PAPER

In-Vehicle Voice Interface with Improved Utterance Classification Accuracy Using Off-the-Shelf Cloud Speech Recognizer*

Takeshi HOMMA[†a)], ***Member***, **Yasunari OBUCHI**[††], ***Senior Member***, **Kazuaki SHIMA**[†††], **Rintaro IKESHITA**[†], **Hiroaki KOKUBO**[†], ***and*** **Takuya MATSUMOTO**[††††], ***Nonmembers***

SUMMARY For voice-enabled car navigation systems that use a multi-purpose cloud speech recognition service (cloud ASR), utterance classification that is robust against speech recognition errors is needed to realize a user-friendly voice interface. The purpose of this study is to improve the accuracy of utterance classification for voice-enabled car navigation systems when inputs to a classifier are error-prone speech recognition results obtained from a cloud ASR. The role of utterance classification is to predict which car navigation function a user wants to execute from a spontaneous utterance. A cloud ASR causes speech recognition errors due to the noises that occur when traveling in a car, and the errors degrade the accuracy of utterance classification. There are many methods for reducing the number of speech recognition errors by modifying the inside of a speech recognizer. However, application developers cannot apply these methods to cloud ASRs because they cannot customize the ASRs. In this paper, we propose a system for improving the accuracy of utterance classification by modifying both speech-signal inputs to a cloud ASR and recognized-sentence outputs from an ASR. First, our system performs speech enhancement on a user's utterance and then sends both enhanced and non-enhanced speech signals to a cloud ASR. Speech recognition results from both speech signals are merged to reduce the number of recognition errors. Second, to reduce that of utterance classification errors, we propose a data augmentation method, which we call "optimal doping," where not only accurate transcriptions but also error-prone recognized sentences are added to training data. An evaluation with real user utterances spoken to car navigation products showed that our system reduces the number of utterance classification errors by 54% from a baseline condition. Finally, we propose a semi-automatic upgrading approach for classifiers to benefit from the improved performance of cloud ASRs.

key words: *speech recognition errors, natural language understanding, car navigation, noisy environment, cloud speech recognition*

1. Introduction

Voice input interfaces are widely used in in-vehicle systems, i.e., for car navigation, car audio, hands-free phone systems, and so on, because its hands-free and eyes-free aspects allow users to drive more safely. Most commercialized in-vehicle systems can be operated through voice with embedded speech recognizers. Due to the limited computational resources of embedded computers, these systems can receive only fixed sentences, i.e., voice commands.

Meanwhile, multi-purpose cloud-based automatic speech recognition services (cloud ASRs) have been widely used in recent years, e.g., [1]. A cloud ASR has plenty of computational resources in its backend, so it can utilize acoustic and language models trained from a very large corpus and state-of-the-art speech recognition algorithms. Therefore, it can recognize a wide variety of sentences and words with higher accuracy than an embedded speech recognizer. By using a cloud ASR, if application developers can prepare a language understanding module for their target applications, they can develop a speech interface that users can use to operate devices with spontaneous utterances without remembering predefined voice commands.

We aim to create a car navigation system that can be controlled with spontaneous utterances by utilizing a cloud ASR. Cars are subject to large noises from the road and by engine noises. A microphone attached at a position far from a driver's mouth, i.e., the ceiling or a sun visor, records not only the driver's speech but also such noises. Most cloud ASRs are not designed to work well in such noisy environments. Such noises degrade the accuracy of speech recognition. In addition, because the language models of cloud ASRs are tuned for general purposes, the models might not be suited to recognizing specific words for car navigation systems. These gaps between cloud ASRs and car navigation systems cause speech recognition errors. These errors induce misinterpretation in language understanding. This misinterpretation results in users not being able to execute the car navigation function that they want.

For improving the accuracy of speech recognition in noisy environments, multi-conditional acoustic model training is known as one promising technique [2], [3]. To improve accuracy for task-specific words, language model adaptation can be applied [4]. However, such model customizations cannot be applied to cloud ASRs because these off-the-shelf cloud ASRs are a "black box" for application developers, so they cannot customize the models. Therefore, we need techniques for reducing the number of speech recognition errors and language understanding errors without needing to modify the internals of cloud ASRs.

The purpose of this study is to improve the accuracy of utterance classification so that a car navigation system can better predict which car navigation function a user wants to

Manuscript received March 6, 2018.
Manuscript revised July 12, 2018.
Manuscript publicized August 31, 2018.
[†]The authors are with R&D Group, Hitachi, Ltd., Kokubunji-shi, 185–8601 Japan.
[††]The author is with Tokyo University of Technology, Hachioji-shi, 192–0982 Japan.
[†††]The author is with Clarion Co., Ltd., Saitama-shi, 330–0081 Japan.
[††††]The author is with Hitachi Automotive Systems Ltd., Atsugi-shi, 243–8510 Japan.
*This is a paper on system development.
a) E-mail: takeshi.homma.ps@hitachi.com
DOI: 10.1587/transinf.2018EDK0001

execute from the outputs of a cloud ASR that contain speech recognition errors. Utterance classification is a language understanding technology, and it is necessary for controlling a car navigation system with spontaneous utterances. Thus, we propose a system for utterance classification that is robust against speech recognition errors without the need to modify the internals of cloud ASRs. We incorporate the following approaches. First, our system applies speech enhancement to user speech signals to reduce the number of misrecognized words. Instead of applying this enhancement solely to user speech signals, our system sends both enhanced and non-enhanced speech signals to a cloud ASR and merges speech-recognition results derived from both speech signals. Thus, we aim to reduce the number of misrecognized words in different noise environments.

Second, to improve the accuracy of utterance classification when the outputs of a cloud ASR still contain misrecognized words, our system incorporates "optimal doping," which mixes transcribed sentences (with no misrecognized words) and sentences recognized by the cloud ASR (with misrecognized words) to create training data for an utterance classifier. Optimal doping provides the classifier with "immunity" against misrecognized words so that it can achieve highly accurate utterance classification regardless of whether the cloud ASR outputs misrecognized words or not.

Third, our proposed system is evaluated by using actual user utterances with commercial car navigation products. We confirm that our system improves the accuracy of utterance classification with actual utterances.

Finally, we show a method for adapting the proposed system to changes in output characteristics of cloud ASRs. The cloud ASRs usually improve their recognition accuracies frequently. This means that the characteristics of speech recognition results may change unpredictably. Therefore, it is important that our method can maintain good utterance classification accuracies even when the output characteristics are changed. Our experiment shows that it is possible to maintain good classification accuracies by adding new speech recognition results of user's actual utterances with the corresponding utterance classes. We also show that transcribing user's actual utterances, which must be done by humans, is not necessary to maintain the classification accuracies. Therefore, we indicate a future possibility to utilize the proposed system with an automatic adaptation technique to cloud ASR's outputs.

2. Related Research

Many researchers have reported cloud ASR systems that have been put into practical use. Schalkwyk et al. proposed a system with a large-scale language model and acoustic model [1]. Kamado et al. reported a cloud ASR specialized for noise robustness [5]. Other researchers reported practical services that utilize cloud ASRs [6], [7]. However, they did not discuss methods for utilizing cloud ASRs in various applications.

There is little research on how to utilize cloud ASRs for specific applications. Twiefel et al. reported a method for improving the speech recognition accuracy of a cloud ASR in a specific task by using the phoneme information obtained from speech recognition results and task knowledge [8]. They did not evaluate how their method performed with language understanding. Homma et al. reported an utterance classification method for car navigation systems with improved accuracy for inputs of error-prone speech-recognized sentences [9]. They utilized speech enhancement and multiple utterance classifiers by choosing the most likely utterance class from the outputs of the classifiers. However, they did not evaluate the effectiveness of the proposed method with real user utterances used on commercial products. In addition, they did not show how their proposed methods should adapt to the changes in output characteristics of a cloud ASR.

Many methods were proposed to improve speech recognition accuracies by combining outputs of multiple ASRs. The techniques to choose a confident speech recognition result include various algorithms from voting to deep learning-based ones [10]–[16]. Another researcher proposed a method to combine multiple language understanding results using multiple ASRs [17]. Most of these methods are assumed to utilize multiple ASRs, each of which uses different algorithms, different acoustic models, or different language models. When multiple ASRs are utilized, it is easy to generate various speech recognition results. However, our combination method must work in a situation where only one single off-the-shelf ASR is used, and this situation is not well-studied yet. Thus, the novel point in ASR combination is to show a combination method which works even when only one ASR is available. To generate various speech recognition results from one ASR, we use speech enhancement to generate various speech signals to be input to the ASR. We also confirm that our method improves speech recognition accuracies.

In this paper, first, we propose a system that utilizes a cloud ASR with a language understanding function for a specific application. Second, we report the effectiveness of our system by using speech data obtained from real users of the application. Finally, we show a method to maintain good language understanding accuracies when output characteristics of a cloud ASR are changed.

Our research dealt with important problems that developers face when a single cloud ASR is used in real applications for a long time. Specifically, we dealt with these problems: how to adapt an off-the-shelf cloud ASR to specific applications, and how to adapt application systems to changes of the cloud ASR. To the best our knowledge, we do not know of any research on these subjects.

3. Proposed System

3.1 System Configuration

A cloud ASR is usually prepared by the vendor of a speech recognition service, so that application developers cannot

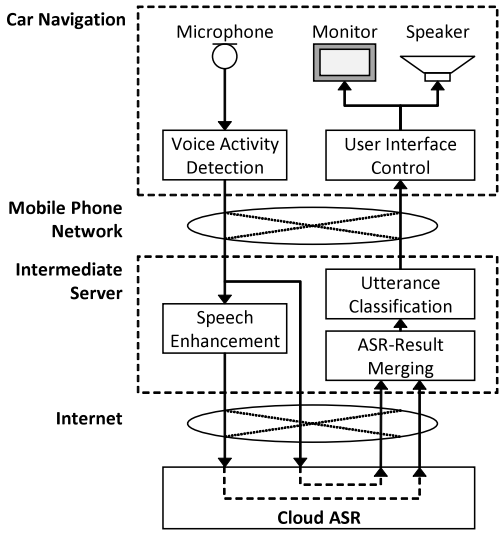

Fig. 1 The system configuration for car navigation systems utilizing a cloud ASR.

customize the internals of the ASR. All the developers can customize are the components between the user and the cloud ASR. Following this limitation, we designed a system for a voice interface used for car navigation that realizes utterance classification that is robust against speech recognition errors without the need to modify a cloud ASR. Figure 1 shows the configuration of the system.

Car navigation systems perform voice activity detection by the statistical noise estimation [18] of sound recorded at a microphone. Speech signals judged as those of a voice are encoded and sent to an intermediate server.

The intermediate server is assumed to be prepared by application developers. This server performs speech enhancement on decoded incoming speech signals and then sends the speech signals to a cloud ASR. Non-enhanced speech signals are also sent. The speech recognition results for both signals, i.e., sentences, are sent back to the server. Then, the server merges the results. Finally, it performs utterance classification that predicts which car navigation function the user wants to execute.

The reason we put speech enhancement and utterance classification on the intermediate server and voice activity detection (VAD) in the car navigation system is as follows. For speech enhancement and utterance classification, we have to consider the possibility that both will need to be modified during operation. For example, if the output characteristics of a cloud ASR are changed, we should adjust the speech enhancement and utterance classification to work well depending on the new output characteristics of the cloud ASR. In addition, our system sends enhanced and non-enhanced speech signals to a cloud ASR. Putting the speech enhancement in a car navigation system would increase network traffic on mobile phone networks, causing a delay in response time. Therefore, we put the enhancement and classification on the intermediate server.

In terms of VAD, we emphasized having the system

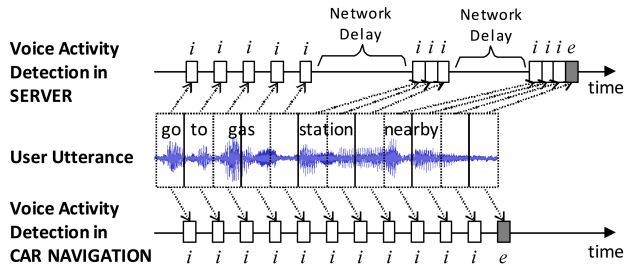

Fig. 2 A comparison of voice activity detection processes in a server and a car navigation system. The voice activity detection judges whether the user is speaking or has finished speaking based on incoming speech signals. The symbol *i* shows that the process judged that the user is speaking (in-speech). The symbol *e* shows that the process judged that the user has already finished speaking (end-of-speech).

respond quickly to a user's voice. To respond quickly, it is necessary to quickly control the beginning and ending of a voice recording along with a user's actual utterance. Figure 2 shows a comparison of VAD processes in either a server or a car navigation system. The VAD module receives a segment of speech signals with a constant time length periodically. For one segment of the speech signals, the VAD module judges whether the user is currently speaking, i.e., in-speech, or if the user has already finished speaking, i.e., end-of-speech. When the VAD module judges end-of-speech, the car navigation system usually notifies the user that the recording has ended by showing an icon on the screen or saying "please wait a second." If we put the VAD module on the server, we must consider the delay of this notification. The speech signals are sent to the server via a mobile phone network. In car navigation systems, users speak voice commands in high-speed driving conditions, which results in the car passing through areas with bad reception during voice operation. This causes a communication delay as shown in Fig. 2. This communication delay eventually causes a delay of the end-of-speech notification to the user. If the delay of the end-of-speech notification happens, the user may think that the system failed to receive her/his voice, and the user might say the voice command again. In this way, the delay of the end-of-speech notification degrades the usability of car navigation systems. Therefore, we decided that the car navigation system handles voice activity detection[†].

3.2 Speech Enhancement and Merging ASR Outputs

Speech enhancement is one popular way to improve the accuracy of speech recognition, e.g., [20]. However, it is difficult to maintain suitable speech enhancement parameters for speech recognition because the parameters vary depending on the noise conditions. In addition, some researchers reported that there are limitations to speech enhancement in terms of improving the accuracy of speech

[†]The voice activity detection we used in this study was also confirmed to work robustly in in-vehicle noisy environments, including the sounds of turn signal flashers [19].

recognition; there are cases in which speech enhancement could not improve accuracy or it sometimes degraded accuracy [21], [22]. Taking these limitations into consideration, we take a hybrid approach in which we obtain speech recognition results for both enhanced and non-enhanced, i.e., raw, speech signals and then merge both results. Thus, we aim to consistently reduce the number of speech recognition errors for all noise conditions.

As shown in Fig. 1, the intermediate server performs speech enhancement on a speech signal coming from a car navigation system. The server sends not only an enhanced signal but also a raw signal to a cloud ASR at the same time. Then, the cloud ASR returns a speech recognition result in the form of N-best sentences for each speech signal. The intermediate server then "merges" the ASR results, in which these N-best sentences are merged.

It is not possible to remove all recognition errors appearing in 1-best sentences by speech enhancement only. Even if a 1-best sentence is not completely correct, we can raise the possibility of correctly estimating which car navigation function a user has specified if at least one of the N-best sentences has correct words. Keeping this point in mind, we designed an algorithm for merging ASR results to reduce the number of word errors appearing in N-best sentences rather than focusing on only 1-best sentences.

Details on the merging are as follows. We assume that two sets of N-best sentences A and B are obtained. One is derived from a raw speech signal, and the other is derived from an enhanced speech signal. Each N-best sentence in A and B is noted as A_i and B_i (i is the N-best rank), respectively. The algorithm sorts these N-best sentences in the order of $A_1, B_1, A_2, B_2, \cdots$. The sorted sentences are then used as a merged recognition result. If there are sentences that are the same among the sorted sentences, we leave only a sentence in the top rank within these duplicated sentences. We define a prioritized speech recognition result (A) as being a result for which the confidence score of a 1-best sentence obtained from a cloud ASR is the largest among the speech recognition results.

As a speech enhancement algorithm for our system, we implemented bidirectional OM-LSA (BOMLSA) speech estimator [23]. OM-LSA [24] is an extension of FFT-based a priori SNR estimator of Ephraim and Malah [25]. Signal gain for a given frequency is optimized to minimize the distortion measure $E\left[(\log X(k) - \log \hat{X}(k))^2\right]$, where $X(k)$ is the spectral amplitude of the k-th frequency bin and $\hat{X}(k)$ is the corresponding estimation. The optimal gain is further modified with respect to the speech presence probability. BOMLSA is a combination of two OM-LSA estimators, one of which employs forward (past to future) estimation and the other employs backward (future to past) estimation. It is known to be more robust than OM-LSA in various environments including a running car.

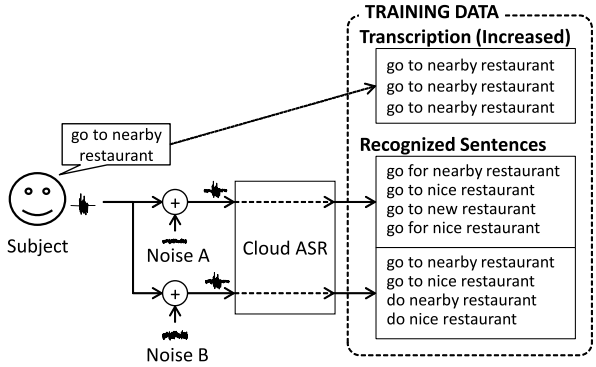

Fig. 3 The method for using recognized sentences and transcriptions as training data for the utterance classifier. This figure assumes the scaling factor r is 3, so one transcription ("go to nearby restaurant") is increased to 3 items in training data (©2016 IEEE [9]).

3.3 Utterance Classification

In this study, we use an utterance classification algorithm with machine learning. A straightforward procedure for developing an utterance classifier based on machine learning is as follows [26]. First, developers make training data for the classifier. The sentences in the data are usually sentences transcribed from participants' utterances in a laboratory experiment for data collection. Wizard-of-Oz is a well-known experimental method for collecting training data [27]. Each training sentence must have a corresponding class label that indicates the user's intent in a target task, i.e., one of the car navigation functions in this study. The class labels are usually annotated by humans. Once we prepare the training data, we can train the utterance classifier by using a machine learning algorithm such as naive Bayes [26], logistic regression [9], or convolutional neural network [28].

If we take this straightforward approach, misrecognized words in a sentence input to the classifier could degrade the accuracy of utterance classification. To provide the classifier with immunity against misrecognized words, we added speech recognition results with misrecognized words to the training data of the classifier. In other words, we "doped" misrecognized words into the training data.

However, as this method adds "noise" to the training data, it is possible that the utterance classifier will fail to predict a correct utterance class from an input sentence that is perfectly correct. This means that a car navigation system could sometimes execute the wrong function even though a user uttered the correct words and the ASR also returned the correct words. The user might become upset in such a situation. To avoid this side effect, we adjusted the number of recognized sentences and transcribed sentences (having no misrecognized words) in the training data. We call this method "optimal doping." With optimal doping, the classifier can achieve good accuracy regardless of whether the input sentence has misrecognized words or not.

Figure 3 shows the method for creating training data with optimal doping. First, we collect subjects' spontaneous

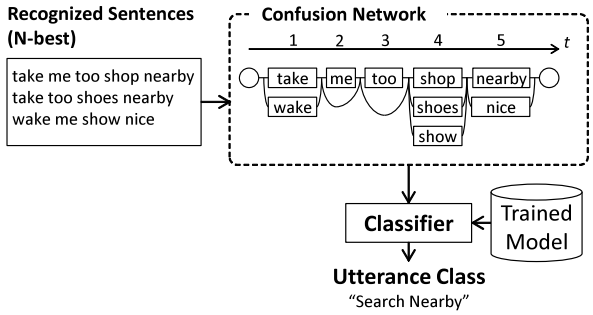

Fig. 4 The feature construction for utterance classification using a confusion network.

utterances when they attempt to execute a car navigation function by voice. Each utterance is labeled with one utterance class that corresponds to the function that the subject wants to execute.

The speech data of the collected utterances are mixed with noise sounds recorded in cars. The mixed speech data are sent to a cloud ASR. The cloud ASR returns a speech recognition result in the form of N-best sentences for each utterance. The recognized sentences are used as the training data for the utterance classifier.

In addition, the transcribed sentences of the collected utterances are also added to the training data. We increase the number of transcribed sentences in the data by a scaling factor. If the scaling factor is 10, 10 sentences that are the same as one transcribed sentence are added to the training data. By using the constructed training data, a multiclass utterance classifier is trained that predicts one utterance class from a given utterance. We use word N-gram features derived from each training sentence to train the classifier.

Figure 4 shows the utterance classification method. For classification, we used N-gram features generated from a confusion network derived from speech recognition results. Utterance classification based on a confusion network robustly works against speech recognition errors [29], [30]. First, we obtain the N-best sentences of a speech recognition result on the basis of the method shown in Sects. 3.1 and 3.2. Second, we convert the N-best sentences into a confusion network by taking the word alignment [10], where the 1-best sentence is regarded as the base word transition network. Third, we obtain word N-grams that appear along with the paths in the network. Last, the obtained word N-grams are then converted to a bag-of-N-gram feature vector. The vector is input to the utterance classifier. The classifier predicts the probabilities of utterance classes. The top one utterance class that has the largest probability is adopted as the prediction result.

A method for calculating feature values, i.e., values of feature vectors, is as follows. We use the confidence scores of recognized words appearing in the confusion network. However, assuming the a normal specification of the cloud ASR [31], [32], we assume that a cloud ASR outputs a confidence score only for the 1-best sentence.

From the 1-best sentence score ($c(s_1) : 0 \leq c(s_1) \leq 1$), we calculate the sentence score of the k-th sentence ($c(s_k) : 2 \leq k \leq N$) as follows.

$$c(s_k) = \min\left(\frac{1 - c(s_1)}{N - 1}, c(s_1)\right) \quad \text{at } 2 \leq k \leq N. \tag{1}$$

N is the total number of N-best sentences. As shown in Eq. (1), the confidence values of sentences from second rank to N-th rank take the same value. From these sentence scores, we calculate the word confidence score of a word w that appeared at time T with the following equation.

$$c(T, w) = \sum_{j=1}^{N} c(s_j)\delta(w|_{t=T} \in s_j). \tag{2}$$

$\delta(w|_{t=T} \in s_j)$ is a function that becomes 1 if sentence s_j yields word w at time T, otherwise 0.

In addition, the confusion network has "null words," which are paths containing no words, e.g., the paths parallel to "me" and "too" in Fig. 4. We also assign word confidence scores to these null words as follows.

$$c(T, \text{null}) = 1 - \sum_{j=1}^{m} c(T, w_j). \tag{3}$$

$c(T, w_j)(1 \leq j \leq m)$ are word confidence scores of all m words appearing at time T.

A feature value $c(w_1 w_2 \cdots w_p)$ of a p-gram consisting of words w_1, w_2, $\cdots$, and w_p appearing at times T_1, T_2, $\cdots$, and T_p is defined as follows.

$$c(w_1 w_2 \cdots w_p)|_{t=T_1,T_2,\cdots,T_p} = \sqrt[p]{\prod_{i=1}^{p} c(T_i, w_i)}. \tag{4}$$

If word N-grams were generated at two or more of the times, the maximum feature value is adopted.

In this study, we used word 1-grams and 2-grams to make a feature vector. 2-grams are generated along paths, whether the paths include null words or not. For example, the confusion network in Fig. 4 generates 2-grams such as "take too" and "take shop." The feature value of "take too" becomes:

$$\sqrt[3]{c(1, \text{"take"})\, c(2, \text{null})\, c(3, \text{"too"})}. \tag{5}$$

Logistic regression was adopted as the algorithm for training the utterance classifier with LIBLINEAR [33]. We used MeCab [34] with NAIST Japanese dictionary [35] to divide a sentence into words.

4. Collecting Experimental Dataset

In a laboratory environment, we collected an experimental dataset that contained data necessary for constructing our proposed system. The dataset contained data on subjects' spontaneous speech utterances spoken to execute specific car navigation functions, transcriptions of the utterances,

their speech recognition results in a cloud ASR, and their labels of car navigation functions. These were used to make the training data for the utterance classifier. In this section, we also evaluate speech enhancement and utterance classification by using the experimental dataset.

4.1 Interview-Based Collection of Experimental Dataset

We collected utterances in an interview. Subjects were told to assume that they were executing one function of a car navigation system with spontaneous utterances [36]. One hundred Japanese people (aged 18–69, 50 males, 50 females) participated in this experiment.

The interviewer first described a situation in which a car driver would want to execute a car navigation function by voice. Then, the interviewer asked each subject what she/he was likely to say to the car navigation system in order to execute the function. The subject was encouraged to answer with two or more utterances for one situation.

All subjects' answers were transcribed. After finishing the interview, the subjects were asked to speak the transcribed answers in order to record speech samples for a speech recognition experiment. This recording was done by using a headset microphone.

The recorded utterances were then labeled with a corresponding utterance class, which was one of the car navigation functions that the subject wanted to execute with an utterance. Finally, we obtained 5,408 utterances with 18 utterance classes.

To simulate noisy environments in cars, we mixed the recorded utterances with road noises that were recorded by driving on a highway (high noise condition: HN) and while idling (low noise condition: LN). Each signal-to-noise ratio (SNR) for mixing was determined to be the same as the actual SNR measured during recording for each condition. We call the utterances mixed with highway noise HN-R (high noise, raw) and those with the idling noise LN-R (low noise, raw). Furthermore, we made HN-E (high noise, enhanced) and LN-E (low noise, enhanced) speech samples by applying a speech enhancement algorithm to HN-R and LN-R, respectively. We used the bidirectional OMLSA algorithm [23] for the enhancement.

These speech samples were input to a cloud ASR to obtain speech recognition results. The maximum number of N-best sentences was 5. The word error rate (WER) for 1-best sentences was 27.7% for HN-R and 15.7% for LN-R. The obtained speech recognition results were used as the experimental dataset. In addition, we also included the transcribed sentences in the experimental dataset.

Prior to making the experimental dataset, we performed another speech recognition experiment by using a speech dataset to determine suitable parameters for speech enhancement. The dataset had utterances in which various names of landmarks were spoken. We mixed them with noises and applied speech enhancement in the same manner as making HN-E speech samples. Then, we input them to the same cloud ASR. The parameters of the speech enhancement were adjusted so as to minimize the WER of 1-best recognized sentences in this experiment.

4.2 Evaluating Speech Enhancement

We evaluated the effect of speech enhancement by using the experimental dataset.

4.2.1 Evaluation Metrics

Usually, speech recognition accuracy is evaluated with a WER calculated from a 1-best recognized sentence. However, our proposed system aims to correctly classify utterances by utilizing N-best sentences rather than only 1-best sentences. To evaluate the effect of speech enhancement with N-best sentences, we incorporated the N-best WER. Given N-best sentences, the N-best WER in the sentence at n-th rank is calculated as follows.

$$\mathrm{WER}(n) = \frac{\sum_{i=1}^{M} \left\{ \min_{j=1,2,\cdots,n} (I_{ij} + D_{ij} + S_{ij}) \right\}}{\sum_{i=1}^{M} W_i} \tag{6}$$

W_i is the number of words in the i-th utterance, and M is the total number of utterances. I_{ij}, D_{ij}, and S_{ij} are numbers of words counted as insertion, deletion, and substitution, respectively, in a recognized sentence at the j-th rank of the recognition result of the i-th utterance.

4.2.2 Evaluation Result

From the speech recognition results of raw speech signals and enhanced speech signals, we merged speech recognition results by using the method shown in Sect. 3.2. We call the merged results from HN-R and HN-E "HN-M" (high noise, merged) and the merged results from LN-R and LN-E "LN-M" (low noise, merged).

Here, we evaluate 1-best to 5-best WERs for each speech recognition result. As described in Sect. 4.1, one speech recognition result for HN-R, HN-E, LN-R, or LN-E has 5-best sentences at most. Therefore, one speech recognition result for either HN-M or LN-M has 10-best sentences at most. To evaluate WERs in HN-M and LN-M, we only use 5-best sentences in these speech recognition results.

Figure 5 shows 1-best to 5-best WERs. In the HN condition, HN-M resulted in a lower WER than HN-R and HN-E for all N-best ranks. In the LN condition, although LN-E caused the WER to increase, LN-M resulted in a lower WER than LN-R and LN-E except the 1-best WER, which had the same WER as LN-R. This means that our merging of ASR results can reduce the WER not only in noisy environments but also in silent environments.

4.3 Evaluating Utterance Classification

We evaluated the effect of the proposed system by evaluating the accuracy of utterance classification when speech

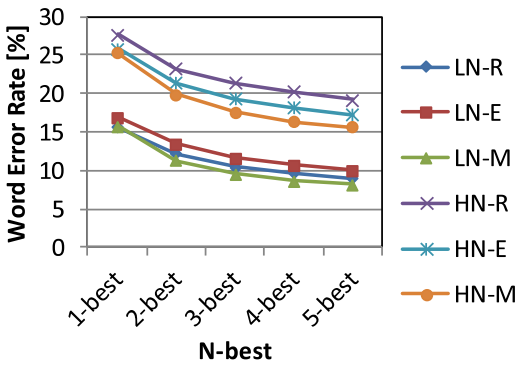

Fig. 5 Word error rates in the experimental dataset.

recognition results are input.

4.3.1 Evaluation Method

Utterance classification was evaluated by subject-based 10-fold cross validation. We divided the experimental dataset into 10 subsets, and each subset contained different subjects' utterances. In the evaluation, nine subsets were used for the training of the utterance classifier, and the remaining one subset was used for evaluation input. By iterating this process 10 times with different combinations of training subsets and an evaluation subset, we evaluated the accuracy of utterance classification for the overall experimental dataset.

The dataset contained utterances for 18 utterance classes, i.e., 18 kinds of car navigation functions. Table 1 shows the utterance number ratio of each utterance class in the experimental dataset. The utterance class of "search destination nearby" is to search a destination around the position of the user's vehicle. Sample utterances with this class are "search gas station nearby" and "I want to go to a restaurant near my car." The class of "search" is to perform a destination search without any particular limitations. Sample utterances with this class are "gas station" and "I want to go to a restaurant."

The method for making training data is as follows. As a baseline condition, we trained the utterance classifier by using only transcribed sentences. When we used speech recognition results as the training data, we trained the classifier by using different combinations of data. The first condition is "raw," in which speech recognition results obtained from HN-R and LN-R are used. The second condition is "enhanced," in which speech recognition results obtained from HN-E and LN-E are used. The last condition is "combined," in which speech recognition results obtained from HN-R, LN-R, HN-E, and LN-E are used. All N-best sentences (5-best maximum) obtained from the cloud ASR were used as the training data. In addition, we further set training conditions in which the training data were a mix of speech recognition results and transcribed sentences, where the scaling factor r was 1, 10, 100, and 1,000.

The evaluation data were as follows. When the data were speech recognition results, we set two conditions. The first is that a confusion network converted from N-best recognized sentences is input to the utterance classifier. When the data were HN-R, HN-E, LN-R, and LN-E, the maximum N-best number was 5. HN-M data were made by merging HN-R and HN-E, and LN-M data were made by merging LN-R and LN-E. Therefore, the maximum N-best number of HN-M and LN-M was 10. The second condition is that a confusion network converted from only a 1-best sentence (with a confidence score of 1) is input. Furthermore, we made another condition in which the transcribed sentences, i.e., sentences with no word errors, were input to the utterance classifier. In this condition, we assumed that a confusion network converted from a 1-best sentence (with a confidence score of 1), which is the same as a transcribed sentence, is input to the utterance classifier. The maximum dimension of an input feature vector was 13k.

Table 1 The utterance class distribution in the experimental dataset (©2016 IEEE [9]).

Class	Ratio [%]
search	34.4
search nearby	8.4
search around destination	3.1
search along route	2.0
search in nation	2.8
change search area to neighborhood of current position	3.1
change search area to neighborhood of destination	2.3
change search area to nation	2.6
show more search result	4.3
go home	2.8
go back to previous screen	5.0
quit voice operation	4.4
yes	8.7
no	8.1
reset	2.9
start guidance	2.3
say search results	2.0
search again	0.9

4.3.2 Evaluation Result

We evaluate utterance classifiers by using the classification error rate (CER). The CER is calculated as $M_{\text{mis}}/M_{\text{total}}$, where M_{total} is the total number of utterances in the evaluation data, and M_{mis} is the number of misclassified utterances. From the utterance class distribution of the experimental dataset (Table 1), the chance rate of the CER is 65.6% if the classifier always estimates "search" which is the most frequent class in the data.

Table 2 shows CERs of the utterance classifier. There were three conditions for training data: (a) transcribed sentences only, (b) recognized sentences only, and (c) both transcribed sentences and recognized sentences. For cases (b) and (c), CERs in Table 2 were obtained when the evaluation input was N-best. However, for case (a), CERs in Table 2 were obtained when the evaluation input was 1-best because these CERs were lower than the CERs for the N-best input conditions.

If the evaluation inputs were speech recognition results, lower CERs were obtained when the training data were a mix of speech recognition results and transcribed sentences.

Table 2 Classification error rates (CERs) in evaluation using the experimental dataset [%]. r is the scaling factor. Bolded numbers show the lowest CER for each evaluation condition (Trans.: transcriptions, Recog.: recognized sentences, Enh.: enhanced, Cmb.: combined).

Evaluation Data	Training Data															
	(a) Trans.	(b) Recog.			(c) Recog. + Trans.											
		Raw	Enh.	Cmb.	Raw				Enh.				Cmb.			
					r=1	r=10	r=10^2	r=10^3	r=1	r=10	r=10^2	r=10^3	r=1	r=10	r=10^2	r=10^3
Trans.	**4.0**	6.3	5.8	6.2	5.5	4.9	4.4	4.2	5.1	4.7	4.2	**4.0**	5.5	5.2	4.6	4.2
HN-R	21.0	15.2	14.9	14.3	15.0	14.7	14.9	17.0	14.8	14.7	14.8	17.0	14.3	14.0	14.0	15.9
HN-E	19.0	12.5	12.5	11.8	12.4	12.3	12.6	14.8	12.4	12.3	12.4	14.6	11.8	11.5	11.8	13.5
HN-M	18.5	12.0	12.1	11.3	11.7	11.7	12.1	14.5	12.0	11.7	12.0	14.5	11.3	11.2	**11.1**	13.3
LN-R	13.4	8.2	8.2	8.0	8.0	7.8	7.9	9.7	8.1	8.0	7.9	9.6	7.9	7.6	7.3	8.5
LN-E	14.4	8.8	8.6	8.2	8.8	8.5	8.5	10.5	8.4	8.3	8.4	10.2	8.3	8.0	7.7	9.2
LN-M	13.3	7.9	8.0	7.7	7.9	7.7	7.7	9.8	7.8	7.7	7.9	9.6	7.7	7.4	**7.1**	8.8

The lowest CERs were obtained when the training data condition was the combined condition with a scaling factor (r) of 100 and the evaluation input was the merged condition (LN-M or HN-M). The reason a lower CER was obtained by increasing the scaling factor is that a bigger scaling factor can successfully leverage correct words existing in N-best sentences to predict correct utterance classes.

We examined the CER when the evaluation inputs were transcribed sentences. The CERs were increased more by using recognized sentences as training data than when only transcriptions were used as training data. When the scaling factor was increased, however, the rise in CER was mitigated. When the scaling factor was 100, the amount of increase from the condition when the training data were only transcribed sentences was just 0.2–0.6%. These results show optimal doping could avoid the side effect of the CER increasing for the transcription input.

5. Evaluation with Real User Logs

We conducted an evaluation of the proposed system by using user logs in which actual car navigation users uttered phrases to commercial car navigation products used daily.

5.1 User Logs

The user logs we used were speech utterances of actual users obtained from a commercial-use speech recognition service for car navigation, named "Clarion Intelligent VOICE" [19], [37], [38]. This service utilizes a cloud ASR to provide a speech interface that users can use to operate car navigation functions with spontaneous utterances. We evaluated our system by using 8,717 Japanese utterances obtained from this service. The utterances were processed to make enhanced speech samples by using the method shown in Sect. 4. All raw and enhanced speech samples were sent to the same cloud ASR in the same manner as the experiment in Sect. 4. The speech recognition results were obtained over a one month period 1.5 years after the recognition results of the experimental dataset were obtained. All the user logs we used were anonymized before we started the research.

Human annotators transcribed all utterances. The annotators also labeled correct utterance classes, i.e., a car navigation function that the user wants to execute with an utterance, for all utterances.

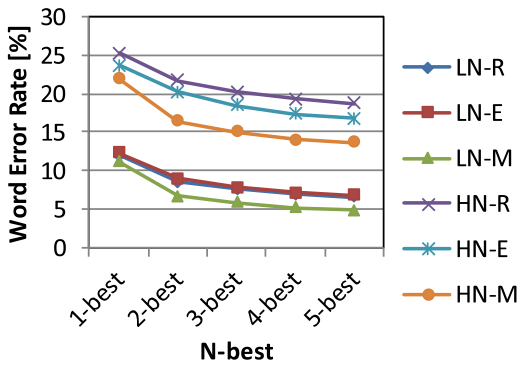

Fig. 6 Word error rates in the user logs.

5.2 Evaluating Speech Enhancement

As we showed in Sect. 4.2, the effect of the speech enhancement varied depending on the background noise. Therefore, we first estimated the SNR of each utterance in the user logs and then examined the WER reduction when the SNR was lower than 10 dB (HN: high noise condition, 3,556 utterances) and when it was higher than 10 dB (LN: low noise condition: 5,161 utterances). We used the WADA algorithm [39] to estimate SNR.

Figure 6 shows the WERs for the user logs, which show WERs when no enhancement was done (R: raw), when it was done (E: enhanced), and when the recognition results of the raw signal and the enhanced signal were merged (M: merged).

For both HN and LN conditions, the lowest WERs were obtained for the merged conditions (HN-M or LN-M). This shows that our proposed system could successfully reduce WERs for real user utterances in commercial car navigation systems. We calculated the WER reduction rate of the 5-best WERs from the raw to merged conditions, which resulted in a word error reduction of 26.0% (6.6% to 4.9%) for LN and 27.4% (18.7% to 13.6%) for HN.

The speech data in the user logs contain reverberation in a car room, whereas the speech data in the experimental dataset does not. Therefore, we suspected that WERs in

Table 3 The utterance class distribution in the user logs used for utterance classification evaluation.

Class	Ratio [%]
search	71.3
search nearby	7.1
search around destination	0.1
search along route	0.3
show more search result	4.4
go home	5.7
start guidance	11.1

the user logs might be higher than ones in the experimental dataset. However, the reverse was true: 1-best WERs in LN-R condition are 15.7% for the experimental dataset, and 11.3% for the user logs. We investigated reasons for this.

One possible reason is the time gap of obtaining speech recognition results: the results of the user logs were obtained 1.5 years after the results of the experimental dataset were obtained. During this 1.5 years, the cloud ASR might have improved in terms of accuracies. To validate this reason, we calculated WERs in utterances that were commonly spoken in both the experimental dataset and the user logs. The 1-best WERs in LN-R condition of these utterances resulted in 10.3% for the experimental dataset, and 12.8% for the user logs. These WERs have no significant difference ($p = 0.09$; Z-test [40]). Therefore, this reason seems incorrect.

Another possible reason is the difference in uttered sentences between the experimental dataset and the user logs. Here, we consider dividing words in an utterance into a POI query and other words. The POI query is the place name appearing in the utterance that a user wants to search. When an uttered sentence is "go to an Italian restaurant nearby," the POI query is "an Italian restaurant." Here, we refer to the words of the POI query as IP (Inside of POI), and we refer to the rest, e.g., "go to" and "nearby," as OP (Outside of POI). The experimental dataset was collected by an interview-based corpus creation method, which tends to collect more various kinds of OP words than ones that appeared in actual user utterances to car navigation products [36]. In the utterances with two major utterance classes, "search" and "search nearby," the ratios of OP words are 39% for the experimental dataset, and 11% for the user logs. This value demonstrates that there is a clear difference of the uttered sentences between the experimental dataset and the user logs. We presume that the difference in uttered sentences makes the difference of WERs. In other words, the language model in the cloud ASR we used seems to fit the actual utterances in car navigation products more closely.

5.3 Evaluating Utterance Classification

Next, we evaluated the utterance classification. Actual user logs contain various utterances spoken to execute the various car navigation functions of actual products. However, our user logs contained highly skewed data; the distribution in numbers of utterances for each car navigation function was highly uneven. Therefore, we chose a number of functions in which the numbers of utterances were sufficient

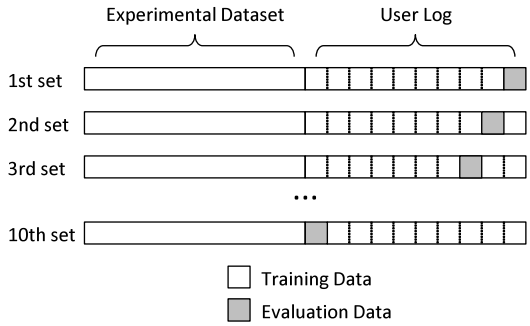

Fig. 7 The data splitting method in the utterance classification evaluation when both experimental dataset and user logs were used as training data.

enough to evaluate our proposed system. Specifically, from all 8,717 utterances, we extracted 6,164 utterances that belonged to the 7 utterance classes shown in Table 3. The 1-best WER for these 6,164 utterances in the merged condition was 14.8%.

We made the training data for the utterance classifier not only from the experimental dataset obtained in Sect. 4 but also from the user logs for the following reason.

The recognition results for the dataset and logs were obtained from the same cloud ASR but at a different time. A cloud ASR might be updated during this time gap. As application developers cannot know when such an update happens, the ideal way to make training data for a classifier is to send a speech dataset to the cloud ASR frequently and to update the training data of the classifier with new recognition results. However, it is likely not possible to send thousands of speech samples to a cloud ASR frequently and to replace the training data because this requires a lot of transaction fees for a cloud ASR and labor time. It might be better if the previous recognition results were to work fairly well as the training data of the classifier. Thus, we evaluated the classifier when the training data consisted of the previous speech recognition results, i.e., the experimental dataset, and the current recognition results, i.e., the user logs, and both.

When the training data included the user logs, the evaluation was done by user-based 10-fold cross validation, in which each subset of the user logs had different user utterances. When the training data consisted of both the experimental dataset and user logs, as shown in Fig. 7, the experimental dataset was used as common training data among all subsets' evaluations. The number of utterances of the experimental dataset in Sect. 4 was 5,408. From them, we extracted 3,101 utterances for training corresponding to the utterance classes in Table 3. The maximum dimension of an input feature vector was 19k. The chance rate of the CER is 28.7% if the classifier always estimates "search" which is the most frequent class in the data.

Table 4 shows CER results. Following the results of Sect. 4, the CERs when the training data contained speech recognition results were obtained by using the evaluation inputs in the merged condition with N-best inputs. Meanwhile, the CERs when the training data were only transcribed sentences were obtained by using the evaluation in-

Table 4 Classification error rates [%] in evaluation of the user logs. r is the scaling factor. Asterisks mean significant differences with $^{*}p < 0.01$ and $^{**}p < 0.001$ from the reference condition indicated by † symbol (sign test). Bolded is the lowest CER among each section. ‡ symbol shows the condition where the summed CER of a recognition input and a transcription input was the lowest (Trans.: transcriptions, Recog.: recognized sentences).

Training Data					**Evaluation**	
Trans.		Recog.			Recog.	Trans.
Dataset	Log	Dataset	Log	r		
✓					7.6	0.6
	✓				6.9	0.7
✓	✓				† **6.5**	**0.1**
		✓			6.8	2.9
			✓		** **5.2**	**1.5**
		✓	✓		** **5.2**	2.1
✓		✓		1	6.7	2.7
✓		✓		10	6.2	2.4
✓		✓		100	** **5.8**	1.2
✓		✓		1,000	* 6.1	**0.4**
✓			✓	1	** 4.7	0.9
✓			✓	10	‡ ** **4.6**	0.4
✓			✓	100	‡ ** 4.8	**0.2**
✓			✓	1,000	** 5.2	0.3
	✓	✓		1	* 6.0	1.9
	✓	✓		10	** **5.5**	1.2
	✓	✓		100	** 5.8	0.5
	✓	✓		1,000	6.6	**0.3**
	✓		✓	1	** **5.0**	1.3
	✓		✓	10	** **5.0**	1.0
	✓		✓	100	** 5.2	**0.5**
	✓		✓	1,000	6.3	**0.5**
✓	✓	✓		1	* 5.9	1.8
✓	✓	✓		10	** **5.5**	1.2
✓	✓	✓		100	** 5.7	0.2
✓	✓	✓		1,000	6.6	**0.1**
✓	✓		✓	1	** **4.7**	0.8
✓	✓		✓	10	‡ ** **4.7**	0.3
✓	✓		✓	100	** 4.9	0.2
✓	✓		✓	1,000	** 5.6	**0.1**
✓	✓	✓	✓	1	** 5.1	1.9
✓	✓	✓	✓	10	** **4.9**	1.4
✓	✓	✓	✓	100	** **4.9**	0.3
✓	✓	✓	✓	1,000	** 5.5	**0.1**

puts in the merged condition with 1-best inputs.

First, we focus on the CERs when the training data were only transcribed sentences. By using both the experimental dataset and user logs as the training data, we obtained the lowest CER (6.5%). The reason we obtained a lower CER for these training data is that the variation in words in the training data was increased by using both types of data, so the training data had a broader word coverage for the users' actual user utterances [36].

Second, we focus on CERs when the training data contained speech recognition results. We obtained the lowest CER (4.6%) when the training data were the combination of transcriptions of the experimental dataset and speech recognition results of the user logs. Also, the scaling factor for the lowest CER was 10. This lowest CER was significantly lower than the CER in the reference condition (6.5%, as shown by the † symbol in Table 4), where the classifier was trained with only transcriptions ($p < 0.001$, sign test). In addition, when the training data contained the recognition results of user logs without the recognition results of the experimental dataset, 12 out of 13 conditions showed a significant CER decrease from the reference condition.

When the training data contained not the recognition results of the user logs but those of the experimental dataset, we observed relatively weak CER reduction effects. Specifically, 8 out of 13 conditions in this setting showed a significant CER decrease from the reference condition.

Third, we examine the accuracy of utterance classification on the transcription input. As we mentioned in Sect. 3.3, we should decrease the CER for recognition inputs while avoiding an increase in CER when the input has no misrecognized words, i.e., transcriptions. To evaluate the total performance of utterance classification, we simply summed two CERs when the evaluation input was speech recognition results and when the input was transcriptions. The lowest summed CER (5.0%) was obtained for the three conditions shown with the ‡ symbol in Table 4. Among these conditions, we assumed one "adopted" condition where the training data were transcriptions of the experimental dataset and recognition results of the user logs with a scaling factor of 100. This adopted condition showed a CER of 4.8% for recognition result inputs and 0.2% for transcription inputs. This adopted condition could successfully reduce CERs for recognition inputs significantly from the reference condition ($p < 0.001$, sign test). At the same time, the CER increase for transcription inputs was just 0.1% from the reference condition.

Finally, we compare these CERs with a condition in which the utterance classifier was made by a straightforward development method. In this condition, specifically, the training data were only transcribed sentences in the experimental dataset. In the evaluation, we input 1-best speech recognition sentences with no speech enhancement. The CER in this condition was 10.4%. The classification error reduction rate for the adopted condition (4.8%) reached 54%.

6. Discussion

As we showed, our proposed system improved the performance of utterance classification on real user logs for car navigation with a cloud ASR. In this section, we review the evaluation results and discuss how to apply our proposed system to actual product services.

6.1 Effectiveness of Laboratory Data and Actual User Data for Utterance Classifier Training

The utterance classification results shown in Table 4 revealed that the lowest CER was observed when the recognition results for the user logs were included in the training data. In comparison, the recognition results of the experimental dataset had less of an effect on CER reduction.

Although both recognition results were obtained from the same cloud ASR, user-log speech recognition results were obtained 1.5 years after the recognition results of

the experimental dataset were obtained. Therefore, we assume that the characteristics of the speech recognition errors changed during this time gap. To prove this assumption, we investigated the similarity in the characteristics of speech recognition errors among four clusters as follows.

(a) Recognition results of experimental dataset obtained from 50 subjects' utterances in HN-M condition (Dataset-H)

(b) Recognition results of experimental dataset obtained from 50 subjects' utterances in LN-M condition (Dataset-L)

(c) Recognition results of user logs in merged condition in first half period (Log-1)

(d) Recognition results of user logs in merged condition in second half period (Log-2)

The subjects who uttered speech samples were different between (a) and (b). The users in (c) and (d) were also different from each other. We investigated the similarity in the characteristics of speech recognition errors among these clusters as follows. We first enumerated the same utterances (u_i) that were commonly uttered for all of the clusters. For each sentence u_i, we calculated the similarity of speech recognition errors between cluster X and cluster Y ($\mathrm{sim}(X, Y, u_i)$) with the following equation.

$$\mathrm{sim}(X, Y, u_i) = \frac{\sum_{j=1}^{m} \left(f_x(w_j) f_y(w_j) \right)}{\sqrt{\sum_{j=1}^{m} f_x(w_j)^2} \sqrt{\sum_{j=1}^{m} f_y(w_j)^2}} \tag{7}$$

$w_1, w_2, \cdots, w_m$ are misrecognized words appearing in cluster X, cluster Y, or both. $f_x(w_j)$ and $f_y(w_j)$ indicate numbers of the occurrences of word w_j in the recognition results of utterance u_i in cluster X and cluster Y, respectively. The word counting was done for all words appearing in the 10-best recognized sentences. The value of Eq. (7) goes to 0 if no same misrecognized words exist in cluster X and cluster Y. It goes to 1 if both X and Y have the same misrecognized words and same distributions of occurrence probability for each misrecognized word.

Our hypothesis is that the similarity becomes small if two clusters of speech recognition results were obtained in different periods, and the similarity becomes large if two clusters of the recognition results were obtained in a same period. Thus, our expectation is that the similarities of (a) vs. (c), (a) vs. (d), (b) vs. (c), and (b) vs. (d) become small, and the similarities of (a) vs. (b) and (c) vs. (d) become large.

Table 5 shows the similarities between the clusters. As this table indicates, the similarities within experimental datasets or within the user logs were high (0.84–0.85), while the similarities between the experimental datasets and the user logs were relatively low (0.55–0.60). From this result, we confirmed that the characteristics of speech recognition errors are different between the experimental dataset and the user logs. This result indicates that our proposed system performs with good utterance classification accuracy if the recognized sentences in the training data and in the evaluation input have similar characteristics of speech recognition errors. Otherwise, if the training data and the evaluation input have different word error characteristics, our proposed system achieves limited improvement in terms of utterance classification accuracy.

Table 5 Similarities in the speech recognition errors among the experimental dataset and the user logs. These are shown as averages and 95% confident intervals.

	(b) Dataset-L	(c) Log-1	(d) Log-2
(a) Dataset-H	0.84 ± 0.16	0.57 ± 0.17	0.60 ± 0.11
(b) Dataset-L	-	0.55 ± 0.18	0.55 ± 0.15
(c) Log-1	-	-	0.85 ± 0.08

6.2 How to Use Proposed System in Commercial Systems

As we discussed in Sect. 6.1, we figured out that it is important for the recognized sentences in the training data of the utterance classifier to have characteristics similar to the speech recognition errors of real users. Cloud ASRs are continually updating their models and algorithms to continuously improve the speech recognition performance. Therefore, the recognized sentences in the training data should also be updated with up-to-date speech recognition results of the cloud ASR. In addition, speech enhancement with fixed parameters, which were best-tuned at one point in time, might degrade speech recognition accuracy in the near future due to updates of the cloud ASR. Therefore, the enhancement should be also updated following these updates.

With this in mind, we discuss a proper configuration for deploying our proposed system for real use. We believe that our system should be combined with a monitoring scheme that detects changes in the output characteristics of a cloud ASR.

The parameters of the speech enhancement should be adjusted to maximize the speech recognition accuracy for users' utterances. To adjust parameters, it is necessary to prepare an "evaluation set" that contains a fairly small number of previous users' utterances transcribed manually. By sending the evaluation set to the cloud ASR periodically, the monitoring scheme can evaluate the current speech recognition accuracy. If it detects that accuracy has been degraded, it can adjust the parameters of the speech enhancement so as to maximize the accuracy in the evaluation set.

Once current speech recognition results are obtained from the evaluation set, they can be also utilized to monitor the accuracy of utterance classification by inputting the recognition results to the classifier. If our system detects that the accuracy of the classification has degraded, it can replace the training data of the classifier from old speech recognition results to current speech recognition results. The experiment results in Sect. 5 revealed that the lowest CER was achieved when the training data contained transcriptions of the experimental dataset and the recognition results of the user logs. This means that no transcription for the user logs is needed

to train the classifier to achieve the lowest CER. Therefore, we can improve the accuracy of utterance classification by just including current speech recognition results for user utterances in the training data.

The fact that we do not need manually transcribed user logs for training the classifier brings the possibility of automatically improving utterance classification by using real user logs we obtain every day. The only missing parts are the correct utterance classes of user utterances. Basically, correct utterance classes should be labeled manually. Meanwhile, research on dialog systems showed the possibility of automatically improving language understanding by utilizing clues extracted from dialog histories between a user and an agent [41], [42]. One piece of future work is to realize such automatic improvements for utterance classification in in-vehicle systems.

In this automatic updating scenario, we assume that an updated classification model is deployed to a server which provides the utterance classification service to users. To deploy the model, we must take 3 steps: stopping the utterance classification service, replacing the model, and restarting the service. This stopping step causes a problem of service downtime, in which users cannot utilize the utterance classification service. However, we can avoid causing this service downtime by designing the utterance classification service with redundancy. Specifically, we will adopt a multi-cluster configuration, where the same utterance classification service runs on multiple clusters. In addition, we will use a load balancer which first receives a request for utterance classification and then forwards the request to one of the clusters. In this configuration, we can update the model in a single cluster whereas the other clusters continue to provide the service. Once we finish updating the model in one cluster, we move to update the model in another cluster. In this way, we can update the utterance classification model with no downtime.

6.3 How to Increase Number of Utterance Classes for Real Products

Recent car navigation products support a tremendous number of functionalities including not only car navigation but also hands-free communication, and audio functions. Thus, the utterance classifier has to classify more than 100 classes. Meanwhile, the number of classes in this study is 18 at most. We must extend this number to over 100.

However, it is not straightforward to classify so many classes. The biggest difficulty in classifying many classes is that the machine learning-based classifiers must be trained from a large amount of training data for each class. Collecting the training data is time-consuming and costly. Indeed, recent research on utterance classifiers, even though they use state-of-the-art deep learning-based techniques to construct utterance classifiers, were conducted in conditions where the number of utterance classes was less than 50 [28], [43], [44].

One method to solve this difficulty is to use an utterance classification method which combines a machine learning-based classifier and a rule-based classifier, as proposed in [45]. For the rule-based classifier, the developers first make rules which show salient phrases appearing in utterances having a particular class. Then, the rule-based classifier determines an utterance class by judging whether the utterance matches the phrases. The rule-based classifier can classify all the classes if the developer prepares at least one phrase for each class. However, rule-based classifiers usually have lower accuracies than machine learning-based classifiers. Therefore, to develop an utterance classifier not only having good accuracies but also being developed with less cost, it is a good choice that utterance classes which all users frequently speak will be classified by a machine learning-based classifier, and the classes that the users infrequently speak will be classified by a rule-based classifier.

Another choice is a 2-step approach as follows. In the first step, an initial dialog system is launched using a rule-based utterance classifier. In the second step, the system alters the classifier from a rule-based one to a machine learning-based one when abundant number of training data are collected from real users. In order to realize this 2-step approach, we proposed a language understanding method that chooses a suitable output either from rule-based or machine learning-based language understanding modules depending on the amount of training data [46]. We confirmed that this method achieves good language understanding accuracies whether training data are large or small. One of our future studies is to utilize this method for the utterance classifier which classifies many classes required in car navigation products.

6.4 Reducing Response Time for Improved Usability

Finally, we focus on reduction of the response time: the time length from the end of a user utterance to the start of a corresponding system response. In this paper, we proposed a system where all the speech recognition processes are done by a cloud server. An important advantage of this system is to achieve highly-accurate ASR. However, as explained in Sect. 3.1, this configuration sometimes causes longer response time than the system with an embedded speech recognizer.

Miller [47] showed a criterion that the response time in conversation systems should be within 2 seconds. If we use an embedded speech recognizer for a car navigation system, we can satisfy this criterion for voice commands for most car navigation functionalities. However, when we use a cloud ASR for a car navigation system, the response time varies depending on the status of network communications. This sometimes makes the response time more than 5 seconds. Therefore, we are still on the way to maximize user satisfaction by reducing the response time.

To reduce the response time, we think one good configurations is a hybrid ASR, where we send incoming speech signals to not only a cloud ASR but also an embedded speech recognizer at the same time [48]. The basic idea be-

hind the hybrid ASR is that the user's expectation on the response time is different depending on the car navigation functionalities that the user intends to execute. Most users expect very quick responses when they intend basic functionalities, e.g., zooming in/out on a map, turning on/off the audio, and going back to the previous screen. Meanwhile, users may allow a longer response time when the users intend a functionality that needs complicated processes, e.g., destination search and music search. Keeping this in mind, we propose a new module in a car navigation system which classifies whether the user's intent of an incoming utterance requires a quick response or not. If this classification result indicates that it requires a quick response, the car navigation system executes a function determined from the speech recognition result from the embedded ASR. Otherwise, the car navigation system waits for the speech recognition result from the cloud ASR. This configuration of the hybrid ASR can improve user satisfaction by reducing response time. One of our future studies is to validate the efficiency of the hybrid ASR configuration.

7. Conclusion

In this paper, we proposed a novel in-vehicle voice system with improved accuracy in utterance classification under the condition that the connected cloud ASR is a black box and cannot be modified. Our proposed system includes speech enhancement and utterance classification that makes a voice interface robust against speech recognition errors without requiring that the internals of the cloud ASR be modified. Evaluation results using actual user utterances showed that our system reduces the number of utterance classification errors by 54% from a baseline condition. Furthermore, we proposed "optimal doping," in which the training data of a classifier are constructed by using both speech recognition results and transcriptions. Optimal doping suppressed the increase in errors for accurate transcription inputs to just 0.1% while improving classification accuracy for recognized sentence inputs. Finally, we showed a method to maintain good utterance classification accuracies by just including current speech recognition results for user utterances in the training data.

The next studies are as follows. The first will be on validating our methods using various cloud ASR services. The second will be on establishing automatic improvement for utterance classification through the use of actual user logs. The third will be on increasing the number of utterance classes that our methods can deal with. The last will be on reducing response time by incorporating hybrid ASR to improve usability.

References

[1] J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, and B. Strope, "Your word is my command: Google search by voice: A case study," Advances in Speech Recognition: Mobile Environments, Call Centers and Clinics, ed. A. Neustein, pp.61–90, Springer, New York, 2010.

[2] Y. Fujita, R. Takashima, T. Homma, and M. Togami, "Data augmentation using multi-input multi-output source separation for deep neural network based acoustic modeling," Proc. Interspeech, San Francisco, USA, pp.3818–3822, Sept. 2016.

[3] H. Kokubo, A. Amano, and N. Hataoka, "Robust speech recognition for car environment noise," IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J83-DII, no.11, pp.2190–2197, Nov. 2000.

[4] J.R. Bellegarda, "Statistical language model adaptation: Review and perspectives," Speech Commun., vol.42, pp.93–108, Jan. 2004.

[5] N. Kamado, S. Fujimura, Y. Iwase, Y. Aono, H. Masataki, T. Yamada, and R. Otsuya. "Introduction of noise-robust ASR platform based on HTML5," IPSJ SIG Technical Reports, 2015-SLP-108 (3), pp.1–6, Oct. 2015. (in Japanese)

[6] J.R. Bellegarda, "Large-scale personal assistant technology deployment: the Siri experience," Proc. Interspeech, Lyon, France, pp.2029–2033, Aug. 2013.

[7] R. Sarikaya, "The technology behind personal digital assistants: An overview of the system architecture and key components," IEEE Signal Process. Mag., vol.34, no.1, pp.67–81, Jan. 2017.

[8] J. Twiefel, T. Baumann, S. Heinrich, and S. Wermter, "Improving domain-independent cloud-based speech recognition with domain-dependent phonetic post-processing," Proc. AAAI, Québec, Canada, pp.1529–1535, July 2014.

[9] T. Homma, K. Shima, and T. Matsumoto, "Robust utterance classification using multiple classifiers in the presence of speech recognition errors," Proc. IEEE SLT, San Diego, USA, pp.369–375, Dec. 2016.

[10] J.G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," Proc. IEEE ASRU, Santa Barbara, USA, pp.347–354, Dec. 1997.

[11] T. Utsuro, Y. Kodama, T. Watanabe, H. Nishizaki, and S. Nakagawa, "Combining outputs of multiple LVCSR models by machine learning," Syst. Comput. Jpn., vol.36, no.10, pp.9–15, Sept. 2005.

[12] D. Hillard, B. Hoffmeister, M. Ostendorf, R. Schlüter, and H. Ney, "iROVER: Improving system combination with classification," Proc. NAACL-HLT, Rochester, USA, pp.65–68, April 2007.

[13] S. Li, Y. Akita, and T. Kawahara, "Semi-supervised acoustic model training by discriminative data selection from multiple ASR systems' hypotheses," IEEE/ACM Trans. Audio Speech Lang. Process., vol.24, no.9, pp.1524–1534, Sept. 2016.

[14] V. Soto, O. Siohan, M. Elfeky, and P.J. Moreno, "Selection and combination of hypotheses for dialectal speech recognition," Proc. ICASSP, Shanghai, China, pp.5845–5849, March 2016.

[15] Y. Fujita, R. Takashima, T. Homma, R. Ikeshita, Y. Kawaguchi, T. Sumiyoshi, T. Endo, and M. Togami, "Unified ASR system using LGM-based source separation, noise-robust feature extraction, and word hypothesis selection," Proc. ASRU, Scottsdale, USA, pp.416–422, Dec. 2015.

[16] N. Sawada and H. Nishizaki, "Recurrent neural network-based phoneme sequence estimation using multiple ASR systems' outputs for spoken term detection," Proc. Interspeech, San Francisco, USA, pp.3688–3692, Sept. 2016.

[17] M. Katsumaru, M. Nakano, K. Komatani, K. Funakoshi, T. Ogata, and H.G. Okuno, "Improving speech understanding accuracy with limited training data using multiple language models and multiple understanding models," Proc. Interspeech, pp.2735–2738, Brighton, United Kingdom, Sept. 2009.

[18] Y. Obuchi, R. Takeda, and N. Kanda, "Voice activity detection based on augmented statistical noise suppression," Proc. APSIPA ASC, Hollywood, USA, Dec. 2012.

[19] Y. Obuchi, "Speech processing for car navigation systems," Technical Report of IEICE, EA 114(274), pp.3–8, Oct. 2014. (in Japanese)

[20] W. Zhu and D. O'Shaughnessy, "Using noise reduction and spectral emphasis techniques to improve ASR performance in noisy conditions," Proc. IEEE ASRU, St. Thomas, USA, pp.357–362, Nov.-Dec. 2003.

[21] X. Cui and A. Alwan, "Noise robust speech recognition using feature

compensation based on polynomial regression of utterance SNR," IEEE Trans. Speech Audio Process., vol.13, no.6, pp.1161–1172, Nov. 2005.
[22] L. Deng, A. Acero, M. Plumpe, and X. Huang, "Large-vocabulary speech recognition under adverse acoustic environments," Proc. ICSLP, Beijing, China, pp.806–809, Oct. 2000.
[23] Y. Obuchi, R. Takeda, and M. Togami, "Noise suppression method for preprocessor of time-lag speech recognition system based on bidirectional optimally modified log spectral amplitude estimation," Acoust. Sci. & Tech., vol.34, no.2, pp.133–141, March 2013.
[24] I. Cohen and B. Berdugo, "Speech enhancement for non-stationary noise environments," Signal Process., vol.81, no.11, pp.2403–2418, Nov. 2001.
[25] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator," IEEE Trans. Acoust. Speech Signal Process., vol.32, no.6, pp.1109–1121, Dec. 1984.
[26] C. Chelba, M. Mahajan, and A. Acero, "Speech utterance classification," Proc. ICASSP, Hong Kong, China, pp.I-280–I-283, April 2003.
[27] C.T. Hemphill, J.J. Godfrey, and G.R. Doddington, "The ATIS spoken language systems pilot corpus," Proc. 3rd DARPA Speech and Natural Language Workshop, Hidden Valley, USA, pp.96–101, June 1990.
[28] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," Proc. ACL, Baltimore, USA, pp.655–665, June 2014.
[29] M. Henderson, M. Gašić, B. Thomson, P. Tsiakoulis, K. Yu, and S. Young, "Discriminative spoken language understanding using word confusion networks," Proc. IEEE SLT, Miami, USA, pp.176–181, Dec. 2012.
[30] D. Hakkani-Tür, F. Béchet, G. Riccardi, and G. Tur, "Beyond ASR 1-best: Using word confusion networks in spoken language understanding," Comput. Speech Lang., vol.20, no.4, pp.495–514, Oct. 2006.
[31] IBM, "Speech to text: API reference," https://www.ibm.com/watson/developercloud/speech-to-text/api/v1, accessed Jan. 21. 2018.
[32] Google, "Cloud speech API basics," https://cloud.google.com/speech/docs/basics, accessed Jan. 21. 2018.
[33] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," J. Mach. Learn. Res., vol.9, pp.1871–1874, Aug. 2008.
[34] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to Japanese morphological analysis," Proc. EMNLP, Barcelona, Spain, pp.230–237, July 2004.
[35] "NAIST Japanese dictionary," https://osdn.net/projects/naist-jdic, accessed June 11. 2018.
[36] K. Shima, T. Homma, R. Ikeshita, H. Kokubo, Y. Obuchi, and J. She, "Interview-style-based method of collecting spontaneous speech corpus for car navigation systems," IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J101-D, no.2, pp.446–455, Feb. 2018.
[37] Clarion. Co., Ltd., "Clarion Intelligent VOICE," http://www.clarion.com/us/en/products-personal/service/IntelligentVoice/, accessed Oct. 6. 2017.
[38] A. Yano, T. Honda, A. Hayashi, H. Miyazawa, and H. Sawajiri, "Car information system for added value in connected cars," Hitachi Review, vol.95, no.11, pp.68–71, Nov. 2013. (in Japanese)
[39] C. Kim and R.M. Stern, "Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis," Proc. Interspeech, Brisbane, Australia, pp.2598–2601, Sept. 2008.
[40] S. Nakagawa and H. Takagi, "Statistical methods for comparing pattern recognition algorithms and comments on evaluating speech recognition performance," J. Acoust. Soc. Jpn. (Japanese Edition), vol.50, no.10, pp.849–854, Oct. 1994.
[41] K. Ono, R. Takeda, E. Nichols, M. Nakano, and K. Komatani, "Toward lexical acquisition during dialogues through implicit confirmation for closed-domain chatbots," Proc. Workshop on Chatbots and Conversational Agent Technologies (WOCHAT), Sept. 2016.
[42] H. He, A. Balakrishnan, M. Eric, and P. Liang, "Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings," arXiv:1704.07130v1, April 2017.
[43] R. Sarikaya, G.E. Hinton, and A. Deoras, "Application of deep belief networks for natural language understanding," IEEE/ACM Trans. Audio Speech Lang. Process., vol.22, no.4, pp.778–784, April 2014.
[44] B. Liu and I. Lane, "Joint online spoken language understanding and language modeling with recurrent neural networks," Proc. SIGDIAL, Los Angeles, USA, pp.22–30, Sept. 2017.
[45] Y.-Y. Wang, A. Acero, C. Chelba, B. Frey, and L. Wong, "Combination of statistical and rule-based approaches for spoken language understanding," Proc. ICSLP, Denver, USA, pp.609–612, Sept. 2002.
[46] T. Homma, A.S. Arantes, T. Gonzalez, and M. Togami, "Maximizing SLU performance with minimal training data using hybrid RNN plus rule-based approach," Proc. SIGDIAL, Melbourne, Australia, pp.366–370, July 2018.
[47] R.B. Miller, "Response time in man-computer conversational transactions," Proc. AFIPS '68 Fall Joint Computer Conference (Part I), pp.267–277, San Francisco, USA, Dec. 1968. DOI: 10.1145/1476589.1476628
[48] T. Homma, R. Zhang, T. Matsumoto, and H. Kokubo, "Speech recognition apparatus and speech recognition system," Japan Patent, JP 2018-81185 A, 2018.

Takeshi Homma received an A.E. degree from the Tsuruoka National College of Technology, Japan, B.E. and M.E. degrees from Hokkaido University, Japan, and a Ph.D. degree in advanced disciplinary studies from the University of Tokyo, Japan in 1998, 2000, 2002, and 2005, respectively. In 2005, he joined the Central Research Laboratory, Hitachi, Ltd. He worked at the Clarion Co., Ltd. from 2010 to 2013. From 2016 to 2018, he was a senior researcher in the R&D Division, Hitachi America, Ltd. He is a senior researcher in the Research & Development Group, Hitachi, Ltd. His research interests include speech-enabled human machine interfaces and spoken language understanding. He is a member of IEICE, ASJ, IPSJ, and IEEE.

Yasunari Obuchi received B.S. and M.S. degrees in physics from the University of Tokyo in 1988 and 1990, respectively. He received a Ph.D. in information science and technology from the University of Tokyo in 2006. From 1992 to 2015, he was with the Central Research Laboratory and Advanced Research Laboratory, Hitachi, Ltd. From 2002 to 2003, he was a visiting scholar at the Language Technologies Institute, Carnegie Mellon University. He was a visiting researcher in the Information Technology Research Organization, Waseda University from 2005 to 2010. He also worked at the Clarion Co., Ltd. from 2013 to 2015. Since 2015, he has been a professor at the School of Media Science, Tokyo University of Technology. He was a co-recipient of the Technology Development Award of the Acoustical Society of Japan in 2000. He is a member of IEEE, the Information Processing Society of Japan, Acoustical Society of Japan, and Society for Art and Science.

Kazuaki Shima received his B.E. degree from the Tokyo University of Technology, Japan in 2005. In the same year, he joined the Clarion Co., Ltd. as a car audio and navigation system engineer. From 2013 to 2015, he was engaged in the development of cloud-based voice recognition services for in-vehicle systems. He currently works in the Experimental Evaluation Dept. of the same company.

Rintaro Ikeshita received B.E. and M.E. degrees in mathematical informatics from the University of Tokyo in 2013 and 2015, respectively. He is a researcher in the Research & Development Group, Hitachi Ltd., Tokyo, Japan. Since joining Hitachi in 2015, he has been working on audio signal processing. He is a member of the ASJ.

Hiroaki Kokubo received B.E., M.E. and Ph.D. degrees from Sophia University, Tokyo, Japan in 1988, 1990, and 2003, respectively. In 1990, he joined the Central Research Laboratory, Hitachi Ltd. From 1995 to 1997, he was a researcher at ATR Interpreting Telecommunications Research Labs. From 2000 to 2004, he was a researcher at ATR Spoken Language Translation Research Labs. From 2011 to 2013, he was a deputy director at the Bureau of Science Technology and Innovation Policy, Cabinet Office, Government of Japan. He is a senior researcher in the Research & Development Group, Hitachi, Ltd. His current research includes speech interfaces.

Takuya Matsumoto received his B.E. and M.E. degrees in computer science from the Kyushu Institute of Technology, Japan in 1986 and 1988, respectively. In 1988, he joined the Computer Division, Hitachi, Ltd. From 2008 to 2017, he was with the Clarion Co., Ltd. He was engaged in the development of speech interfaces for in-vehicle systems from 2014 to 2017. Since 2017, he has been a senior manager in the Technology Development Department, Hitachi Automotive Systems, Ltd. He is a member of the Society of Automotive Engineers of Japan.