A Spoken Dialog System to Access a Newspaper Web Site*

César González-Ferreras†, Rubén San-Segundo-Hernández‡, Valentín Cardeñoso-Payo†

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

‡Departamento de Ingeniería Electrónica
Universidad Politécnica de Madrid, Spain.
lapiz@die.upm.es

Abstract: In this paper we present a spoken dialog system which provides speech access to the information stored in a newspaper web site. The user is allowed to access the contents by query and browse mechanisms. The system is based on an Interaction Model and on an Information Model. The interaction model describes how the interaction with the user is carried out. The information which supports that interaction is described by means of an Information Model. A decision tree and inverted indexes are used depending on the interaction modality chosen by the user.

1 Introduction

Nowadays, there is a huge quantity of on-line information available. The most important repository is Internet, which allows access to contents using a web browser, i.e., visual interaction. However, using speech interaction to access that information would be really useful, because the proliferation of mobile devices which allow Internet access anytime and anywhere, but which have really small displays. Moreover, speech is a modality of interaction which has some advantages over GUI: it is more natural for most of the people, it is more suitable for some environments (eyes-busy) and for some users (blinds).

Current state of the art spoken dialog systems offer a user-friendly interaction using natural language [La99, Zu00]. Those systems are designed to interact with the user in specific tasks, where the way of interaction is well understood. The majority of the systems access structured information stored in a database. Trying to use those dialog systems to access on-line information is really difficult, because information found on the web is, most of the times, free text and lacks the required structure.

Textual information tends to be massive. Displaying it in a visual interface is not a problem, because all the information is presented at once and the user selects the piece he

*This work has been partially supported by *JCyL* under project number VA083/03 and by *MCyT* under project number TIC2003-08382-C05-03.

wants. However, vocal interface is sequential and not persistent, and thus, we have to minimize the information sent to the user. Moreover, the user is not used to interact with a computer using speech to access textual information. We must find an efficient and natural way of interaction.

There are several approaches to make web contents available using speech. Some of them add a vocal interface to an existing web browser, [HT95, Ve03]. Others convert HTML contents into VoiceXML, [Go00, FKL01]. Finally, the solution could be restricted to a limited domain, like in [La97, PCS03], where the dialog system works for selected on-line resources. From the opposite point of view, some traditional Information Retrieval systems have been extended with a vocal interface, [Cr99, Ch02]. However, they emphasize on the search of documents and not on the interaction with the user.

In our work we use two strategies to access information: browse and query. Those strategies are the ones used to access information in the web, and we have adapted them to the vocal interface. Our system is based on an Interaction Model and on an Information Model. The interaction model describes how the system dialogs with the user. The information model describes how the web contents must be processed and structured in order to support that interaction. We have applied our proposal to a newspaper web site.

The structure of the paper is as follows. Section 2 presents browse strategy. Section 3 presents query strategy. Section 4 describes in detail our system and section 5 shows the conclusions of the work.

2 Browse

Browse mechanism is useful when the user does not have a specific information need, and he wants to know which information is available.

2.1 Interaction model

The information must be presented gradually, at different levels of detail. First, the user selects which group of items he wants to access. Then, the items are presented with a headline that describes each option. Once the user chooses an option, the system expands the item description. Then, he can ask for more information about that option or he can go back and choose another option.

2.2 Information model

The information must be organized in groups of items, and all the items in different levels of detail: first a headline, next a short description and finally all the information. If the original information is not organized in this way, we must find an automatic procedure to

process and convert it: clustering and automatic summarization.

The data structure more suitable to such kind of interaction is a decision tree, in which information elements are leafs, and internal nodes are questions to ask the user. When going through the tree, the answers to the questions at each node tells us which descendant node we have to visit next.

3 Query

Query mechanism is useful when the user has a specific information need which he can express as a query. The complexity of the query depends on the system, and it ranges from simple terms to natural language. It depends on the available speech and understanding technology.

3.1 Interaction model

The system searches the text information, and presents the results to the user. The user chooses one result and accesses the information.

3.2 Information model

In order to process the queries efficiently, the use of an index is required. In our case we use an inverted index. An inverted index contains, for each term in the lexicon, a list of documents in which that term appears. We have used the vector space model to build such index, [SWY75].

The vector space model 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 is used to reduce the dimensionality of the space. Given a document, there are several methods to compute the value of each vector coordinate. We have 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:

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

Once we have the documents represented as vectors, we build the inverted index. We only use the terms with high weight values, which are more relevant. That results in a smaller lexicon, which is mandatory because of the limitations of the underlying speech recognition engines.

4 System overview

Our main objective was to build a spoken dialog system to allow users access textual information using speech. The system is based on an interaction model, which describes the interaction using browse and query mechanisms, and on an information model, which structures the information to enable that kind of interaction. We have selected a newspaper domain. We have used VoiceXML as a language to describe dialogs. Our system works for Spanish language.

In the following section, we describe the system architecture in detail. Next, we explain the interaction model and the information model. Finally, a sample interaction is included to show how the system works.

4.1 System architecture

The architecture of the system can be seen in figure 1. The system has 2 main parts: the first one processes all the information and builds the Information Model, and the second one dialogs with the user using the Interaction Model.

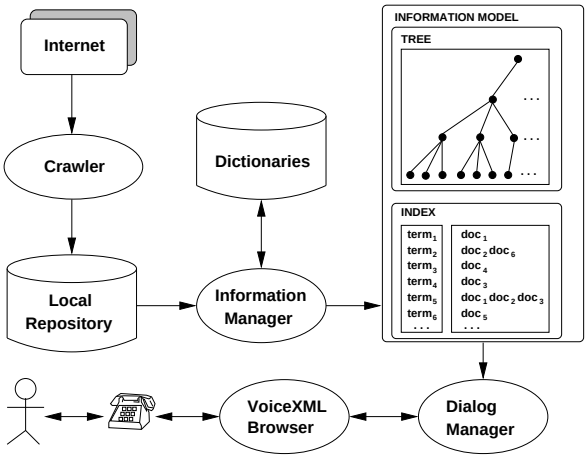

Figure 1: System architecture

First, HTML pages are downloaded from the newspaper web site by a *Crawler*, and stored in a local repository for later use. There are 2 kinds of pages useful for our system: sections and news.

Next, the *Information Manager* builds the Information Model. First all the HTML pages are converted into XML, using Tidy[1] and XSLT pages. With that information the browsing

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

tree is built. Second, the dictionaries are updated to include all the new terms. Finally, using that dictionaries, the inverted index is built.

The *Dialog Manager* reads the Information Model to build VoiceXML pages, which describe how to dialog with the user. We implemented the Dialog Manager as a Java Servlet, in order to communicate it with the VoiceXML browser in a standard way, i.e., using the HTTP protocol. We used Tomcat as servlet container.

The *VoiceXML Browser* dialogs with the user using speech synthesis and speech recognition over the telephone line. The main advantage of using VoiceXML is that the voice applications can be accessed using off-the-shelf technology. In our system, any VoiceXML browser can be used to access the information. We have tested the system using our VoiceXML platform, which is composed of: our own VoiceXML interpreter; speech synthesis and speech recognition engines developed at *Universidad Politécnica de Cataluña*[2]; and a *Dialogic* telephone card.

4.2 Interaction model

The system has two main functionalities: browse and query. Query allows the user to access a specific information, using a query term. On the other hand, browse allows the user to access available information, but without a specific purpose.

We have decided to use a system initiative strategy to control the dialog flow in order to get a higher speech recognition rate. We had a large dynamically generated vocabulary, and our objective was to divide it in several smaller ones, each associated with one state of the dialog. This dialog strategy guides the user during the interaction, and may be preferred by novice users.

The Finite State Diagram used for browse can be seen in figure 2. When the user browses information, the interaction is very similar to going through the decision tree which holds the information (figure 3). First, the user selects the section of the newspaper he wants to access. Next, the system presents all the news in that section, and the user selects one of them. If there are more than five news, they are grouped into blocks. When the user has selected a news, the system presents a short summary of that news, and if the user wants more information, he could access the full news.

The Finite State Diagram used for query can be seen in figure 4. The user first selects a section of the newspaper he wants to access and later specifies a query term. Then, the system searches for the requested information in the inverted index of that section (figure 5). If there is more than one news related to that term, a list of options is presented to the user, who selects one. Like in browse, the user first access to a short summary of the news, and if he wants more information he accesses the full data.

At some points of the dialog, the information given by the system is large. We decided to enable barge-in in order to let the user interrupt the system.

The system uses two different confirmation strategies, depending on the size of the vo-

[2] http://www.atlas-cti.com

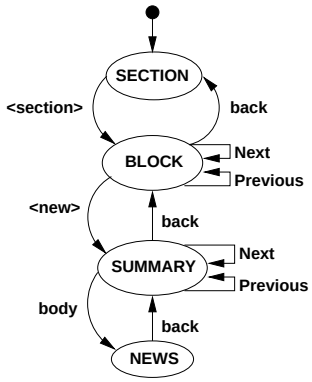

Figure 2: Finite state diagram for browse

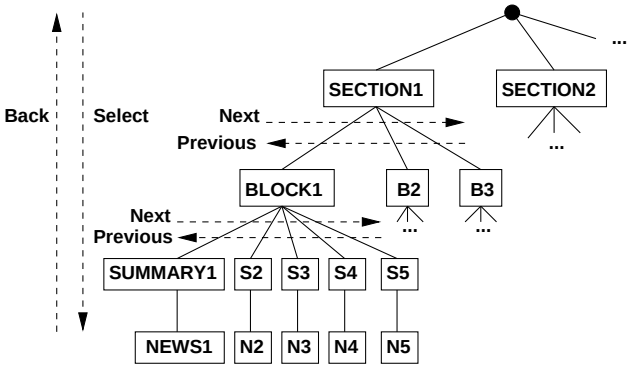

Figure 3: Decision tree

cabulary at a given state. For small vocabularies (less than 25 words) we use implicit confirmation which is faster. For large vocabularies we use explicit confirmation, because the probability of a recognition error is higher.

4.3 Information model

Our information model is built extracting information from the web site of a local newspaper, *El Norte de Castilla*.[3] Contents are updated everyday and we automated the construction of the information model in order to build it daily.

The contents of the newspaper are divided into several sections. Each section contains several news stories. Each news story is composed of several elements: a headline, a short

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

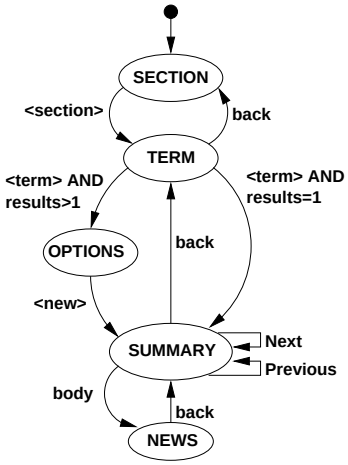

Figure 4: Finite state diagram for query

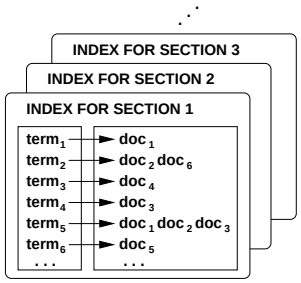

Figure 5: Inverted indexes

summary and a body. Using those structural elements, we build our decision tree. We have news in the leafs of the tree and summaries in the next level. News are grouped into blocks of five elements at most. Finally, a section is composed by several blocks (figure 3).

In order to build the inverted index, each news story is converted into a vector. First, we extract all the terms, next we use the Snowball stemmer[4] and finally we calculate the weight of each term using tf-idf. Using the 25 most relevant components of each news, we build one inverted index for each section of the newspaper (figure 5).

To use tf-idf weighting scheme, we need a document collection. 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, that is, in how many documents of the collection it appears. We have built a different dictionary for each section of the newspaper, in order to obtain more accurate results.

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

4.4 Sample interactions

Figure 6 shows a sample interaction where the user queries for news related to "elections" in the international section. The user selects one of the 3 news found and accesses to the news summary. Figure 7 shows a sample interaction where the user accesses the same news using browse strategy.

System: Welcome to "El Norte de Castilla" newspaper. Remember that you can interrupt the system anytime. You can ask for help whenever you need it. Choose browse or query to access information.
User: Query.
System: What section do you want? local, Spain, international, life, culture or television.
User: International
System: International section. Please, tell me the query term.
User: Elections
System: I understood elections. Is it correct?
User: Yes
System: International Section. There are three news about elections. First: Left parties victory in regional French elections. Second: Karzai announces a delay until September of Afghanistan elections. Third: At least 46 dead in the elections. Please, choose the news you want.
User: Second
System: Headline: Karzai announces a delay until September of Afghanistan elections. International Section. Summary: He justifi es the delay because of the diffi culties to take a census of voters and to convince "lords of war". Please choose body to access the full information or back to return.
User: Exit
System: Thank you for using the system. Good bye.

Figure 6: Sample interaction using *Query* strategy (translated from Spanish)

5 Conclusions

Speech access to Internet contents is really useful, mainly because of the proliferation of mobile devices which allow access to the web anytime and anywhere. However, serious limitations of the voice channel must be overcome in order to deliver textual information to the user using speech.

In this paper we have presented a system which allows speech access to a newspaper web site. The system is based on an Interaction Model and on an Information Model. The interaction model combines browse and query mechanisms in order to allow the user access the information. The information model supports that interaction using two data structures: a decision tree and an inverted index.

All the contents used by the system are automatically obtained from the web, and we process them to build the information model and the grammars used by the recognition engine. The dialog manager uses a system initiative strategy to control the dialog flow, which increases the recognition performance. We used VoiceXML as a language to de-

System: Welcome to "El Norte de Castilla" newspaper. Remember that you can interrupt the system anytime. You can ask for help whenever you need it. Choose browse or query to access information.
User: Browse.
System: What section do you want? local, Spain, international, life, culture or television.
User: International
System: International section. There are ten news. Block one. First: Left parties victory in regional French elections. Second: Tunisia cancels the summit of the Arab League about political reforms. Third: New leader of Hamas in Gaza describes George Bush as "enemy of God and Islam". Fourth: Karzai announces a delay until September of Afghanistan elections. Fifth: United States gives the keys of the first of the twenty five Iraqi ministries. Please, choose the news you want, next or back.
User: Fourth
System: Headline: Karzai announces a delay until September of Afghanistan elections. International Section. Summary: He justifies the delay because of the difficulties to take a census of voters and to convince "lords of war". Please choose body to access the full information or back to return.
User: Exit
System: Thank you for using the system. Good bye.

Figure 7: Sample interaction using *Browse* strategy (translated from Spanish)

scribe dialogs.

As future work, we plan to make an evaluation of the system performance and an usability study. We will study how users respond to the system and this will allow us to validate the adequacy of the Interaction Model proposed to access the information.

References

[Ch02] Chang, E. et. al.: A System for Spoken Query Information Retrieval on Mobile Devices. IEEE Transactions on Speech and Audio Processing. 10(8). November 2002.

[Cr99] Crestani, F.: Vocal access to a Newspaper Archive: Design Issues and Preliminary Investigations. In: ACM Digital Libraries. 1999.

[FKL01] Freire, J.; Kumar, B.; Lieuwen, D. F.: WebViews: Accessing Personalized Web Content and Services. In: International World Wide Web Conference. 2001.

[Go00] Goose, S. et. al.: Enhancing Web Accessibility Via the Vox Portal and a Web Hosted Dynamic HTML & VoxML Converter. In: International World Wide Web Conference. May 2000.

[HT95] Hemphill, C. T.; Thrift, P. R.: Surfing the Web by Voice. In: ACM International Conference on Multimedia. 1995.

[La97] Lau, R. et. al.: WebGalaxy - Integrating Spoken Language And Hypertext Navigation. In: European Conference on Speech Communication and Technology (Eurospeech). 1997.

[La99] Lamel, L. et. al.: The Limsi Arise System For Train Travel Information. In: International Conference on Acoustic, Speech and Signal Processing (ICASSP). 1999.

[PCS03] Polifroni, J.; Chung, G.; Seneff, S.: Towards the Automatic Generation of Mixed-Initiative Dialogue Systems from Web Content. In: European Conference on Speech Communication and Technology (Eurospeech). 2003.

[SWY75] Salton, G.; Wong, A.; Yang, C. S.: A vector space model for automatic indexing. Communications of the ACM. 18(11). November 1975.

[Ve03] Vesnicer, B. et. al.: A Voice-driven Web Browser for Blind People. In: European Conference on Speech Communication and Technology (Eurospeech). 2003.

[Zu00] Zue, V. et. al.: JUPITER: A Telephone-Based Conversational Interface for Weather Information. IEEE Transactions on Speech and Audio Processing. January 2000.