Development and Evaluation of Speech Database in Automotive Environments for Practical Speech Recognition Systems

Yasunari Obuchi and Nobuo Hataoka

Central Research Laboratory, Hitachi Ltd.
Kokubunji, Tokyo 185-8601, Japan
obuchi@rd.hitachi.co.jp, hataoka@crl.hitachi.co.jp

Abstract

Aiming at practical speech recognition systems, we are developing speech databases representing the situation in which the application is used, and evaluating various techniques using the database. Such methodology is expected to contribute to bridge the expectations of the developers and the reactions of the users. We start with the applications in automotive environments, or car navigation systems more precisely. During the data collection, special attention was paid to maintain the spontaneousness of the speaker, to cover failed utterances, and to use the hardware setup suitable for microphone array techniques. After the database is prepared, various techniques are evaluated. In some cases, oracle information is used to find the upper limit of the improvement of a specific module. In other cases, typical improving algorithms are tested. Recognition experiments using two separate decoders indicate that endpoint detection, feature normalization, speaker adaptation, and parallel decoding are promising fields. We also present some modifications of parallel decoding to reduce the computational cost and to realize practical applications.

Index Terms: speech database, automotive environments, module-wise evaluation.

1. Introduction

Speech recognition technologies have been improved in these years, and it is reported that the state-of-the-art speech recognition system provides very high recognition accuracy for various databases. However, a lot of users still insist that the speech recognition systems make too many errors and they are helpful only in limited occasions.

One of the causes of the discrepancy between the research results and the market reaction is that the evaluation database is not representing completely the environment where the system is used. In the research process, the user is controlled well to make clear utterances. Besides, various unexpected events are removed from the database as the wrong data. Although there are some exceptions in which spontaneous speech with unlimited grammar and lexicon is targeted, such applications are too hard to appear in the market. However, even in a simple isolated word recognition task, such events occur more frequently when the system is used as a commercial product and the user is not controlled by the engineer. Consequently, the user feels that the system performance is poor, and that it's better to use an alternative interface modality.

In order to leverage the achievements in various fields of speech recognition research, it is essential to prepare a database that represents the product and to evaluate various technologies on an equitable basis. In this work, we focus on the automotive systems, collect a database that represents automotive environments, and evaluate various methods using the database.

In the database collection process, we paid special attention to (1) spontaneousness of the user's utterance (2) coverage of failed utterances, and (3) microphone setting corresponding to the latest research trend. The first and second requirements are realized in an appropriate instruction in the recording sessions, and the third requirement is realized as a microphone array.

In the evaluation process, we divide the speech recognition system into various modules such as endpointing, speech enhancement, and decoding. For each module we have two options. First, we can make oracle experiments assuming that the module does the perfect job. It gives us the upper limit of the improvement in the module, and we can assess how important the module is. Second, we apply a few typical improvement methods applicable to the module. It gives us the expectation of the improvement in the module, and we can asses how promising the research in this field is. By combining these results, it is expected that the direction of the future research for practical systems would become clear.

2. Navigation Speech Database in Real Automotive Environment

In car navigation systems, speech interfaces are widely used for safety reasons. Taking a large amount of potential users into account, such a system would be a good example of practical speech recognition systems. We have collected the speech data in a real car driving on the urban roads. In this section, we describe the details of the recording session and the database.

The recording was done in downtown Tokyo, where the car was forced to drive slowly with frequent stops due to the traffic jam. Therefore, a large part of the background noise is from the surrounding environment, such as other cars, constructions, shops, railways, etc. The speaker was sitting on the passenger seat, and there was a linear microphone array on the dashboard in front of the speaker. The array consists of seven microphones, which are located at the interval of 10cm, 5cm, 5cm, 5cm, 5cm, and 10cm. Array microphones were labeled as #1 to #7 from the driver's side to the window side, so #4 is the central microphone. These microphones and a reference close-talking microphone are connected to a multi-track recorder (MTR). The close-talking signal was also sent to a laptop PC, in which a speech recognition program is running. The speaker has a push button (PB) to activate the recognition program. After the button is pushed, a prompt message and a beep are given to the speaker, and the recognition program starts to receive the signals. The hardware setup is shown in Fig. 1.

To maintain the spontaneousness of the utterances, we gave

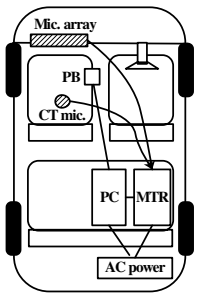

Figure 1: *Hardware setup for data collection in a car.*

Table 1: *Estimated SNR of each microphone data*

mic. ID	SNR (full band)	SNR (400-5500Hz)
1	-5.0	9.3
2	-2.8	12.1
3	-3.4	8.6
4	-3.0	9.2
5	-2.7	11.7
6	-3.8	8.5
7	-2.9	10.5
close-talk	56.7	83.2

the following instructions to the speakers. First, we prepare a booklet of road map on which pre-defined 152 points of interest (POIs) are marked on several pages. We gave the road map to the speaker with the instruction to

- find a POI,
- memorize it,
- close the booklet, and
- utter the name of POI.

Besides, we asked the speaker to select the POI always from a different page of the road map from the previous utterance, and not to select POIs in the same category (station, hotel, park, etc) for many times. These instructions make the speaker think of a lot of things, and prevent them from simply reading the names from top to bottom. To give the same feeling to the speaker as in the real situation, a speech recognition program is running on a laptop PC. When a misrecognition occurs, the speaker is asked to repeat the utterance only once.

Under the conditions described above, we have collected the speech data from 18 speakers (11 male and 7 female, all in their early twenties). There were 3,620 utterances in total, and they were roughly segmented using a fixed time period from the beep. After segmentation, the length of the data was approximately 7 hours in total. These utterances were then labeled by the POI name. Some utterances include wrong pronunciation and hesitation, but they are all labeled as long as the intention of the speaker can be inferred. In this process, we found 28 (0.8%) utterances which could not be labeled as any POI and were categorized as OOV (out of vocabulary). The number of utterances per speaker ranged from 134 to 326, and the number of utterances per POI ranged from 10 to 48. These numbers are supporting the argument that the utterances were made spontaneously.

Next, we extracted the endpoint information using the close-talking data and the POI label using Viterbi alignment. They were

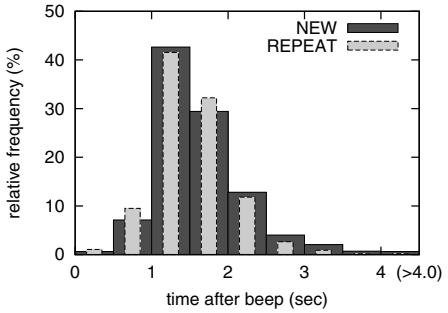

Figure 2: *Histogram of the time between the beep end and the speech onset. Dark gray bars represent the new utterances (average 1.62sec and standard deviation 0.61sec) and the light gray bars represent the repeated utterances (average 1.54sec and standard deviation 0.51sec).*

Table 2: *Results of baseline experiments*

Decoder	Distant-talk	Close-talk
Original	87.3	91.3
Julian	86.0	93.4

used as the "oracle" endpoint information for the noisy data. We then estimated the signal-to-noise ratio (SNR) by comparing the power of the speech and non-speech segments. Table 1 shows the estimated SNR for each microphone. Since the noise spectrum has s strong peak in the low-frequency range, we also calculated the SNR after applying a bandpass filter with the passing band of 400 to 5500Hz. It is interesting that the estimated SNR does not have any correlation with the distance between the speaker and the microphone, although the speech recognition accuracy has a correlation with the distance as mentioned later in this paper.

To confirm the spontaneousness of the utterances, we measured the time between the beep end and the speech onset. Since the speaker was not reading the list, the time to recall the POI name may have large variations. We can compare it with the cases in which the speaker was repeating the misrecognized utterance, for which the hesitation time was expected to be shorter. There are 3024 new utterances and 568 repeated utterances (excluding 28 OOVs), and Fig. 2 shows the histogram. It was clearly proved that the new utterances had larger variations and longer average.

3. Evaluation Results

3.1. Baseline Experiments

After collecting and analyzing the data, we carried out evaluation experiments of 152 POI isolated word recognition. Most of the experiments were done in parallel using our original decoder and Julius [1] to ensure the reliability of the results. For the original decoder, we trained triphone HMMs (1614 states) using 16 hours of clean training data consisting of phonetically balanced sentences. 13 MFCC parameters including C0 and their first and second time derivatives are used. The original decoder has no rejection function. For Julius, the sample acoustic model with PTM triphones, which is distributed with the source code, was used. Among various variations of Julius, the Julian-v3.4.2 grammar-driven decoder with 12 MFCC and log power, plus their first-order time deriva-

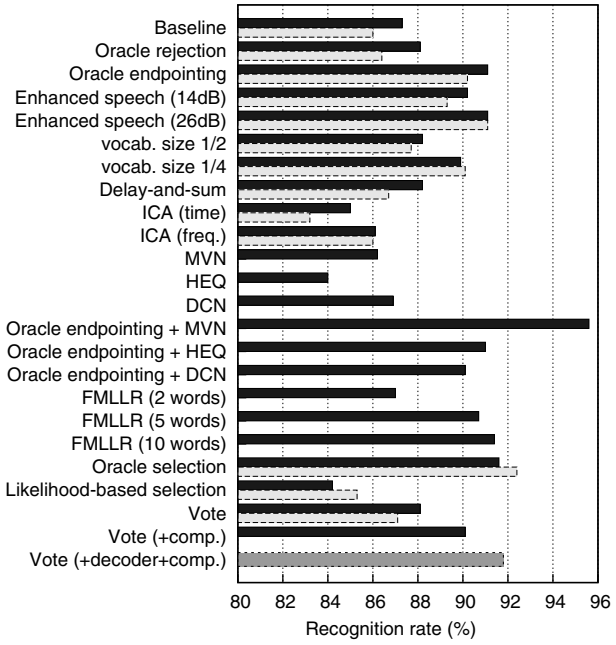

Figure 3: *Summary of evaluation experiments. Black bars represent the original decoder, light gray bars represent Julian, and the dark gray bar at the bottom represents their combination.*

tives is used. The path of the silence models only was allowed, which is interpreted as rejection. All the data were originally sampled by 44.1kHz, but downsampled to 16kHz prior to the experiments. In the baseline experiments, fixed length segments are used without any sophisticated endpointing, and cepstral mean normalization (CMN) was applied.

Table 2 shows the results of the baseline experiments. In the distant-talk microphone experiments, only the central microphone was used. Even though two decoders use different acoustic models, the results are close to each other. In the experiments using the original decoder, the individual recognition rates ranged from 60.3% to 97.4%.

3.2. Evaluation of various modifications

Next, we evaluated modifications in various modules of the speech recognition system. All results are summarized in Fig. 3.

First, the importance of rejection and endpointing are evaluated using oracle information. When we used the oracle information about OOV, 28 OOV utterances were automatically recognized correctly. Since the original decoder does not have rejection function and always misrecognized an OOV utterance, the recognition rate was improved 0.8% absolute. Julian recognized some of 28 OOV utterances correctly, and the improvement was 0.4% absolute. When we used the oracle information of the speech period, the recognition rate increased to 91.1% (original) and 90.2% (Julian), indicating very large improvements.

Two more sets of the oracle experiments were carried out to investigate how the recognition rate changes according to the SNR improvement and vocabulary size refinement. As for the SNR, the distant-talk data were mixed with the close-talk data linearly in the time domain with a varying weight. Two typical points are plotted, where the SNR was calculated after applying the bandpass filter. The improvement from the baseline to 14dB is much larger than the improvement from 14dB to 26dB. The vocabulary size refinement was done simply by splitting the dictionary into two or four groups, and the one including the correct word was used for recognition. Steady improvements were observed every time the vocabulary size was reduced.

Next, two typical microphone array techniques were evaluated. The delay-and-sum beamformer [2] was implemented in a simple manner using the fixed delay between microphones. The signals of seven microphones are first upsampled to 64kHz, then added delays, summed each other, and downsampled to 16kHz again. ICA [3] was tested in the time and frequency domains, using the microphones #3 and #5, next to the central microphone (#4) on the both sides. The results show that the delay-and-sum beamformer provides small improvement (0.9% absolute for the original decoder and 0.7% absolute for Julian), but ICA does not improve the recognition rate at all. We also tried ICA using seven microphones, but we had even larger degradation. These results indicated that the majority of the noise was nondirectional, and some of them are correlated with each other due to reverberation.

In the feature (MFCC) domain, we tried three normalization techniques, Mean and Variance Normalization (MVN), Histogram Equalization (HEQ) [4], and Delta-Cepstrum Normalization (DCN) [5]. These techniques were tested with the original decoder only (because the acoustic model must be re-trained), and no improvement was observed. However, when we applied these methods with the oracle endpointing information, we got high recognition rates. In particular, MVN showed excellent performance with the oracle endpointing, and the recognition rate was 95.6%, which was 4.5% absolute better than CMN with the oracle endpointing. It indicates that the effectiveness of MVN is highly dependent on good estimation of the speech segment. Contrastingly, HEQ and DCN could not improve the recognition rate even with oracle endpointing. Since these techniques have more parameters to estimate, short utterances in these experiments would not be suitable for them.

As for the acoustic model, we tested Feature-space Maximum Likelihood Linear Regression (FMLLR) [6], which is an adaptation algorithm in the feature domain, but equivalent to the constrained one-class MLLR of the acoustic model. Adaptation was executed in an unsupervised manner, in which the reference label was obtained by recognizing the adaptation utterance. The results showed that the recognition rate can be improved if we use five or more words as the adaptation data. The recognition rate was 91.4% when we used 10 words for adaptation, and no more improvement was obtained when using more adaptation data.

Finally, we tried parallel decoding with the hypothesis selection. The signals of seven microphones are recognized in parallel, providing seven hypotheses, and one of them is selected. Oracle selection means that we have the complete knowledge about selection, and the recognition rate was calculated by counting the utterances for which at least one microphone signal was recognized correctly. The oracle selection recognition rate was 91.6% for the original decoder and 92.4% for Julian. These numbers suggest the high potential of the parallel decoding framework, but a standard likelihood-based hypothesis selection offers only poorer results than the baseline. In contrast, another simple approach by a vote by seven microphones is quite effective. The recognition rates were 88.1% (0.8% absolute improvement) for the original decoder and 87.1% (1.1% absolute improvement) for Julian. In addition,

Table 3: *Using six or less microphones for parallel decoding.*

microphones	Recog. rate (%)	ave. runs
1	89.3	3.3
2	89.5	5.6
3	89.7	7.9
4	89.8	10.1
5	90.0	12.2
6	90.1	14.5

Table 4: *Results of N-best parallel decoding.*

N	Recog. rate (%)
2	89.2
3	89.1
4	89.5
5	89.6
10	89.8
20	89.8

we can use seven more hypotheses made with MVN, seven more with HEQ, and seven more with DCN. If we have a vote by all of 28 hypotheses (Vote+comp.), the recognition rate of the original decoder is 90.2%, which means 2.9% absolute improvement from the baseline. Finally, a vote by these 28 hypotheses of the original decoder and 7 hypotheses of Julian (Vote+decoder+comp.) provides even better results, and the final recognition rate was 91.9%.

Among various approaches, parallel decoding showed promising results in terms of the recognition accuracy. However, parallel decoding has an intrinsic problem of the computational complexity. To analyze its applicability to practical systems, we studied more details about parallel decoding. First, it is obvious that we do not have to decode all 28 inputs if the vote is highly one-sided. When we finished k decoding, we can terminate the repetition of decoding without performance degradation if the difference of votes of the first and second hypotheses is (equal to or) larger than $N - k$ (depending on the tie-breaker criterion). We checked how many decoding runs are required for each utterance using 28 hypotheses provided by the original decoder with various feature normalization. We started from microphone #4 with CMN, and then tried #4 with MVN, #4 with HEQ, and #4 with DCN in this order. It is because normalizing a set of feature vectors is much faster than executing feature extraction for another microphone signal. After finishing #4, we continued microphones #3, #5, #2, #6, #1, #7 in this order, in each of which four feature normalization algorithms were applied in the same order as #4. In 2396 utterances, the first 15 decoding runs provided the same hypothesis, and no more repetition was needed. Only 150 utterances required full 28 decoding runs, and the average number of decoding runs was 17.0.

If we are still running out of the computational power, there are some more ways to reduce the computation by introducing approximation. The simplest way is to reduce the number of microphone used. Table 3 shows the results of the reduced experiments. The recognition rate increases smoothly as more microphones are used, and even with only one microphone, parallel decoding with various feature normalization provides good improvement.

Another way to reduce the computation is N-best parallel decoding, in which the best N hypotheses are selected in the first decoding run. All the other hypotheses are abandoned, and the Viterbi matching process becomes quite fast. Table 4 shows the results of N-best parallel decoding. Even with $N = 10$, the recognition rate is close to that of full decoding. Finally, if we combine two approximations, we obtained 89.6% recognition rate by 10-best parallel decoding of 5 microphones, in which 12.0 decoding runs were required in average.

4. Conclusions

In this paper, we introduced a new speech database for development of practical speech recognition systems in automotive environments. The database consists of the spontaneous utterances spoken under the real driving condition. The spontaneousness was proved by analyzing the distribution of the response time of the speaker.

We evaluated various speech recognition modules and algorithms using this database. Oracle experiments showed that it is important to achieve high quality endpoint information of each utterance. It was also found that Mean and Variance Normalization works effectively if correct endpoint information is given.

Evaluation of various algorithms showed that ICA is not suitable for the application of this work, and speaker adaptation and parallel decoding are two promising approaches. In particular, parallel decoding shows its best performance when it is combined with various feature normalization algorithms, and a vote by the hypotheses is adopted. The computational cost is a serious problem of parallel decoding, but it can be weakened by introducing various approximations.

We have shown an example of application-oriented database collection and system evaluation for automotive systems. A strategy for the future research was given by evaluation experiments, such as development of reliable endpoint detection algorithm, speaker adaptation algorithm, and parallel decoding and hypothesis selection algorithms. Similar approach would be helpful for other applications of speech recognition systems.

5. Acknowledgments

The authors are thankful to Prof. Sadaoki Furui of Tokyo Institute of Technology and Prof. Tetsunori Kobayashi of Waseda University for their valuable comments. This work was supported by New Energy and Industrial Technology Development Organization (NEDO), Japan.

6. References

[1] Julius - an Open Source Large Vocabulary CSR Engine, <http://julius.sourceforge.jp/en/julius.html>.

[2] W. Kellermann, "A Self Steering Digital Microphone Array," Proc. ICASSP, Toronto, Canada, 1991

[3] N. Murata, S. Ikeda, and A. Ziehe, "An Approach to Blind Source Separation Based on Temporal Structure of Speech Signals," BSIS Technical Report, 00-6, 2000

[4] A. de la Torre, et al., "Non-linear Transformation of the Feature Space for Robust Speech Recognition," *Proc. ICASSP*, Orlando, FL, USA, 2002

[5] Y. Obuchi and R. M. Stern, "Normalization of Time-derivative Parameters Using Histogram Equalization," *Proc. EUROSPEECH*, Geneva, Switzerland, 2003

[6] M. J. Gales, "Maximum Likelihood Linear Transformation for HMM-based Speech Recognition," *Cambridge University Engineering Department Technical Report*, 1997