音声言語情報処理 29－29
(1999. 12. 21)

クライアント・サーバ型 ATR-MATRIX

シンガー ハラルド、グルーン ライナー、内藤正樹、塚田元、西野敦士、中村篤、匂坂芳典

ATR 音声翻訳通信研究所

〒 619-0288 京都府相楽郡精華町光台 2-2

Tel.: 0774-95-1389 e-mail: singer@itl.atr.co.jp

あらまし ネットワークを介してどこでも手軽に音声翻訳機能の利用が可能な、クライアント・サーバ型音声翻訳システムを試作した。システムの構成上、クライアントーサーバ間のデータ転送帯域が限られることは大きな問題の一つであり、本システムでは、転送情報の圧縮を、入力、出力音声のそれぞれに適した手法で行なうことにより、その解決を図っている。試作したクライアント機能は、Windows95/98、Linux、OSF1 のそれぞれの OS 上で動作可能である。また、携帯電話を用いたモバイル環境 (9.6kbps) での本システムの動作確認も行なった。

キーワード ● 音声翻訳 ● 音声認識 ● システム

Speech Translation Anywhere: Client-Server Based ATR-MATRIX

Harald Singer, Rainer Gruhn, Masaki Naito, Hajime Tsukada, Atsushi Nishino, Atsushi Nakamura, Yoshinori Sagisaka

ATR Interpreting Telecommunications Research Laboratories

2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288

Tel. 0774-95-1389 e-mail: singer@itl.atr.co.jp

Abstract

We describe the implementation of a client-server based speech translation system. To minimize the bandwidth problem, input speech data is preprocessed, compressed and then sent to the recognition server (typically at 5.6 kbps). For synthesizing the translated utterance we implemented a variety of approaches requiring between 256 kbps for high quality speech down to less than 1 kbps for unit information. The client was implemented for Windows95/98, Linux and OSF1. The system was also tested using a 9.6 kbps mobile telephone data connection.

key words ● speech translation ● speech recognition ● system

1 Introduction

ATR-MATRIX [1, 2], a speech-to-speech translation system developed at ATR ITL, is a research prototype with high translation accuracy. It was designed with a highly modular structure, consisting of a speech recognition subsystem (ATRSPREC), a language translation subsystem (TDMT), a speech synthesis subsystem (CHATR) and a main controller (see Figure 1).

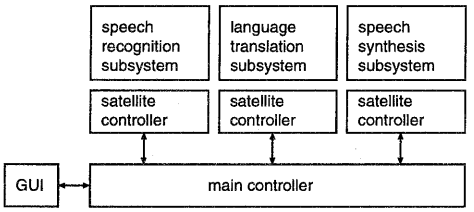

Fig. 1: ATR-MATRIX system architecture

This stand-alone version of ATR-MATRIX has previously been evaluated for naive users [3]. It was also successfully demonstrated in the C-Star international experiment on July 22nd, 1999 [4]. Recently, it was demonstrated that ATR-MATRIX achieves about 500 points on the TOEIC test, which is roughtly equivalent to a Japanese student entering a university[5].

However, running ATR-MATRIX on a standard PC, laptop, or even PDA to achieve "speech translation anywhere" requires a serious research and development commitment. Currently ATR-MATRIX 1) requires about 250 MB RAM, 1GB of disk, a fast CPU (PentiumII-400 class), 2) is a UNIX-based design which has only been implemented on Linux and OSF1, 3) uses proprietary software like C++ class libraries, 4) requires high-quality audio (microphone and AD converter) and 5) as an active research project has frequent updates to programs and models.

To "get ATR-MATRIX out of the laboratory into the real world" we introduce here a client-server system architecture for ATR-MATRIX. Dividing ATR-MATRIX into a client front end and a server main part solves most of the problems listed above or at least hides them from users. The system becomes portable and easy to install and use.

2 Implementation

2.1 Splitting up ATR-MATRIX

The split into client and server for both, recognition subsystem and synthesis subsystem, was chosen 1) to minimize required bandwidth and, 2) to minimize processing requirements on the client. Figure 2 shows the client-server architecture.

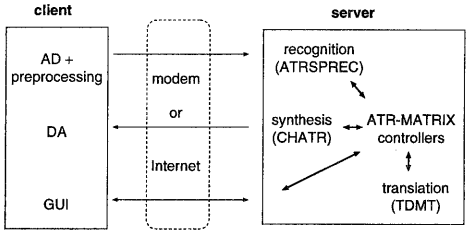

Fig. 2: Splitting up ATR-MATRIX modules into client and server

For the recognition subsystem, we perform feature extraction on the client and then compress the cepstral parameters to 5.6 kbps without degradation of recognition accuracy [6]. [1]

For the synthesis subsystem, unit selection is performed on the server and the start and end time of each selected wave chunk is sent to the client. The client simply concatenates the wave chunks by accessing the locally copied database. The necessary bandwidth is less than 1 kbps. One disadvantage for this scheme is the necessity for a local copy of the database, requiring between 10 and 100 MB per desired speaker.

2.2 Administration of Multiple ATR-MATRIX Servers

To efficiently use resources, a server management system was implemented as shown in Figure 3. Clients log-in to this resource manager by transmitting some client properties and the desired type of service. The resource manager has a list of existing servers and their properties, thus supporting various server types, such as special servers for clients communicating by telephone instead of using a PC or for different language pairs. As session initiation protocol we use SIP [8].

3 Application Examples

3.1 One-Way Japanese-English

An example screenshot of a client running on Windows95 is shown in Figure 4. A Japanese utterance is preprocessed on the client, transferred to the server, decoded, translated, appropriate wave chunks are selected, the file indices for these chunks are transferred to the client, concatenated and played on the client DA.

[1] This has been improved down to 2 kbps at the expense of increased complexity[7].

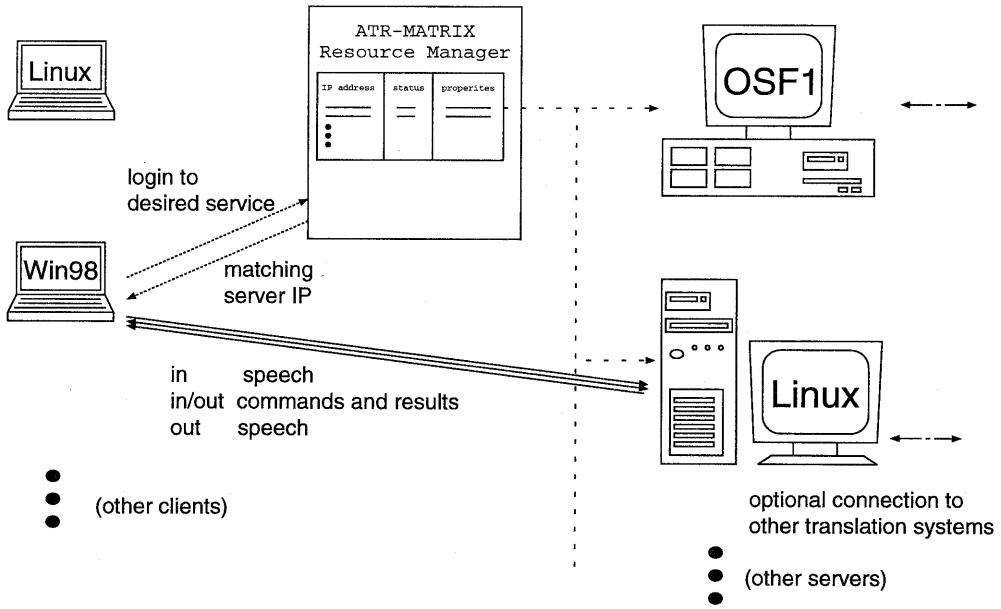

Fig. 3: Resource management for multiple ATR-MATRIX servers

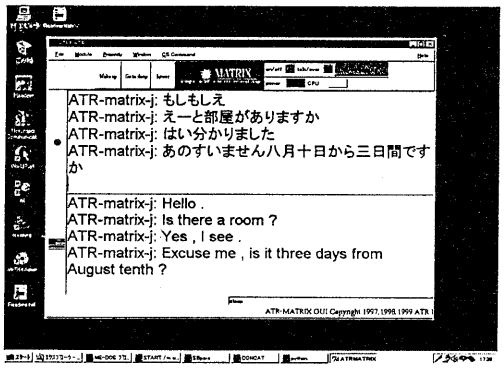

Fig. 4: Screenshot of client ATR-MATRIX on Windows95

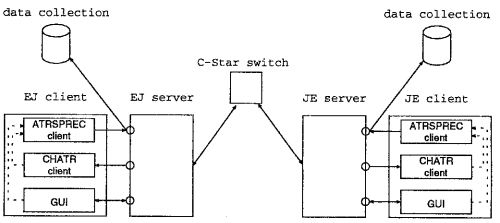

Fig. 5: Bi-directional English-Japanese translation system

3.2 Communicating with Other Translation Systems

For communication of a Japanese speaker with an English speaker, the Japanese-English system of the previous section can be connected to an English-Japanese translation system for example via the C-Star protocol (see Figure 5). The C-Star protocol basically broadcasts recognition and translation results as ASCII strings to all connected parties similar to IRC chat.

For the C-star experiment [4], we used a commercially available "wearable computer", a device consisting of a headset with screen and headphone and a small CPU part that can be carried on a belt around the hip. An American traveller and a Japanese, their wearables connected via mobile LAN to the EJ and JE servers, performed a simple information exchange showing the potential for mobile translation technology.

4 Evaluation

Evaluation experiments were performed to find out how easy the system is usable for an unexperienced user, hereforth called "customer" and which problems arise from unexpected use. The Japanese users were asked to reserve a hotel room using ATR-

MATRIX. A second ATR-MATRIX system was given to an experienced English speaker who acted as hotel clerk.

Each Japanese customer was given a one page technical manual how to use ATR-MATRIX and some pages of dialog content information such as a schedule, telephone and credit card number. The customers had to start with a standalone laptop, i.e. they had to open a modem connection to the Internet and to run ATR-MATRIX by themselves.

Figure 6 explains the experiment setup. Servers were a DEC Alpha 500/500 with 500 MB RAM and Digital UNIX V4.0D for Japanese and a Compaq workstation XP-1000 with 400 MB RAM running Digital UNIX V4.0E for English. Client computers were a Panasonic CF-S22 Laptop with Intel MMX processor (266 MHz), 95 MB RAM and built-in 56 kbps modem for Japanese and a Panasonic AL-N2 Laptop with Intel Pentium 133 and LAN network connection for the English side.

Usually, the main purpose of evaluation experiments is to get recognition rates, translation accuracy ratings etc. for untrained speakers. Here, the main objective of the evaluation experiment reported was to find out problems occuring when people unfamiliar with ATR-MATRIX and the client software are asked to use it. For this purpose, the first experiments are the most valuable ones. The experiments provided insight in the process of evaluation experiments.

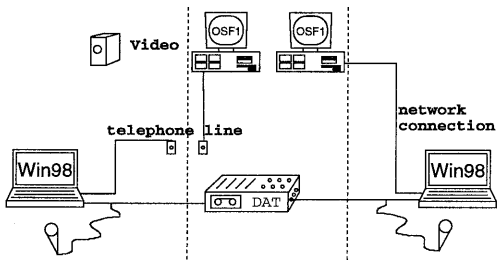

Fig. 6: Client-server ATR-MATRIX evaluation experiment setup.

4.1 Task Performance and Discussion

Table 1 shows a list of customers and how long it took them to complete the task, i.e. to reserve a hotel room. Most customers finished the reservation conversation after some 40 minutes. The total conversation time splits into setup time, including connecting the computer via modem to the network and starting ATR-MATRIX, and the actual dialog time. Customer F0001 could not connect to the dial-in network because the modem line was busy, so the setup time for her is very long.

Tab. 1: Timing information for each customer in hours:minutes:seconds

ID	turns	setup	dialog	total
M0001	40	3:07	35:11	38:18
F0001	80	15:26	55:28	1:10:54
F0002	79	2:06	38:19	40:25
F0003	72	4:20	49:55	54:15

All four speakers completed their task, although the task completion times were far too long. In the "real" world, they would have probably given up. Interestingly, speaker F0003, with a recognition accuracy of under 50% took less time to complete the task than speaker F0001 who achieved more than 80% word accuracy. Although the small sample size makes it impossible to draw any conclusions, it seems that word accuracy and task completion rate are not necessarily highly correlated,

Compared to a simple phone call, "setup time" is also too long, i.e. between 2 and 4 minutes if nothing goes wrong. This is partly because the customers were unsure of what to do and read the instructions carefully before each of the single steps.

4.2 Audio Device Dependent Word Recognition Accuracy

The column "DAT" in Table 2 shows word recognition accuracy for different audio devices. The numbers in column "DAT" are computed using the recorded speech and manually generated transcriptions for scoring. As the speech was recorded on DAT before passing through the client audio device, the calculated accuracy is not identical to the recognition accuracy during the experiment.

Tab. 2: Word accuracy in % for DAT, Yamaha OPL3-SA3, and ESS Technology ESS1876

ID	DAT	OPL3-SA3	ESS1876
M0001	88.28	88.62	89.66
F0001	85.32	83.50	83.80
F0002	72.86	77.51	81.41
F0003	44.03	48.22	47.38
average	72.50	74.17	75.28

To have more precise information about the actual recognition rate the speech that had been recorded on DAT was used as input to the audio device of the laptop computer used in the experiment. The column labelled with "OPL3-SA3" in Table 2

shows the recognition results for this setup. To measure the influence of the laptop audio device on recognition accuracy, the same speech data was filtered through one more laptop with an ESS1876 audio device.

The influence of the audio device is amazingly small. The significant drop in recognition rate we expected to find if we use "laptop audio device speech" instead of high quality speech directly from DAT did not occur. On the contrary we got a slight recognition rate improvement with speech piped through the laptop audio devices.

5 Future Work

The results show that there is still a lot of effort required for building a useful speech translation product. Major unsolved problems are

audio device quality: in a client-server setting, control of the client audio device becomes very difficult. An informal test with 8 different laptops showed that the quality of the AD was quite bad for 6 of them, with audible(!) differences even for the same model. We hope that USB based audio devices will solve this problem.

status display and feedback: the current status of the server should be fed-back to the user. In the standalone ATR-MATRIX system, this is achieved by an additional video channel and 100 msec updates of the current recognition status. This would require too much bandwidth in the client-server setting.

evaluation: overall system evaluation is a time-consuming process that requires bi-lingual speakers to manually grade the translation quality. Automatic evaluation for translation results is still an open research area.

porting: we have ported the client side of ATR-SPREC to WindowsCE and verified that AD and preprocessing is working. However, real-time performance was about 100 times slower, probably due to inefficient floating-point operations and inefficient wrapper code for functions like *getc()*. We are now working on the ultimate light-weight client, i.e. a simple cellular phone. This requires data collection and retraining of acoustic models.

6 Conclusions

With a small effort of overall 6 man-months, it was possible to split-up the complex ATR-MATRIX translation system in a relatively simple client side and a (still complex) server side. Additionally, this client side was ported to Windows95/98, Linux and OSF1. This was made possible by two main design criteria:

- the modular, eventloop-based paradigm of ATR-SPREC [9]
- the use of Python, an object-oriented, cross-platform RAD language.

ACKNOWLEDGMENT

The authors would like to thank all members and the management of ATR ITL for their support. In particular, we would like to thank Ryosuke Iwasawa and the members of the "Technical Support Group" Benjamin Reaves, Koji Takashima, Toshio Ban and Takeshi Matsuda.

References

[1] B. Reaves, A. Nishino, and T. Takezawa. ATR-MATRIX: Implementation of a speech translation system. In *Proc. Acoust. Soc. Jap.*, pages 53–54, Spring 1998.

[2] T. Takezawa, T.Morimoto, Y. Sagisaka, N. Campbell, H. Iida, F. Sugaya, A. Yokoo, and S. Yamamoto. A Japanese-to-English speech translation system: ATR-MATRIX. In *Proc. ICSLP*, pages 957–960, 1998.

[3] F. Sugaya, T. Takezawa, A. Yokoo, and S. Yamamoto. Evaluation on bidirectional Japanese and English ATR-MATRIX. In *Proc. Acoust. Soc. Jap.*, pages 107–108, Spring 1999. (in Japanese).

[4] C-Star. Consortium for Speech Translation Advanced Research (C-Star): C-Star Experiment. http://www.c-star.org/, July 1999.

[5] F. Sugaya. Total evaluation on ATR-MATRIX. http://www.atr.co.jp/event/expo/expo99/, November 1999.

[6] R. Gruhn, H. Singer, and Y. Sagisaka. Scalar quantization of cepstral parameters for low bandwidth client-server speech recognition systems. In *Proc. Acoust. Soc. Jap.*, pages 129–130, Spring 1999.

[7] S. Tsakalidis, V. Digalakis, and L. Neumeyer. Efficient speech recognition using subvector quantization and discrete-mixture HMMs. In *Proc. ICASSP*, pages 569–572, 1999.

[8] M. Handley, H. Schulzrinne, and E. Schooler. SIP:session initiation protocol. Internet-draft RFC 2543, Internet Engineering Task Force, March 1997. ftp://ftp.isi.edu/confctrl/docs/draft-ietf-mmusic-sip-02.ps.

[9] H. Yamamoto, H. Singer, B. Reaves, and Y. Sagisaka. Control and structure of recognition subsystem in the ATR-MATRIX Japanese-English speech translation system. In *Proc. Acoust. Soc. Jap.*, pages 161–162, Spring 1998. (in Japanese).