VoiceXML standard specifies. There is a general grammar for the interaction and specific grammars for search. Searching grammars are generated dynamically from the contents of the news available in the newspaper (using the indexes from the information model). To parse the strings given by the speech recognizer we have built a LALR parser.

The system uses implicit and explicit confirmation strategies. Confidence measures from the speech recognizer are also used to reject low probability recognition hypothesis. The system also allows barge-in in order to increase the system speed for expert users.

To increase the quality of the speech synthesis, we used markup tags to make pauses and emphasize some words. All the system messages have two versions: the most descriptive one, used the first time the user listens to it and a smaller one used the following times.

The dialog manager, the VoiceXML generator and the parser are implemented using Servlet technology. This allows us to connect them in a standard way with the VoiceXML browser, using HTTP protocol. We are using our own VoiceXML browser which integrates TTS and ASR engines from *Universidad Politécnica de Cataluña* [4] and a *Dialogic* telephone card.

3. Evaluation

We carried out an evaluation of the system to measure its usability and to get some feedback from final users. We obtained objective metrics to measure system performance and subjective metrics to measure user satisfaction.

3.1. Procedure

The system was evaluated by 22 users over the telephone. All users solved 5 scenarios. For each of them, users called the system to get the required information. After that, they provided the gathered information into a web page form. The answers to each scenario were used to calculate task success rate. Once all scenarios were completed, users filled a questionnaire with their perception about the system.

3.2. Performance measures

A log with all the details of the interaction between the user and the system was kept for each call. We transcribed the recorded dialogues and evaluated the answers to each scenario. We used all this information to compute the following objective measures:

- Task success rate: 92%.
- Average duration of the calls: 214.53 seconds.
- User turns per call: 9.54.
- Word Error Rate: 18.09%.
- User utterances rejected because of low confidence: 6.01%.
- System turns barged-in: 66.03%.
- False barge-in (number of wrongly detected interruptions divided by total number of interruptions): 9.04%.
- No-input user turns: 0.75%.
- Help user turns: 0.23%.

The system has a high task success rate. Low no-input and help user turns show that users have a clear idea of what to say in each point of the interaction.

Most of the speech recognition errors happened in search strategy and many of them because the user used a word that was not in the vocabulary. The reason is that news domain has a big vocabulary and it is difficult to constrain the user to the words in the vocabulary of the recognizer.

Users learned quickly to use barge-in to interrupt the system. However, some of them had problems with that, because the system wrongly interpreted breathing as an interruption. This was really critical for 3 users that had false barge-in rate higher than 20%.

3.3. User satisfaction measures

To measure user satisfaction users were asked to complete a questionnaire. We used the one proposed by SASSI [9]. It consists of 34 statements rated on a 7-point Likert scale. With these statements six factors can be computed, each of them accounts for an aspect of the users' perception of the system: system response accuracy, likeability, cognitive demand, annoyance, habitability and speed.

Statements were sorted at random (different for each user) to avoid that results depend on the order of presentation. Some scores were converted in order that high scores in all categories

[4] http://www.verbio.com

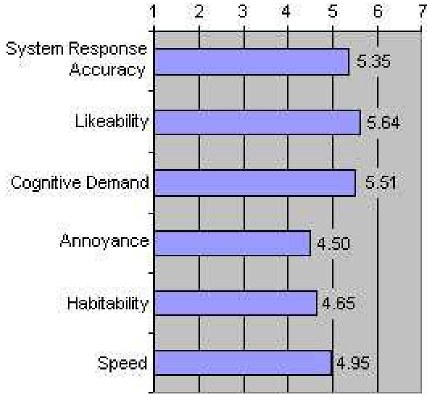


Figure 5: *Factors of user satisfaction.*

are considered good (statements in which a high value is bad were inverted). The resulting factors are shown in figure 5.

There is a positive result of user satisfaction (all factors are higher than 4). Factors with higher value are likeability and cognitive demand, showing that users like the system, think that it is useful, easy to use and does not require a big effort to interact with. Factors with lower value are annoyance and habitability, showing that users think the system is repetitive and boring, and sometimes do not know what to do or what the system is doing.

4. Discussion

Our system allows the use of two strategies to access information: browse and search. Users prefer to use search when they are looking for a specific information, because it is faster. However, when they use words the recognizer can not understand (OOV words) they feel confused, because they do not know what is going wrong. Browse allows users to navigate the contents of the newspaper. Users feel more comfortable because they feel in control of the interaction, although the interaction takes more time.

We have seen in the evaluation that users tend to change from one strategy to the other when errors happen. When users are using one strategy they usually try twice or three times, and if they do not succeed, they change to the other strategy to look for the same information. We have reported some users changing from one strategy to the other twice or three times, until they find the required information or they give up. As a conclusion, having two different interaction strategies can be used as an error recovery mechanism.

Building a system that allows users to access web contents using speech poses some challenges that we had to cope with. First, we had to find some structure in the web contents to allow the use of a spoken dialog system. We organized the news in sections and in blocks inside each section. Each news is divided in three different levels of detail: headline, summary and news. Second, as we decided to use search strategy, we need to dynamically construct grammars for ASR using the contents of the newspaper. We used the vector space model to select the most relevant words. Some problems arose with acronyms and foreign words that our Spanish recognizer was not able to recognize. Third, the output of the system was monotonous and we had to use markup to put pauses and emphasize some words. Acronyms and foreign words were again a problem, since the TTS pronounced them in Spanish.

5. Conclusions

We have presented a spoken dialog system that provides access to a newspaper web site. The system takes the contents from the web and builds an information model, made up of tree and index structures. The interaction in the system is based on two strategies, browse and search. The first allows the users to see what information is available and the second to find specific information. One strategy complements the other if an error happens.

We have evaluated the system to measure its usability. Our results show a high task success rate (92%). We reported that most of the speech recognition errors happened in search strategy, because of out of vocabulary words. Expert users find barge-in very useful to increase speed. User satisfaction study shows that users like the system and think it is useful. However, they also think that it is a bit boring and repetitive.

As future work, we plan to combine more closely both browse and search strategies. It will be useful to make queries while browsing and to browse related news when accessing search results. Also, more work has to be done in ASR for searching, mainly in finding ways to deal with out of vocabulary words. Finally, as users pointed out, system output must be improved making it more pleasant and less monotonous.

6. References

[1] V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T. J. Hazen, and L. Hetherington, “JUPITER: A Telephone-Based Conversational Interface for Weather Information,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, January 2000.

[2] L. Lamel, S. Rosset, J. Gauvain, S. Bennacef, M. Garnier-Rizet, and B. Prouts, “The LIMSI ARISE System,” *Speech Communication*, vol. 31, no. 4, August 2000.

[3] B. Vesnicer, J. Zibert, S. Dobrisek, N. Pavesic, and F. Mihelic, “A Voice-driven Web Browser for Blind People,” in *Eurospeech*, 2003.

[4] S. Goose, M. Newman, C. Schmidt, and L. Hue, “Enhancing Web Accessibility Via the Vox Portal and a Web Hosted Dynamic HTML & VoxML Converter,” in *International World Wide Web Conference*, May 2000.

[5] J. Freire, B. Kumar, and D. F. Lieuwen, “WebViews: Accessing Personalized Web Content and Services,” in *International World Wide Web Conference*, 2001.

[6] J. Polifroni, G. Chung, and S. Seneff, “Towards the Automatic Generation of Mixed-Initiative Dialogue Systems from Web Content,” in *Eurospeech*, 2003.

[7] E. Chang, F. Seide, H. Meng, Z. Chen, Y. Shi, and Y. Li, “A System for Spoken Query Information Retrieval on Mobile Devices,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 8, November 2002.

[8] G. Salton, A. Wong, and C. S. Yang, “A Vector Space Model for Automatic Indexing,” *Communications of the ACM*, vol. 18, no. 11, November 1975.

[9] K. Hone and R. Graham, “Towards a tool for the Subjective Assessment of Speech System Interfaces (SASSI),” *Natural Language Engineering*, vol. 6, no. 3/4, 2000.