The first Voice Recognition Applications in Russian Language for use in The Interactive Information Systems

V.A. Zhozhikashvili, M.P. Farkhadov, N.V. Petukhova (1), A.V. Zhozhikashvili (2)

(1) Institute of Control Sciences RAS, 65 Profsoyuznaya Street, Moscow GSP-4, Russia, 117997. mais@ipu.rssi.ru, serena@ipu.ru

(2) Institute for Information Transmission Problems RAS, 19 Bol. Karetnyi per., Moscow GSP-4, Russia, 101447. zhozh@tais.ru

Abstract

In the article is discussed the problem of the access of broad masses of the population to information. The use of speech recognition in the interactive information systems is the most suitable way to make these systems more open and easily available for people.

The purpose of our work was to create the functioning speech interfaces in Russian language to information and service systems and to analyze the results.

Our efforts resulted in creation of the first voice servers for interactive information and services systems in Russia: computer reservation system Sirena, a system for dispatching of orders for taxi delivery, and a banking system.

We developed several analytical models that allow dialogue optimization and control. One of our models, which enable the calculation of the mean number of retries and appropriate time losses in different conditions, is presented in this paper.

1. Introduction

The authors of this paper for a long time took part in the creation and development of Russian computer reservation system “Sirena” for Airlines.

The large American and European airlines including USSR airline Aeroflot were the first companies who used computers for automated control of resources and booking. SABRE for American Airlines was the first Computer Reservation System (CRS) in USA. In the USSR the creation of the first system of such type was started in the 1960s at the Institute of Control Sciences which was later appointed the head organization for the whole project named "Sirena".

A distinguishing feature of CRS is the existence of the central database of available resources and rapidly changing information and the large number of remote terminals. These terminals are used by agents for access to resources and servicing client information.

For a long time CRS and other similar systems functioned in such mode.

But the necessity of special terminals was a disadvantage for CRS. Although there were more and more of these terminals throughout the world the methods of operation were the same. And only the online access to information from personal computers through Internet changed the situation. Now speech recognition technology allows even easier and more comfortable interaction of clients with CRS and other application systems.

This is the reason why we are interested in using speech recognition technology in systems that provide information and services to people in Russian, and especially in functioning information and service systems.

We believe that the use of speech recognition in the interactive information systems is the most suitable way to make these systems more open and easily available for people.

2. Purpose and subject of work

The purpose of our work was to create the functioning Russian language speech interface for information and service systems and to analyze the results.

For research purposes we have chosen both currently working and proposed technological systems: the airline computer reservation system "Sirena", a system for dispatching taxi services and some others. We also created a laboratory system with special functions for researching the characteristics of base speech recognition software and for defining the influence of different parameters on recognition and understanding of speech.

As base speech recognition platforms we used Speech Pearl of ScanSoft, Inc [1] and speech recognition software of Nuance Communication, Inc [2].

For each technological system we studied user demands on services, technological features of application and ways of access to information.

On the basis of these studies we designed scenarios, created interfaces with application systems and programmed dialogues.

In the process of this work we created a lot of Speech Blocks in Russian language. Among them are Speech Blocks of numbers, dates, times of day, telephone numbers, cities, Moscow streets along with many others, most of which can be broadly used in various applications.

A significant collection of audio files was created.
A classification of types of interaction between voice servers and technological systems was done. We implemented three types of interaction with application systems: specialized real time interface, access via a Web site and interaction via e-mail.

Also, we developed several analytical models that allow dialogue optimization and control. One of our models which enables the calculation of the mean number of retries and appropriate time losses in different conditions, is presented in this paper.

Our efforts resulted in creation of the voice servers for interactive information and services systems: computer reservation system "Sirena", a system for dispatching of orders for taxi delivery, a banking system and others. These are the first full-function systems in Russia with voice access to information and services.

The demo versions of developed systems are accessible at any time.

3. The creating of speech applications for the example of developing of voice interface with "Sirena".

Introducing speech applications should be planned as any other new system development project. However certain aspects of such project are unique for voice-enabled systems.

The development of voice-enabled applications includes four main phases:
- analysis of application technology and determination of functions,
- scenario design,
- programming and creation of audio files,
- testing and tuning.

In the first phase of design we analyzed the clients requests for information from the "Polyet-Sirena" database [3] and defined the most numerous ones. Answering these requests without using agents frees agents from many routine requests and gives company the most revenue by introducing the voice user interface.

We realized that the requests about flights between two cities including requests about actual flight information are the routine requests described above. The service of these types of requests was chosen as the base function of our system.

At the second phase of project we developed scenarios of client - computer interaction. We analyzed some alternative variants of dialog distinguished by number of states, by places of implementation of confirm and correct states and so on. It was defined that it was useful at that phase to apply analytical models for optimization of dialog duration in different conditions and limitations.

As result of this phase we established the dialog construction. It included playing the result of recognizing and asking to confirm and possible to correct the whole input information.

At that phase persona was chosen. Our preference was for a female voice.

The third phase involved writing out the codes, development of Speech Blocks, recording of audio files and creating of interface with "Polyet-Sirena" database.

We found during our research that functioning systems, and especially systems with long history such as Sirena, have specific protocols and formats of input and output messages. On the one hand this feature of protocols and formats makes interaction with these systems difficult, but on the other hand these formats are usually well-structured, which makes programming of interfaces with these systems easier.

"Polyet-Sirena" input massage has a well-defined request form that includes headers, addresses and other protocol information and a number of fields with actual information, inquired by client: request type, code of departure city, code of arrival city, date of departure, preferable time of departure, preferable criteria of searching of flights. These fields are filled on the basis of information, receiving during dialog and coded with accordance of "Polyet-Sirena" language. Input massage is sent to database of "Polyet-Sirena" via the Internet. Sending input message and receiving answer are performed in real time and telephone connect with client is not disconnected. Just in case computer notifies client of sending his request to database and asks to wait a little.

The output message has also form of "Polyet-Sirena" and is formulated in its language. Computer analyses it, extracts the data about flights and stores them in a set of variables. This data is used for generating prompts informing client about flights.

For optimization of dialog dynamic control of dialog progress can be used. For example it can be used to change the sense of correct-and-confirm prompts as a function of phase of dialog when an error was detected and also what sort of error occured. In exactly the same way computer can change the dialog progress on phase of results playing according with the number of flights or client's preference to get at first the information about morning flights, for example.

The fourth phase of project included testing and tuning of system. Dozen people with different voices from different telephones including wireline phones and wireless handsets took part in testing.

Testing detected some errors which were corrected and demonstrated a number of regularities some of which we used for improving quality of system. In particular an asymmetry of wrong recognition of some Russian words was found. It is that there are some pairs of words X and Y where word X often is recognized as word Y from the same grammar but word Y practically never is recognized as X. For such pairs of words algorithmic methods can be used to correct possible errors and get true result. In other cases there was observed misunderstanding of some words regardless of testers. Such errors were corrected by editing of user lexicon.

Functional testing confirmed the system was ready to deployment.
4. Analytical model for calculation of the mean number of retrievals

Let's have a number of n words - \( s_1, s_2, \ldots, s_n \). Computer tries to recognize a word and offers its hypothesis of recognition.

There are three situations possible:
1. Computer recognized a word correctly;
2. Computer recognized a word incorrectly;
3. Computer informed that it couldn't recognize a word (let's denote this message \( s_0 \)).

It is possible to estimate experimentally the probabilities: \( p_i, i = 1, 2, \ldots, n \) - the probability, that word \( s_i \) was uttered; \( q_{ij}, i = 1, 2, \ldots, n, j = 1, 2, \ldots, n \) - the probability, that word \( s_i \) was recognized as word \( s_j \). In particular \( q_{ii} \) - the probability, that word \( i \) was recognized correctly, and \( q_{00} \) - the probability, that word \( i \) was not recognized. Of course \( \sum_{i=1}^{n} p_i = 1 \) and \( \sum_{j=0}^{n} q_{ij} = 1 \) for all \( i \).

It is possible to calculate inverse probabilities, \( r_{ij}, i = 1, 2, \ldots, n, j = 0, 1, 2, \ldots, n \) - the probability, that word \( s_i \) was uttered and computer recognized it as word \( s_j \). In particular \( r_{00} \) is the probability, that computer couldn't recognize a word. To calculate \( r_{ij} \) note that the probability that word \( s_i \) was uttered but computer recognized it as \( s_j \) is equal \( t_j = \sum_{k=1}^{n} p_k q_{ij} \). In accordance with formula of the conditional probability we can find \( r_{ij} = \frac{p_i q_{ij}}{\sum_{k=1}^{n} p_k q_{ij}} \). By the way these probabilities can be defined experimentally too.

Let's consider the following algorithm of computer behavior (algorithm 1).

Let's say that computer recognized an uttered word as \( s_j \). Computer plays prompt presenting recognized word and asks client whether the word was recognized correctly. If the answer is "yes" procedure is ended. Otherwise computer recalls all probabilities \( r_{ij}, r_{ij}, \ldots, r_{nij} \), except for \( r_{ij} \), and picks out the highest one. Let it be \( r_{ij} \). Computer assumes that word \( s_i \) was uttered because it had the highest probability to be uttered. Computer asks client again whether word \( s_i \) was uttered. If it is again not confirmed computer finds the second by value probability \( r_{ij} \) and assumes that the word \( s_i \) was uttered, and so on.

Let's calculate the average number of retrievals for such algorithm. To simplify formulas and avoid double indexes let's enumerate words. Let's mark a word that was uttered according to computer as number 1, a word that could be transformed to \( s_1 \) with maximum probability as number 2, next by probability word as number 3, and so on. Thus word numbering is such that \( r_{11} \geq r_{21} \geq \ldots \geq r_{n1} \). (The first inequality means that if computer recognized a word as \( s_1 \), this word was most likely uttered otherwise the recognition system is disabled!). Thus computer would offered words for confirming to client in the following order: \( s_1, s_2, s_3 \) and so on. With probability \( r_{11} \) there would be exactly one retry - \( s_1 \), because hypothesis of computer is correct. With probability \( r_{21} \) there would be two retries - \( s_1 \) and \( s_2 \), with probability \( r_{31} \) there would be three retries - \( s_1, s_2 \) and so on. Thus the mean number of retrievals \( l \) would be obtained via formula \( l = l \cdot r_{11} + 2 \cdot r_{21} + 3 \cdot r_{31} + \ldots + n \cdot r_{ni} \).

The same algorithm is used if computer can not recognize a word. In this case computer finds maximum probability from \( r_{10}, r_{20}, \ldots, r_{n0} \) and if it would be \( r_{00} \) for example computer offers word \( s_1 \) as the most probable one. If client doesn't confirm it computer finds the next by value probability, and so on. If we again enumerate words in such order that \( r_{10} \geq r_{20} \geq \ldots \geq r_{n0} \) we will have the same formula for the mean number of retrievals \( l \): \( l = l \cdot r_{10} + 2 \cdot r_{20} + 3 \cdot r_{30} + \ldots + n \cdot r_{n0} \).

The mean number of retrievals calculated via this method will be the specific value for each recognized word (it isn't seen from formula because word number is not presented in formula explicitly but enumeration of words depends on specific word and, therefore, the result of calculating depends on the word).

Let's try to write all formulas not using explicit enumerating words. Say, computer recognized a word as \( s_j \). Let order the probabilities \( r_{ij}, r_{ij}, \ldots, r_{nj} \) in order of descent. As the result we have the sequence \( r_{m_{1j}} \geq r_{m_{2j}} \geq \ldots \geq r_{m_{nj}} \). Thus \( m_{ij} \) is the number of word that was uttered with the most probability on condition that computer recognized a word with number \( j \). (we remind that as a rule \( m_{ij} = j \) because if computer recognized some word, just this word was uttered with the most probability. By the way this is not important for us). \( m_{ij} \) is the number of word that has the second by value probability, again on condition that computer recognized a word with number \( s_j \), and so on. Now formula for \( l \) is:

\[
l = 1 \cdot r_{m_{1j}} + 2 \cdot r_{m_{2j}} + 3 \cdot r_{m_{3j}} + \ldots + n \cdot r_{m_{nj}} = \sum_{k=1}^{n} k \cdot r_{m_{kj}}.
\]

Now we can write the formula for the mean number of retrievals. Let's remember, the probability \( t_j \) that computer recognized word as \( s_j \) is equal \( t_j = \sum_{k=1}^{n} p_k q_{kj} \). Thus the mean number of retrievals is calculated by formula

\[
l = \sum_{j=1}^{n} t_j l_{1j} = \sum_{j=1}^{n} \left( \sum_{k=1}^{n} p_k q_{kj} \sum_{k=1}^{n} k \cdot r_{m_{kj}} \right).
\]
Numerical example

This is an example of calculation of the mean number of retries if computer has to recognize uttering words that mean numerals from 0 to 9. In this case grammar consists of ten words in Russian language. Assume probabilities are equal, that is
\[ p_i = p_j = \ldots = p_9 = 0.1. \]
The probabilities \( q_{ij} \) were obtained experimentally. They are presented in Table 1.

<table>
<thead>
<tr>
<th>( q_{ij} )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.98</td>
<td>0.02</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.02</td>
<td>0.98</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.02</td>
<td>0.02</td>
<td>0.86</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.12</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0.94</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.06</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.02</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.96</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0.04</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0.04</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.94</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.96</td>
</tr>
</tbody>
</table>

\( q_{ij} \) in the line \( i \) and column \( j \) is the probability that word \( i \) was recognized as word \( j \).

Let's define probabilities \( r_{ij} = \frac{p_i q_{ij}}{\sum_{k=1}^{n} p_k q_{kj}} \). In this case, because all probabilities are equal, this formula can be written as \( r_{ij} = \frac{q_{ij}}{\sum_{k=1}^{n} q_{kj}} \). Obtained values are presented in Table 2.

<table>
<thead>
<tr>
<th>( r_{ij} )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.98</td>
<td>0.02</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.02</td>
<td>0.88</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.02</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0.02</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.05</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.85</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.15</td>
<td>1</td>
<td>0.04</td>
<td>0</td>
<td>0.02</td>
</tr>
<tr>
<td>7</td>
<td>0.03</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.92</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0.05</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.04</td>
<td>0</td>
<td>0.83</td>
</tr>
</tbody>
</table>

Now we can calculate the mean number of retries. Obtained values are presented in Table 3 (algorithm 1).

<table>
<thead>
<tr>
<th>( \bar{r} )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.02</td>
<td>1.25</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1.15</td>
<td>1</td>
<td>1.12</td>
<td>1</td>
<td>1.26</td>
<td></td>
</tr>
</tbody>
</table>

In upper line there are hypothesis of computer about uttered word, in lower one - the mean number of retries. That is if computer recognized a word as \( i \) the according mean number of retries would be equal to a value presented in the second line.

For comparison let's do the same calculation for the simpler algorithm (algorithm 2). Assume that computer plays numeral after recognition and asks to inform it whether uttered word was recognized correctly and to repeat a word if recognition was wrong. In this case the mean number of retries is equal \( 1/q_i \) if a word \( s_i \) was uttered. The mean numbers of retries for this algorithm are presented in Table 4.

<table>
<thead>
<tr>
<th>( \bar{r}_{ij} )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.02</td>
<td>1</td>
<td>1.02</td>
<td>1.16</td>
<td>1.06</td>
<td>1</td>
<td>1.32</td>
<td>1.04</td>
<td>1.06</td>
<td>1.04</td>
<td></td>
</tr>
</tbody>
</table>

In contrast to Table 3 the upper line of Table 4 consists of real uttered words but not hypothesis of computer about uttered ones.

The mean number of retries for algorithm 1 is \( l = 1,08 \), for algorithm 2 is \( l = 1,072 \).

5. Conclusion

With the rapid advances in speech recognition technology many companies in USA, Europe, Japan and other countries began to use voice-enabled solutions in their practice. Using speech recognition allows company to reduce costs, to improve customer service and to generate additional revenue.

Our research demonstrates that now the time has come to deploy such systems in Russia also. Regrettfully, the quality of telephone networks in Russia is insufficient but at present old hardware everywhere is being replaced by modern equipment and the quality of telephone lines is improving. Besides we made sure that it is still possible to achieve good result if the right logic of dialog is found, good grammars are developed and if high quality hardware is used in voice servers.

Our Speech Blocks and dialog constructs successully work in the Telepat system [4] that provides its users voice service for control of theirs WebMoney purses. This service was deployed in April, 2004.

6. Acknowledgments

This work was supported by companies Polyet-Sirena, TAIS, Regiontrunk, WebMoney and Comptek.

We wish to thank all our colleagues for help and supporting.

7. References