Internet Speech Recognition Server

Miroslav HOLADA
SpeechLab, Department of Electronics & Signal Processing,
Technical University of Liberec, Halkova 5,
461 17 Liberec, Czech Republic

ABSTRACT

The goal of this article is to describe the design of Internet speech recognition server. The reason for building this server is the fact, that communication speed of Intranet and Internet rapidly grows, and we can divide speech recognition process to client and server parts. Such solution would allow a wider use of speech recognition technologies because all users, including those that have relatively obsolete hardware incapable of speech recognition, would be served with speech recognition from the side of our server. The article discuses net data flow reduction, client-server structure and present two demo applications.

Keywords: recognition server, dialogue system, client-server, MOS, HMM.

1. INTRODUCTION

The robust speech recognizer for the Czech language was developed in our team SpeechLab in 2001. It allows recognition via telephone line or via standard microphone and recognizes isolated words and short phrases. Its vocabulary can contain thousands of items, and its recognition rate is higher than 98 percent. The recognizer is based on the HMMs (Hidden Markov's models). Recently we have used first 13 MFCC and their 1st and 2nd derivatives as recognition features. The HMMs are optimized for microphone recognition and the recognition via telephone line.

The recognizer consumes a lot of computing time and huge memory. This disadvantage does not allow to apply our recognizer in common applications such as speech controlled computer in the office, speech applications control for disabled persons or simple speech information system in the buildings and companies. The recognizer could be used only in specialized systems with corresponding equipment such as telephone dialogue information system or phone banking.

It is necessary to approach a speech recognizer like a "black box" to allow a wide use of speech recognition technologies, because most applications or system administrators will not have deep knowledge of speech processing. They wouldn't know how to configure recognizer, witch features to use or how to configure a noise reduction. However, they would like to use the newest versions of speech recognizers in their applications.

Our speech recognition server offers one of the possible solutions. We have only one recognition server with corresponding hardware and a lot of various applications.

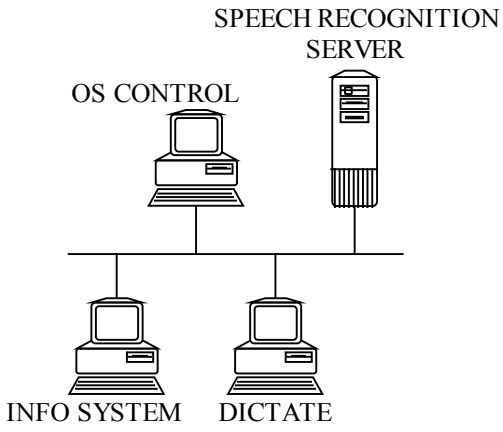

***Figure 1**. One power speech recognition server and several applications.*

2. CLIENT – SERVER ARCHITECTURE

The design of client – server architecture arises from common net structure where server offers speech recognition for client's programs. When we were determining which part of recognition process would run and where, we had the next criteria:

- The math operation on client side should be minimal in view of minimal hardware requirements.
- The user can select among several kinds of recognizers and recognition models.
- The user cannot use speech recognizer without server provider authorization.

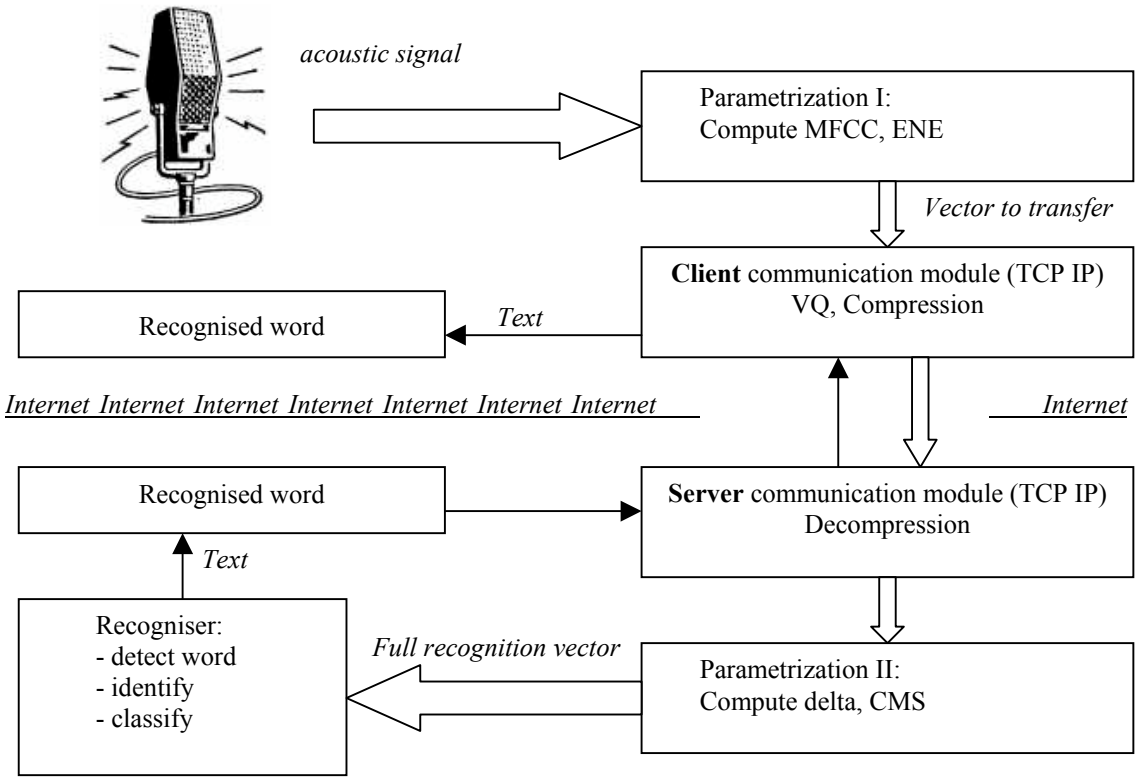

Figure 2. *Designed architecture of client–server distributed system.*

- The provider can modify and upgrade speech recognizers without the necessity of upgrading user's applications.

Speech recording and computing of basic recognition features runs only on client's side. These features are transferred to the recognition server via Internet. On the recognition severdelta recognition features are computed and demanded recognition is executed. Finally, the recognized text is sent back to client's speech application.

The structure of system is shown in figure 2. The most important are communication modules that contain communication interface between application and net card. There are also compression and decompression algorithms implemented implemented in the system.

Our server allows anonymous users the connection with a certain limitation (size of vocabulary, vocabulary kinds, duration of connection). Registered users can attach without restrictions, and they are limited only by speech recognition server speed (number of connected users).

3. COMMUNICATION PROTOCOLS AND TIME RESPONSE

The time response between the finishing of the spoken word and the incoming of recognized text is the most important property of client-server speech recognition system. Because of the fact that the dialogue system runs in real time, it is necessary to ensure that the recognizer's response is shorter than two seconds. The user can lose context of dialog if the response is longer. Of course, the recognition rate is also important, but it is not a subject of this article.

We have designed a prototype of recognition server and a trial client application. The tests showed that response time depends on data flow, server speed, and net connection. The communication is based on TCP-IP protocol that allows connection via the Internet. The server speed is given by hardware configuration, and the LAN net connection is required. The data flow is reduced when recognition features are transferred instead of recorded speech. Moreover, it is possible to send compressed speech, but compression and decompression degrease quality of speech and consume processor time.

Our compression method is based on feature scalar quantification. It is a lossy compression with fixed rate coders. The feature cannot be recovered exactly, though hopefully it would sound similar to the original. Each recognition MFCC feature is situated in specific range that we can split into 2^N regions. (N is number of bits.) The size of each region depends on the percent of occurrence for actual feature on reference set. Our reference set

was recorded by 80 different speakers, and in all it presents tens of hours of spoken speech records.

Designed technique of data flow reduction requires a prearranged quantification table, and self compression and decompression consists only in table search. This solution doesn't consume a lot of computing time and doesn't add next time delay in recognition process.

In our project we do not use well-known vector quantification (VQ) of MFCC because it is demanding of computational cost on client side. VQ is also more sensitive to various noises for example from phone line than scalar quantification.

We can't restore original wave signal after compression, so we can't measure quality of this signal by MOS (Mean Opinion Score). Hence we compare recognition results with results from uncompressed recognition.

When we reduce the data flow from 32 bits per one feature (4-bytes float point format) to 16 bits and then decompress it again to 32 bits, the recognition score fallby 1%. We can compress every feature down to 10 bits without considerable recognition score reduction.

Fixed bit rate for one feature	Set A	Set B
Uncompressed feature (4 bytes)	98.7%	95.4%
16 bits	98.7%	92.0%
12 bits	98.7%	91.8%
10 bits	98.6%	91.1%
8 bits	98.3%	88.2%
6 bits	94.7%	59.2%
4 bits	92.5%	47.4%

Table 1. *The recognition experiments with compressed features. Set A was recognised by isolated-word speech recogniser and set B by sub-word unit speech recogniser.*

Table 1 shows results of experiments with various compression bit rate. Set A presents recognition with isolated words recognizer. Set B presents sub-words unit recognizer, which is used for recognition of huge vocabularies. This recognizer has more decaying recognition score.

In present time our extended recognition server offers selection of saved vocabularies, creation new vocabulary, specification of group of items from current vocabulary for actual recognition, vocabulary listing, ordering of recognized words by scores, and other services. Our server offers also TTS (text to speech) services for complete using in dialogue system. It is available both in on-line and off-line mode. It means that client application sends a text message to server and TTS subsystem generates a wav file that is sent back to the client.

The distributed speech recognition system works with the Czech speech models (based on HMMs), but generally the system can work with various languages, it depends only on recognizer's setting.

We have prepared also a test application that allows repeated real on-line simulation of speech recognition. It replays beforehand record of words and sends it into server in the same way as real speech application.

The test was focused to verify hardware requirement on client's side. Table 2 shows that there are no considerable differences between computers with newer processors (Athlon, Pentiun III) and older computers.

Client's computer configuration Server – Pentium III 550 MHz	Min. Response [ms]	Aver. Resp. [ms]	Max. Resp. [ms]
Athlon 1.4 GHz	380	602	1001
Pentium III 550 MHz	383	606	1018
Pentium III 450 MHz	361	597	1002
Celeron 400 MHz	379	607	1021
Pentium MMX 233 MHz	385	607	1005
Pentium 120 MHz	380	604	1014

Table 2. *The dependence of response between server and client on client's hardware.*

The next experiment was focused on testing how the response time depends on number of clients connected at the same time. It has shown that connection response doesn't depend linearly on the number of clients. The reason for this is that a real recognition engine can process only one word at a time. The recognizer of isolated words works this way, but continuous speech recognizer can process the whole sentence, so that the user doesn't have to wait after uttering each word.

4. EXAMPLES OF CLIENTS

We have prepared two examples of client applications. They are simple drawing studio controlled by voice and demo dialogue information system. The "drawing studio" contains simple command for the pen moving (up, down, left, right), sizing (thin, middle, far, smaller, bigger) and color (white, black, …). Next, common commands

are supported too (back color, basic shapes, filling, deleting, text writing – after this command user has to write text manually). The vocabulary contains 150 words and short phrases, because most commands could be said by several synonyms.

When vocabulary size is lower and command set is reduced, the recognition score is higher. On the other hand, user's comfort falls and users don't have alternative options in their communication.

The second example shows very simple computer controlled dialogue system. It offers information about virtual mail-order company (stock list, prices, names of employers).

The both applications demonstrate all now supported functions of recognition server. The developer can download also source code with comments and make the tests or the changes. After the first experiences he can write his or own speech recognition applications.

Communication protocols between server and client are described and published on our web pages. In the future we want our university students and developers to use our far Internet recognizer server in their applications. It will enable using speech technologies for wide users.

The communication is controlled by simple commands, for example: DATA (data block from client to server, size of each block is under 2048 bytes, blocks are sent in 300 ms intervals), UMSG (user message – text message is sometimes useful for debug communication), PING (it tests connection and response time), DMSG (it sends text of recognized word from server to client).

5. CONCLUSION

Designed system allows to provide speech recognition applications via network on common office computers without high-performance hardware requirements. It opens new changes of speech technology usage.

The tests have also shown, that providing two or more speech applications (based on isolated words or short phrases recognition) by a single recognition server at the same time are possible.

The time response is similar when speech application runs on computer with processor Pentium 120 MHz or Pentium III 650 MHz, because speech recognition runs on the fast server. The time response strongly depends on connection speed and providing this system with modem networking could be troublesome. In the future we are going to propose even higher data flow reduction, especially, we want to implement float bit rate compression.

Acknowledgments:

This work was supported by CESNET Development Fund (grant no. 004R1/2002), by the Grant Agency of the Czech Republic (grant no.102/02/0124) and project MSM 242200001.

REFERENCES

[1] X. Huang, A. Acero, H-W. Hon: Spoken language Processing, A Guide to Theory, Algorithm, and System Development, Upper Saddle River, New Jersey 07458.

[2] Larsen, L.B.: Voice Controlled Home Banking - Objectives and Experiences of the ESPRIT OVID Project. Proc. of IVTTA'96, Basking Ridge, 1996.

[3] Holada M., Nouza J.: A City Information System Operating on Telephone. To appear on IVVTA Workshop, Torino, Italy, Sept. 29 – 30,1998, pp. 141-144.

[4] NOUZA J.: Speech Processing Technology Applied in Public Telephone Information Services. Proc. of 4th World Conference on Systemics, Cybernetics and Informatics (SCI 2000), Orlando, July 2000, vol. IV, pp.308-313 (ISBN 980-07-6690-1).

[5] NOUZA J., Holada M.: A Voice-Operated Multi-Domain Telephone Information System. Proc. of 25th Int. Conference on Acoustics, Speech and Signal Processing (ICASSP2000), Istanbul, June 2000, vol.VI, pp.3755-3758 (ISBN 0 7803-6296-9).

[6] Burget L., Motlicek P., Grezl F., Jain P.: Distributed speech recognition. Radioengeneering. Vol. 11, No. 4, Pp. 12-16, December 2002.

[7] http://itakura.kes.vslib.cz