A voice e-mail client based on natural language

Leandro Rodríguez Liñares[1], Carmen García Mateo[2], and Víctor Darriba Bilbao[1]

[1] E.T.S.E Informática
Universidade de Vigo
32004 - Ourense (SPAIN)
[2] E.T.S.E Telecomunicación
Universidade de Vigo
36200 - Vigo (SPAIN)

Abstract. Last years have witnessed the emergence of a new class of human-computer interfaces that combines serveral human language technologies to enable humans to converse with computers using speech. In this paper, we describe one of these system, devised for accessing an e-mail server by means of natural language.

1 Introduction

Over the last years, it has become more and more usual to interact with automatic informational systems using some kind of speech interface[1]. In such interfaces, three distinct technologies are involved: speech recognition, text-to-speech conversion and dialogue control.

Many speech based interfaces can be considered conversational in the sense that there is an actual conversation between the system and the user. These systems can be classified depending on the role played by the system in the conversation. We have *system initiative* dialogue systems: in this case the interaction is fixed and the user must respond some questions made by the system. On the other side we have *user initiative* systems. In this case, the system is passive: it only asks when it needs additional information. The "how can I help you?" systems fall into this category. In these interactions, the user sentences do not have any predefined structure and the natural language understanding is a key aspect. Between these extremes, we have *mixed initiative* systems, where both parts take an active role in order to sucessfully complete the conversation.

In this paper, a voice e-mail client that run on a Linux-based PC is presented. It is based on an *user initiative* system and its development involved the adaptation of a Text-to-Speech Converter and a Voice Recognizer. Both speech tools are bilingual (Spanish/Galician) and make use of the sound card as input/output device although it would be easy to adapt the system to use a telephone control card. All the examples shown in this paper are in Spanish, but translation is provided to English in all cases.

The rest of this paper is structured as follows: first, a general description is made from a functional point of view. Then, the modules that integrate the

system are presented. In section 4 a deep description of the natural language analysis module is included. In the last section, conclusions and future improvement are discussed.

2 General Description

As stated in the introduction, the system presented in this paper was designed to allow the users to consult an e-mail account using natural language. Its structure can be seen in fig. 1. The system accepts as input sentences obtained by means of a Speech Recognizer. The output consists of the requested information synthesized by a Text-to-Speech Converter. These tools (Speech Recognizer and Text-to-Speech Converter) were previously developed and they were adapted to work within the system presented in this paper. Both tools are described in sections 3.2 and 3.3, respectively.

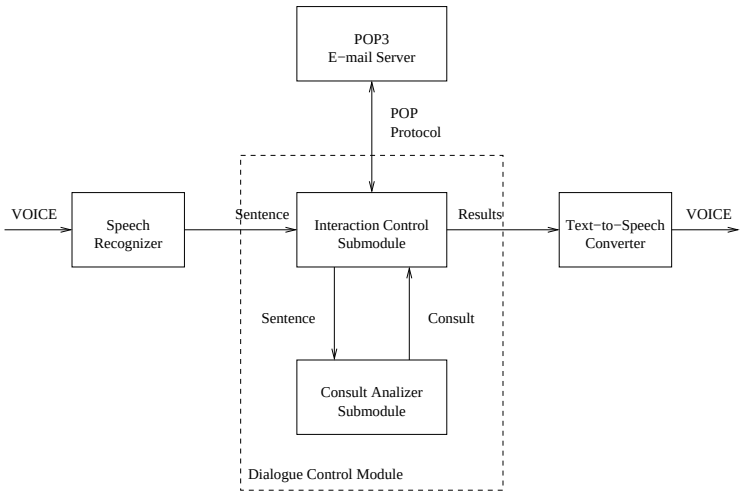

Fig. 1. Architecture of the system

The kernel of the e-mail client is the *Dialogue Control Module*, that can be divided up into the *Interaction Control Submodule* and the *Consult Analizer Submodule*.

The goal of the system is to be able to access an e-mail server to perform some of the following tasks:

- To mark some messages.
- To consult if there are new messages and to count them.
- To ask for some characteristics of the messages like date, sender or subject.
- To read some of the messages of the mailbox.
- To delete some of the messages of the mailbox.

In each case there are different ways to select the messages to consult, read, count, mark or erase. That is, there are distinct criteria to filter the messages:

- By their position in the messages list.
- Asking for the N first or last messages.
- Specifying the sender.
- Specifying the date of the messages. It is possible to request the messages sent in a certain date or before or after a specific day.
- Asking only for new or read messages.

Every moment, a list of the messages in the mailbox is mantained, and the system puts marks on it depending on the wishes of the user. Then, the actions will be applied over the marked messages. This allows the user to select messages in several steps, which is very useful in case the user maintains many messages in the mailbox.

Besides, the criteria can be combined. It is possible, for example, to ask for the last message of a specific sender or the first message received in one day.

In section 4.3 an example of an interaction is included.

3 Architecture of the System

In this section the functions and main details of the modules of the system are presented. These modules can be seen in fig. 1.

3.1 Dialogue Control Module

This module is the core of the system. Their functions include the interaction with speech tools (Speech Recognizer and Text-to-Speech Converter) and the E-mail server.

Interaction Control Submodule The input of this submodule consists of the text recognized by the Speech Recognizer. The sentence is delivered in text format to the Consult Analizer Submodule, which responds with the information that has to be requested to the POP server. From this information, the Interaction Control Submodule plays a POP3 client role and obtains the requested fields of the specified messages. These fields are then delivered to the Text-to-Speech Converter, that "reads" them to the user.

In case the Analizer is not able to parse the recognized sentence, the Text-to-Speech Converter will synthesize an error message.

This submodule was developed using C programming language, and it includes a communication interface with a Prolog [2,3] interpreter. This is necessary because the Consult Analizer Submodule, explained in the following section, was programmed in Prolog.

Consult Analizer Submodule This submodule receives the sentence that has to be parsed as a Prolog list. The interface of the Prolog program was implemented as a predicate consulta/6. It has one input term and five output terms:

consulta(+A, -Accion, -Indicador, -Numero, -Remite, -Fecha)

A contains the sentence to analize, while the rest of the variables will be instatiated depending on the result of the parsing:

- Accion *(Action)* is the action that must take place. It can be contar *(count)*, marcar *(mark)*, leer *(read)*, borrar *(delete)*, consultar_remite *(obtain_from)*, consultar_fecha *(obtain_date)*, or consultar_tema *(obtain_subject)*.
- Indicador *(Indicator)* will contain the selection criteria for the messages. Some possible values are, among others, todos *(all)*, número *(number)*, leídos *(read)*, remite *(sender)*, fecha_desde *(date_from)* or fecha_antes *(date_before)*. This criteria can be combined using the "+" operator. For example, if the sentence refers to the messages sent by a specific sender in a specific date, Indicador *(Indicator)* will contain remite+fecha *(sender+date)*.
- Número *(Number)* is the number or the position of messages to process.
- Remite *(Sender)* is the sender of the messages the user wish to process.
- Fecha *(Date)* is used in case the user wants to specify the date of the messages to process. In the current version of the system, the parsing capabilities were limited in such a way that this variable can only contain the values hoy *(today)*, ayer *(yesterday)*, antesdeayer *(the day before yesterday)*, and from lunes *(monday)* to domingo *(sunday)*. This way, the limit to specify a message using the date is one week.

If some field is not filled in the parsing, it will contain the string null.

The system is designed to use more than one filtering criterium. In this case, the parameters of the distinct criteria will be combined. In table 1 some examples of such combinations can be seen.

3.2 Speech Recognizer

The Speech Recognizer used in the presented system is studied in depth in [4]. It is based on a recognition engine that uses 25 phoneme models. These models are HMM's (Hidden Markov Models) [5] trained using a telephone voice database. Later, the models were adapted to be used in recognizing voice obtained from the PC sound card. The HMM's are left-to-right models with 3 states and 16 mixtures/state.

The voice is sampled at 8 KHz. Energy and 12 mel-cepstra coefficients are obtained using a 20 msec. window with a shifting of 10 msec. Liftering (factor 22) is applied and the first and second derivatives of the coefficients are appended to the parameters. The vector length thus obtained is 39 each frame.

In order to get some equalization, mean substraction is applied to the vectors. For calculating the mean, a certain number of vectors is needed for accuracy reasons. However, this implies that in an online environment the Speech Recognizer is not going to work properly until some voice has been processed.

Table 1. Examples of consults analysis

Sentence	Acción *(Action)*	Indicador *(Indicator)*	Número *(Number)*	Remite *(Sender)*	Fecha *(Date)*
"Borra los dos primeros mensajes de Manuel"	borrar	remite+primeros	2	Manuel	null
(Delete the first two messages from Manuel)	*(delete)*	*(from+first)*	*(2)*	*(Manuel)*	*(null)*
"Lee los mensajes de Manuel recibidos hoy"	leer	remite+fecha	null	Manuel	hoy
(Read the messages from Manuel received today)	*(read)*	*(from+date)*	*(null)*	*(Manuel)*	*(today)*
"Dime el asunto del primer mensaje nuevo"	consultar_tema	posición+noleídos	1	null	null
(Tell me the subject of the first new message)	*(obtain_subject)*	*(position+unread)*	*(1)*	*(null)*	*(null)*

The system works as follows: the speech recognizer is listening to the sound card and applying a VAD (Voice Activity Detector) to detect when the user starts to talk. This VAD is based on energy thresholds. When voice is detected, the Speech Recognizer starts to calculate and store parameter vectors until the end of the user's sentence is detected by the VAD. In this moment, recognition takes place, and the recognized sentence is transmitted to the Interaction Control Submodule. Then, the Speech Recognizer will go into inactive state until the Interaction Control Submodule allows it to start listening again.

3.3 Text-to-Speech Converter

The used Text-to-Speech converter was *Cotovía* [6], that is a bilingual Spanish/Galician speech synthesizer based on concatenation of units. It can synthesize masculine or feminine wide band voice (8 KHz). Pitch and articulation rate can be configured by the user.

Cotovía works as a module independent from the rest of the system. It integrates all the facilities necessary to work as an isolated application. Among this facilities, its preprocessing capabilities play a key role in the system.

4 Natural Language Parsing

In this section basic aspects of the natural language parsing are addressed: language model generation and specification of the senders of the messages. Besides, an example of interaction is provided.

4.1 Language Model Generation

The Speech Recognizer makes use of a language model that is based on the ARPA format. This format is very similar to the one used by the HTK speech recognition tools [7]. The language model consists on a lexicon and a set of unigram, bigram and trigram probabilities. From the lexicon and the probabilities set, a recognition network is constructed. This network is used by the recognition engine, that applies Viterbi algorithm to find the most likely sentence in the utterance given the model.

Due to the probabilistic nature of the language model, a language corpus is necessary in its estimation procedure. The language corpus has to be large enough to get an accurate estimation. From the corpus, the lexicon and the n-grams probabilities are obtained.

The best recognition results are achieved if the corpus is adapted to the recognition task. That is, it should contain a set of sentences that is a good representation of the phrases the Speech Recognizer will have to cope with. In our system, the natural language parser was implemented using Prolog. This makes very easy transforming the parser into a correct sentence generator. In this case, "correct" is used in the sense of "agrees with the grammar defined by the parser".

The steps taken in the language model estimation process are presented in the following:

1. An informal definition of the grammar for the consults was made. This grammar contains the usual gramatical structures for asking about e-mail information. For example, léeme los mensajes... *(read the messages...)*, ¿podrías decirme...? *(could you tell me...?)*, quiero que cuentes los correos... *(I want you to count the mails...)*.
2. A parser for this grammar was implemented in Prolog using DCG (Defined Clause Grammars). Some of the parser rules are included in table 2. The shown rules are slightly different from the ones actually present in the parsing grammar: they have been modified only to be more easily understood. The concordance in number and genre was taken into account to avoid the generation of a huge corpus.
3. The parser was modified into a correct sentence generator. With such generator, a corpus of correct sentences was created.
4. The language model was estimated. For this purpose, the set of software tools described in [8] was used.

4.2 Address Book

Although in a complex system like the presented one there are plenty of implementations details, the Address Book is one specially important aspect. The reason behind this is that the sender is a field that the user specifies quite often to select messages from the inbox. However, the "From:" field has a very complex structure, and the only required string is the e-mail address of the sender.

Table 2. Some examples of parsing rules

frase → frase_verbal. *(phrase → verbal_phrase.)*
frase_verbal → verbo_desiderativo, verbo, artículo, nombre_mensajes, condición. "quiero contar los mensajes..." *(verbal_phrase → desiderative_verb, verb, article, messages_word, condition.)* ("*I want to count the messages...*")
condición → ['enviados'], ['por'], Remite. "... enviados por Manuel" *(condition → ['sent'], ['by'], Sender.)* ("*... sent by Manuel*")
condición → ['recibidos'], ['antes'], ['del'], Fecha. "... recibidos antes del viernes" *(condition → ['received'], ['before'], Day.)* ("*... received before friday*")

The system will be able to select the messages from a sender only if it can establish some equivalence between the name present in the sentence (an alias) and the e-mail address included in the "From:" field of the e-mail header. To solve this problem, the system includes an Address Book that stores the correspondence between the alias and the electronic address.

4.3 An example of interaction

An example of interaction between the system and the user is included in table 3. In the table, U: stands for "user", while S: is for "system". The scenario is as follows: the user has 20 messages in the mailbox, 8 of them sent by Manuel. He/she received 10 messages today, and 2 of them are from Manuel. The user is only interested in an e-mail sent today by Manuel whose subject is "Noticias importantes" *(Important news)*.

The interaction is quite similar to the one kept with a human operator. The system understands natural sentences and several commands like inicia *(init)*, sí/no, *(yes/no)* and a few others.

5 Conclusions and future improvement of the system

In this paper, a speech based e-mail client is presented. The system is user initiative, and Prolog was used for parsing the user's sentences. A similar application for web-searching was also developed. This proves the flexibility of this structure, which makes possible to adapt it to perform other remote access tasks.

Upgrading and updating the system is quite simple, due to the design of the Consult Analizer Submodule. New words can be easily added to the lexicon, and new sintactic structures or combinations of the old ones can be included in the parsing grammar. However, the generation process of the language model

Table 3. An example of interaction

S: Hay 20 mensajes, 0 marcados *(There are 20 messages, 0 marked)*
U: Marca los mensajes recibidos hoy *(Mark the messages received today)*
S: Marcados 10 mensajes *(10 mesagges marked)*
U: Inicia *(Init)*
S: Hay 20 mensajes, 0 marcados *(There are 20 messages, 0 marked)*
U: Marca los mensajes enviados por Manuel *(Mark the messages sent by Manuel)*
S: Marcados 8 mensajes *(8 mesagges marked)*
U: Lee los asuntos de los mensajes recibidos hoy
(Read the subjects of the messages received today)
S: Tienes 2 mensajes enviados hoy por Manuel
(You have 2 messages sent today by Manuel)
S: Mensaje número 1. Asunto: saludos *(Message number 1. Subject: greetings)*
S: Mensaje número 2. Sin asunto *(Message number 2. No subject)*
S: Hay 20 mensajes, 2 marcados *(There are 20 messages, 2 marked)*
U: Borra los mensajes *(Delete the messages)*
S: ¿Estás seguro? *(Are you sure?)*
U: Sí *(Yes)*
S: Borrados. Hay 18 mensajes, 0 marcados
(Deleted. There are 18 messages, 0 marked)
...

is far from being comfortable. The generation of the language model from the grammar is an aspect where we are working in this very moment.

One important drawback of the system is that the senders' aliases have to be included in the parsing grammar. Then, adding a new alias implies repeating the whole process of language model estimation. New versions of the system will allow to introduce or remove automatically new aliases in the language model when the user wishes to modify the address book.

One important aspect of the dialogue systems is the naturalness of the interaction. Future versions of the system will include the following new features:

- Both the Speech Recognizer and the Text-to-Speech Converter are independent modules. This means that while they access the sound card, they do not allow other applications to use it. This implies that "barge-in", (interruption of the Text-to-Speech Converter by the user) is not possible. The interaction between the Interaction Control Submodule and the speech tools has to be improved to make "barge-in" possible.
- In the present state of the system, the parser only accepts correct sentences according to the grammar. It would be very useful if the parser was able to identify which part of the sentence is not correct and ask the user about this particular information.

The voice e-mail client here presented is under constant revision, and naturalness of interaction is an aspect where we are working hard. If this paper is

accepted, its final version will include new improvements that will be included in the system. A practical exhibition will be made in the conference as well.

References

1. Victor W. Zue and James R. Glass: Conversational Interfaces: Advances and Challenges. Proceedings of the IEEE (2000) 1166–1180
2. M. Vilares, M. Alonso and A. Valderruten: Programación lógica. Ed. Tórculo (1998)
3. Michael A. Convington: Natural Language Processing for Prolog Programmers. Prentice Hall (1994)
4. Antonio Cardenal López: Realización de un Reconocedor de Voz en Tiempo Real para Habla Continua y Grandes Vocabularios, (Ph. D. thesis). Dpto. de Tecnoloxías das Comunicacións, ETSE Telecomunicación, Universidade de Vigo, SPAIN (2001)
5. L. R. Rabiner: A Tutorial on Hidden Markov Models and Selected Applications i n Speech Recognition. Proceedings of the IEEE (1989) 257–285
6. Xabier Fernández and Eduardo Rodríguez: Segmental Duration Modelling in a Text-to-Speech System for the Galician Language. Proceedings of the EuroSpeech (1999)
7. Steve Young: Hidden Markov Model Toolkit (HTK). http://htk.eng.cam.ac.uk (2002)
8. Ronald Rosenfeld and Philip Clarkson: CMU-Cambridge Statistical Language Modelling Toolkit v2. Carnegie Mellon University, Cambridge University (1996)