Retrospective Speech Recognition for Spoken Dialogue Systems: Exploiting Subsequent Utterances to Enhance ASR Performance

Ryu Takeda and Kazunori Komatani
SANKEN, University of Osaka
8-1 Mihogaoka, Ibaraki, Osaka, Japan
{rtakeda, komatani}@sanken.osaka-u.ac.jp

Abstract

Spoken dialogue systems would benefit from the ability of self-correction, namely, –revising earlier recognition results once later utterances are available, as humans often do in dialogue. However, conventional automatic speech recognition (ASR) frameworks mainly process user utterances sequentially and rely only on the preceding context. To address this limitation, we propose Retrospective Speech Recognition (RSR), which refines past recognition results by exploiting its subsequent utterances. We formulate and implement an RSR model for a dialogue system situation where system utterances can also be utilized. Each past user utterance is processed with an interpretable syllabogram representation, which integrates preceding and subsequent utterances within a shared domain between the signal and text levels. This intermediate representation also helps reduce orthographic inconsistencies. Experimental results using real Japanese dialogue speech showed that utilizing the subsequent utterances improved the character error rate by 0.10 points, which demonstrates the utility of RSR. We also investigated the impact of other factors, such as utilization of system utterances.

1 Introduction

Spoken dialogue systems would benefit from the ability of self-correction, namely, –revising earlier recognition results once later utterances are available, as humans often do in dialogue. If systems can correct past mis-recognitions after a sequence of conversations, dialogue breakdowns and incredulity from the user can be reduced. In other words, being able to detect and revise previous recognition errors, even retrospectively, is crucial not merely for ASR accuracy, but for maintaining coherent dialogue and reliable belief updates.

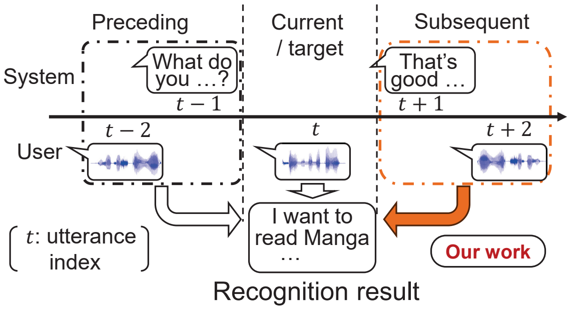

Figure 1: Proposed RSR for a dialogue system situation – utilizing subsequent context and system utterances

However, conventional automatic speech recognition (ASR) frameworks are not dialogue-oriented, i.e., they mainly process *user* utterances sequentially and rely only on the *preceding* context as shown on the left side of Fig. 1. While preceding utterances serve as a constraint in the recognition process, such constraints are often implemented by utilizing past recognition results as a prompt text for the language models (LMs) used in ASR (Radford et al., 2022). Context embedding has also been used instead of the raw text context (Masumura et al., 2021; Gong et al., 2024), and only preceding *system* utterances have also been exploited in the dialogue system area (Lee et al., 2024b). The context length is usually controlled by pre-defined parameters such as window size.

To address the limitation, we propose Retrospective Speech Recognition (RSR) , which refines past recognition results by exploiting the subsequent utterances (right side of Fig. 1). In this work, we formulate and implement an RSR model for a dialogue system situation where system utterances can also be utilized. Each past user utterance is processed with an interpretable syllabogram representation (pronunciation symbols), which integrates preceding and subsequent utterances within a shared domain between the signal and text (LM) levels. In our framework, a syllable sequence of each utterance signal is recognized by syllable ASR (S-ASR). Then, the recognized syllable sequences of preceding and subsequent utterances are con-

Proceedings of the 16th International Workshop on Spoken Dialogue Systems Technology, pages 182–192
February 26–March 1, 2026. ©2026 Association for Computational Linguistics

verted into the character sequence (ASR result) by a syllable-to-character translation (SCT) model.

Our approach, which leverages an intermediate representation, also mitigates orthographic inconsistencies. This capability is particularly important in dialogue systems where proper nouns often play a crucial role in understanding and maintaining context. For example, a spelling error in a system utterance text such as “ひげ団” for “髭男” does not matter as long as their pronunciations are the same.

Note that the proposed RSR scheme is suitable for spoken dialogue systems in terms of both processing latency and dialogue flow. The latency of RSR does not matter in dialogue systems because sequential ASR and RSR can run in parallel, and the RSR results can be selectively used only when considered necessary. As for the dialogue flow, subsequent utterances tend to assist RSR, since real dialogues typically stay on the same topic for several turns, and topic shifts are usually indicated by discourse markers. In addition, system utterances do not suffer from recognition errors, which will help with the correct recognition of dialogue context.

Our main contributions are as follows.

- We proposed a new formulation and a model for RSR under the dialogue system situation.
- We demonstrated the effectiveness of the RSR approach for real spoken dialogue data under several conditions: with and without system utterances and different context lengths.

2 Preliminaries

2.1 Assumption and Notations

We assume that an input signal is segmented into *utterance-wise* speech signals to cut down non-speech signal sections in advance. The segmentation is achieved on the basis of manual annotation or automatic estimation using voice activity detection (VAD) techniques. Here, the *pause* length is one of major criteria for the segmentation. Note that a user utterance sometimes continues under this condition.

The notations of variables related to the input and output of ASR are as follows. We denote the input speech features corresponding to the t-th utterance as $\mathbf{x}_t$ and its character sequence representation as $\mathbf{c}_t$, which is the output of ASR. Here, if the t-th utterance corresponds to the system, $\mathbf{c}_t$ is given because the system speech signal is usually generated from text by using text-to-speech technology. We need to estimate the hidden $\mathbf{c}_t$ corresponding to $\mathbf{x}_t$ if the t-th utterance corresponds to the user. Note that this problem setting is considered as *semi-supervised estimation* because the $\mathbf{c}_t$ of system utterances is partially “observed,” as shown in Fig. 2. Hereafter, the notation of $\mathbf{y}_{a:b}$ means the sequence vectors $[\mathbf{y}_i, ..., \mathbf{y}_j]$ from index i to j. $\mathbf{y}_t$ can be $\mathbf{x}_t$, $\mathbf{c}_t$, and so on.

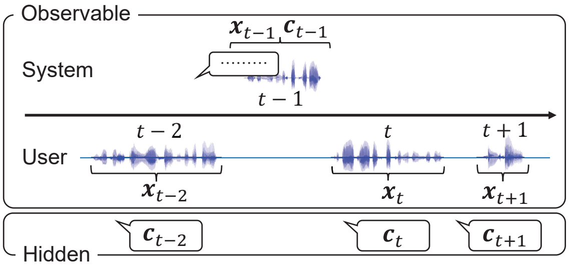

Figure 2: Utterance-wise notations in this paper

2.2 Sequential ASR over Utterances

The sequential ASR using preceding utterances can be generally formulated as the estimation of $\mathbf{c}_t$ given $\mathbf{x}_{1:t}$. Conceptually, we need to solve the following maximum posterior problem:

$$\hat{\mathbf{c}}_t = \mathrm{argmax}_{\mathbf{c}_t} p(\mathbf{c}_t|\mathbf{x}_{1:t}). \quad (1)$$

The modeling of the posterior probability is usually difficult due to the long sequence of $\mathbf{x}_t$.

The problem of Eq. (1) is often transformed and its acceptable solution is found by greedy search. The typical procedure consists of three steps: assume latent variables, decompose the joint probabilistic density function (PDF) of the variables and $\mathbf{c}_t$ into a directed graph, and apply directed greedy search for each vector. For example, $\mathbf{c}_{1:t-1}$ is often assumed as the latent variables, and if we decompose it into the factorial model, as shown in Fig. 3, we can estimate $\mathbf{c}_t$ *recursively* by using the previous estimations $\hat{\mathbf{c}}_{1:t-1}$, as

$$\hat{\mathbf{c}}_t = \mathrm{argmax}_{\mathbf{c}_t} p(\mathbf{c}_t|\mathbf{x}_t, \hat{\mathbf{c}}_{1:t-1}). \quad (2)$$

If $\mathbf{x}_i$ corresponds to system utterance, the estimation process is skipped because $\mathbf{c}_i$ is given.

Note that the actual implementation of the conditional PDF depends on the policy of the model designer. For example, a neural language model, decoder architecture, and embedding vector may be used to capture the language context $\hat{\mathbf{c}}_{1:t-1}$ that can be truncated to $\hat{\mathbf{c}}_{t-d:t-1}$ by a given context-window length d.

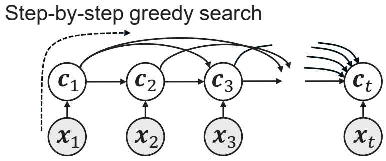

Figure 3: Typical graphical model and greedy search

2.3 Utterance-wise ASR via Syllabogram

We explain the foundation of utterance-wise ASR via syllabogram, that is, the sequence of pronunciation symbols. Note that we can use phoneme representation instead of syllabogram. This kind of ASR models estimates $\mathbf{c}_t$ from $\mathbf{x}_t$ via a syllabogram sequence $\mathbf{s}_t$ as an intermediate representation. The model assumes the following joint probability and its decomposition:

$$p(\mathbf{s}_t, \mathbf{c}_t|\mathbf{x}_t) = p(\mathbf{s}_t|\mathbf{x}_t)p(\mathbf{c}_t|\mathbf{s}_t), \tag{3}$$

where $p(\mathbf{s}_t|\mathbf{x}_t)$ and $p(\mathbf{c}_t|\mathbf{s}_t)$ represent a syllable-based ASR (S-ASR) model and a syllable-to-character translation (SCT) model, respectively. Each model can be implemented by a neural encoder-decoder architecture using Transformer. Since SCT is a seq2seq model for symbols, we can apply any neural models developed in the natural language processing area. Data augmentation based on an S-ASR error simulator is applied when training the SCT model to improve the robustness against S-ASR error (Takeda and Komatani, 2025).

As described in Section 2.2, greedy search can be applied to obtain an acceptable solution, as

$$\hat{\mathbf{s}}_t = \text{argmax}_{\mathbf{s}_t} p(\mathbf{s}_t|\mathbf{x}_t), \tag{4}$$
$$\hat{\mathbf{c}}_t = \text{argmax}_{\mathbf{c}_t} p(\mathbf{c}_t|\hat{\mathbf{s}}_t). \tag{5}$$

Note that the decoding cost function used in actual ASR/S-ASR models is further tuned in practice. For example, ESPnet (Watanabe et al., 2018) utilizes the weighted average score of CTC, attention, and a language model of shallow fusion.

3 Proposed Method: RSR

3.1 General Formulation

The general RSR problem is to estimate $\mathbf{c}_t$ from the first to the latest T-th utterances, $[\mathbf{x}_1, ..., \mathbf{x}_T]$. Here, t satisfies the relation of $1 \leq t \leq T$. As a formality, we need to solve the following problem:

$$\hat{\mathbf{c}}_t = \text{argmax}_{\mathbf{c}_t} p(\mathbf{c}_t|\mathbf{x}_{1:T}). \tag{6}$$

Since Eq. (6) is hard to solve and implement, we introduce a context window to truncate the utterance sequence. With context window parameters a and b for preceding and subsequent utterances, respectively, Eq. (6) becomes a local estimation problem as

$$\hat{\mathbf{c}}_t = \text{argmax}_{\mathbf{c}_t} p(\mathbf{c}_t|\mathbf{x}_{t_a:t_b}), \tag{7}$$

where $t_a = t - a$ and $t_b = t + b$ $(1 \leq t_a, t_b \leq T)$. Note that this formulation includes the sequential ASR setting: Eq. (7) is reduced to Eq. (1) if $a = t - 1$ and $b = 0$, except for the recursive (efficient) structure.

3.2 Model and Solution via Syllabogram

Our solution for Eq. (7) is based on the same models of ASR via syllabogram, i.e., the S-ASR and SCT models in Sec. 2.3. This approach provides a framework for symbolic integration of preceding and subsequent utterances, which makes it easy to utilize system utterance information.

We first assume latent variables for Eq. (7) based on the general procedure described in Sec. 2.2. The syllabogram sequences $\mathbf{s}_{t_a:t_b}$ in addition to the character sequences $\mathbf{c}_{t_a:t_b}$ except for $\mathbf{c}_t$ are assumed as latent variables. Therefore, we consider the joint PDF of $\mathbf{s}_{t_a:t_b}$, $\mathbf{c}_{t_a:t_b}$ and $\mathbf{x}_{t_a:t_b}$.

We decompose the joint PDF into the product of PDFs via syllabogram as

$$p(\mathbf{c}_{t_a:t_b}, \mathbf{s}_{t_a:t_b}|\mathbf{x}_{t_a:t_b}) = \\ p(\mathbf{c}_{t_a:t_b}|\mathbf{s}_{t_a:t_b})p(\mathbf{s}_{t_a:t_b}|\mathbf{x}_{t_a:t_b}). \tag{8}$$

The former conditional PDF exactly corresponds to the SCT model of which input and output sequence are the concatenated characters and syllabograms over utterances, respectively. The latter conditional probability is further decomposed into the utterance-wise PDF by assuming a feature-level conditional independence among utterances, as

$$p(\mathbf{s}_{t_a:t_b}|\mathbf{x}_{t_a:t_b}) = \textstyle\prod_{j=t_a}^{t_b} p(\mathbf{s}_j|\mathbf{x}_j), \tag{9}$$

where $p(\mathbf{s}_j|\mathbf{x}_j)$ represents the S-ASR model. Note that the $\mathbf{s}_i$ corresponding to a system utterance is obtained from the given character sequence $\mathbf{c}_i$ without this S-ASR process by using pronunciation dictionaries or grapheme-to-phoneme conversion (Bisani and Ney, 2008; Yolchuyeva et al., 2019).

There are two kinds of graphical model dependent on the SCT model, as shown in Fig. 4, which affects the inference direction of greedy search. The forward model estimates $\mathbf{c}_t$ *recursively* from the previously estimated $\hat{\mathbf{c}}_{t_a:t-1}$ and $\hat{\mathbf{s}}_{t_a:t_b}$, as

$$\hat{\mathbf{s}}_j = \text{argmax}_{\mathbf{s}_j} p(\mathbf{s}_j|\mathbf{x}_j) \ \ (j = t_a, ..., t_b) \tag{10}$$
$$\hat{\mathbf{c}}_t = \text{argmax}_{\mathbf{c}_t} p(\mathbf{c}_t|\hat{\mathbf{c}}_{t_a:t-1}, \hat{\mathbf{s}}_{t_a:t_b}) \tag{11}$$

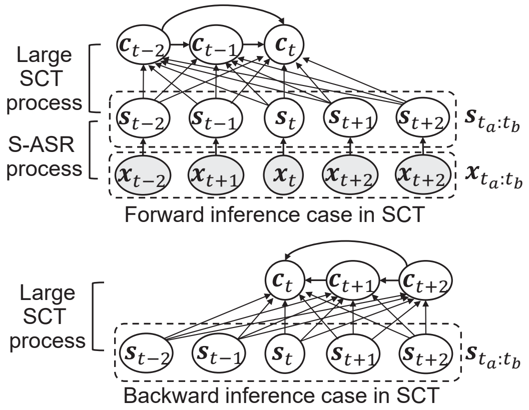
Large SCT process
S-ASR process
c_{t-2} c_{t-1} c_t
s_{t-2} s_{t-1} s_t s_{t+1} s_{t+2} $s_{t_a:t_b}$
x_{t-2} x_{t+1} x_t x_{t+2} x_{t+2} $x_{t_a:t_b}$
Forward inference case in SCT
Large SCT process
c_t c_{t+1} c_{t+2}
s_{t-2} s_{t-1} s_t s_{t+1} s_{t+2} $s_{t_a:t_b}$
Backward inference case in SCT

Figure 4: Two kinds of graphical model

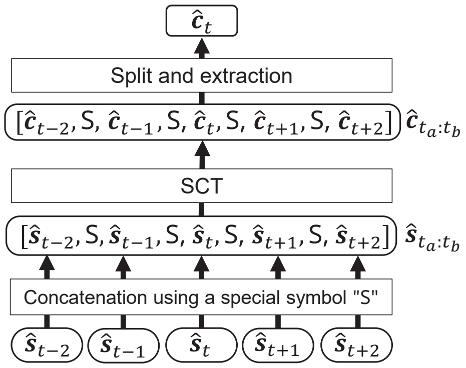
$\hat{c}_t$
Split and extraction
$[\hat{c}_{t-2}, S, \hat{c}_{t-1}, S, \hat{c}_t, S, \hat{c}_{t+1}, S, \hat{c}_{t+2}]$ $\hat{c}_{t_a:t_b}$
SCT
$[\hat{s}_{t-2}, S, \hat{s}_{t-1}, S, \hat{s}_t, S, \hat{s}_{t+1}, S, \hat{s}_{t+2}]$ $\hat{s}_{t_a:t_b}$
Concatenation using a special symbol "S"
$\hat{s}_{t-2}$ $\hat{s}_{t-1}$ $\hat{s}_t$ $\hat{s}_{t+1}$ $\hat{s}_{t+2}$

Figure 5: Extraction of a recognition result $\hat{\mathbf{c}}_t$

In contrast, the backward model estimates $\mathbf{c}_t$ *recursively* from the previously estimated $\hat{\mathbf{c}}_{t+1:t_b}$ and $\hat{\mathbf{s}}_{t_a:t_b}$, as

$$\hat{\mathbf{c}}_t = \text{argmax}_{\mathbf{c}_t} p(\mathbf{c}_t | \hat{\mathbf{c}}_{t+1:t_b}, \hat{\mathbf{s}}_{t_a:t_b}). \tag{12}$$

We investigate which model is preferable for RSR through experiments.

3.3 Implementation of Large SCT Process

There are two issues in the actual implementation of a large SCT process stemming from its character-by-character estimation: utterance boundary and backward inference. The first issue is that we cannot distinguish $\hat{\mathbf{c}}_t$ from $\hat{\mathbf{c}}_{t_a:t_b}$ because the character sequence estimated by the SCT model does not include landmarks for utterance boundaries. The second issue is that the backward inference is also required in the character-by-character estimation within each utterance $\mathbf{c}_t$.

The utterance boundary issue is solved by introducing a special symbol S into $\hat{\mathbf{s}}_{t_a,t_b}$ that represents a boundary of each utterance (Fig. 5). Since the SCT model automatically inserts the utterance-boundary symbols in $\hat{\mathbf{c}}_{t_a,t_b}$ according to those in $\hat{\mathbf{s}}_{t_a,t_b}$, we can separate the output character sequence into each utterance-wise result. For example, if the input sequence of SCT is $[\hat{\mathbf{s}}_{t-1}, \mathsf{S}, \hat{\mathbf{s}}_t, \mathsf{S}, \hat{\mathbf{s}}_{t+1}]$, the corresponding output sequence will become $[\hat{\mathbf{c}}_{t-1}, \mathsf{S}, \hat{\mathbf{c}}_t, \mathsf{S}, \hat{\mathbf{c}}_{t+1}]$. The result of the target utterance $\hat{\mathbf{c}}_t$ can be extracted by a simple string manipulation: 1) index the special symbols in $\hat{\mathbf{s}}_{t_a,t_b}$, 2) split $\hat{\mathbf{c}}_{t_a,t_b}$ into each segment, and 3) extract the $\hat{\mathbf{c}}_t$ according to the indices of $\hat{\mathbf{s}}_t$.

Table 1: Training set for each model

Model	S-ASR
Data	Paired data (audio & text) (Japanese 10 corpora)
Size	Over 12,500 hours (audio)

Model	SCT
Data	Text in paired data + unpaired text (10 corpora + BCCWJ, Wiki40b-ja)
Size	Over 400 million characters

The backward inference issue is solved by introducing a reversed order to the input and output sequences of the forward SCT model. The solution is simply to use an order-reversed input and output in both the training and inference phases. With a symbol-order-reverse operator $\mathsf{reverse}(\cdot)$, the input and output sequences become $\mathbf{c}_k^r = \mathsf{revserse}(\mathbf{c}_k)$ and $\mathbf{s}_k^r = \mathsf{revserse}(\mathbf{s}_k)$, respectively. The estimation of $\hat{\mathbf{c}}_t$ is obtained by $\mathsf{reverse}(\hat{\mathbf{c}}_t^r)$, where $\hat{\mathbf{c}}_t^r$ is the output of the reversed SCT model.

4 Experiment

4.1 Data set

Training data for S-ASR: The training speech data over 12,500 hours were generated by data augmentation of a seed data set (Table 1). Ten *public* Japanese speech corpora with transcriptions were utilized: CSJ (Maekawa, 2003), S-JNAS, TWM, JEIDA-JCSD, ETL-WD, RIKEN-DLG[1], APP, APPDIC[2], SLC-3[3], and JVS (Takamichi et al., 2019). Simulated speech-rate perturbation, reverberation, and background noise were applied to augment the seed data set. Impulse responses measured at 540 positions in a real room (RT_{20} 640 ms) were utilized to simulate various reverberations. The background noise data consisted of MUSAN (Sny-

[1] https://research.nii.ac.jp/src/list.html
[2] https://www.atr-p.com/products/sdb.html
[3] https://alaginrc.nict.go.jp/slc-outline.html

Table 2: Test set

Corpus	Hazumi
Version	1712, 1902, 1911, 2105
Topic	User's hobby and experiences: travel, manga, music, etc...
Operation	Wizard of Oz

Table 3: Statistics of Hazumi

No. of user utters	16,200
No. of characters in whole user utters	**228,242**
No. of user utters per exchange (avg.)	1.6
No. of system utters per exchange (avg.)	1.0
No. of utters per exchange (avg.)	2.6

der et al., 2015), WHAM! (train set) (Wichern et al., 2019), the ProSoundEffects corpus[4], and random noises. The signal-to-noise ratio (SNR) was randomly selected from $-10, -5, 0, 5, 10$, and 20 dB.

Training text for SCT: The training text consisted of transcriptions from the paired data, BCCWJ text (Kikuo et al., 2014), unpaired Wiki-40B (ja) text (Guo et al., 2020), and Wikipedia title data (Table 1). Japanese morphological analyzer Mecab (Kudo et al., 2004) with the NEologd (Sato et al., 2017) and UniDic (Ogiso et al., 2012) dictionaries were used to obtain syllabogram representation (Katakana) of text. The spellings and representation of numbers were standardized in accordance with the transcription rules of CSJ. For example, some alphabetical words were represented by Katakana for the LM of C-ASR.

Test set: The test sets comprised four Hazumi$\{1712, 1902, 1911, 2105\}$ (Komatani et al., 2023) corpora (11.5 hours in total) featuring real recordings of human-machine interactions in a spoken dialogue system using Wizard of Oz method. The system provides topics, and users talk about a TV show, manga, music, and other subjects on the basis of *their own experiences*. Note that a human operator listened to and responded to the user. Therefore, the word uttered by the user could sometimes be included in the subsequent system utterance text with correct spellings. There were $16,200$ user utterances in total, and the number of user utterances per exchange was 1.6. Here, an exchange is a pair of system utterances and user utterances in a turn. These settings and corpus statistics are summarized in Tables 2 and 3.

4.2 Model configuration

S-ASR: Almost all configurations (e.g., the NN architectures) were the same as the ESPnet CSJ recipe with Transformer ASR and LM (Watanabe et al., 2018). The number of parameters was 97M for ASR and 50M for LM. The vocabulary size was about 170, including *silent* and special symbols. The number of training epochs was set to 30 with a default scheduler, and the models were then tuned further with fixed learning rates of 2.0×10^{-5} and/or 2.0×10^{-6}. The final parameters were obtained by averaging the parameters over several epochs from each learning rate.

SCT: T5 for conditional generation (Raffel et al., 2020) was used as the SCT model. We trained this model from scratch with parallel text, the default T5's loss function, and the AdamW optimizer (Loshchilov and Hutter, 2018). The number of layers was 12, and the number of parameters was 110M. The vocabulary size was approximately 11,100 Japanese characters, defined by JIS X 0213 (Japanese Industrial Standard for coded character sets). The number of training epochs was set to 10 with fixed learning rates of first 10^{-4} and then 10^{-6}. The other configurations remained default.

Other settings: The beam size during decoding was set to 40 in S-ASR and 15 in SCT. The experiments were conducted on Nvidia RTX A6000 GPUs. The statistical significance of the character error rate (CER) differences between two methods was assessed using the probability of improvement (POI) in % via the bootstrap method (Bisani and Ney, 2004) in the Kaldi toolkit (Povey et al., 2011) under 95% confidence interval settings.

4.3 Results

The character error rate (CER) of RSR was compared with those of utterance-wise and sequential ASR under several conditions: with and without utilizing system utterances and with different window lengths (a, b) and forward/backward inferences. Here, we present the summarized main results first and then follow the detailed results, such as the impact of window length. In the following tables and figures, *Both* denotes recognition with both system and user utterances, and *User* denotes that with only user utterances. In the case of *User* condition, utterance index t counts up only user utterances. *Fwd.* and *Bwd.* mean the forward and backward inference models of SCT in Fig.4, respectively.

[4] http://www.prosoundeffects.com

Table 4: Main results: CER (↓) in %. *Pre.* and *Sub.* mean preceding and subsequent utterances used for recognition, respectively.

	Baseline			Proposed
	Utt.-wise	Sequential		RSR
	–	Pre.	Sub.	Pre.+Sub.
(a, b)	$(0, 0)$	$(9, 0)$	$(0, 9)$	$(9, 9)$
Both+Fwd.	11.57	11.12	–	*11.03*
Both+Bwd.	11.68	–	11.18	11.07
User+Fwd.	11.57	11.30	–	11.13
User+Bwd.	11.68	–	11.23	11.12

Main Results (Table 4)

The CERs of RSR were better than those of the sequential model. The CER for RSR with *Both+Fwd.* was 0.09 points higher than that of the sequential setting with the PoI of 100%. As for the sequential models, the preceding utterances improved the CERs compared to the result using subsequent utterances.

Comparing the *Both* and *User* conditions, the CER of the *Both+Fwd.* condition was improved by 0.10 points over that of the *User+Fwd.* condition. A key finding is that the CER of RSR under the *User+Fwd.* condition was improved by 0.17 compared to that of the sequential model, even if we use only user utterances that usually include syllable recognition errors. This demonstrates the pure impact of subsequent utterances without a semi-supervised estimation situation.

The forward inference of RSR performed better than the backward inference by 0.04 points under *Both* condition. In contrast, the backward the inference of both sequential setting and RSR was effective under the *User* condition. This may be caused by the syllable recognition errors for user utterances when there are almost no syllable recognition errors for system utterances.

Detailed Results of Sequential ASR (Fig. 6)

We found that the CER under the sequential setting improved as the window length became longer, but the improvement was limited as shown in Fig. 6. Here, the CER with window length parameter 0 corresponds to that of utterance-wise ASR. For example, the utilization of nine preceding utterances improved CER by 0.45 points compared to the utterance-wise ASR in the case of *Both+Fwd.*.

The performance improvement of the backward inference (*Bwd.*) under sequential condition demonstrates the effectiveness of utilizing subsequent utterances. In terms of the efficiency of CER improvement over window length, the forward inference was better than the backward one under the *Both* condition, while the backward inference was superior under the *User* condition. This tendency is similar to what was observed in the main results.

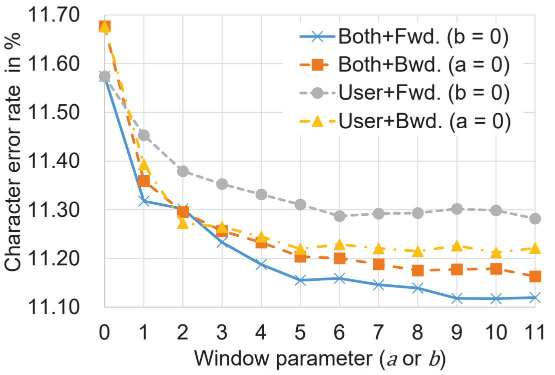

Figure 6: CER (↓) vs. window length under sequential condition: $a = 0$ or $b = 0$.

Table 5: CER gain (↑) vs. window length under RSR (Both + Fwd.) condition. Gain is the difference between CERs of RSR and utterance-wise ASR (11.57%).

$a \backslash b$	0	1	2	3	4	5	6	7	8	9
1	.26	.26	.37	.42	.45	.47	.48	.49	.49	.48
2	.27	.32	.41	.44	.49	.49	.50	.51	.52	.52
3	.34	.38	.45	.47	.49	.49	.50	.52	.52	.53
4	.39	.39	.46	.47	.49	.50	.52	.52	.53	.54
5	.42	.41	.48	.49	.50	.51	.52	.54	.53	.54
6	.41	.43	.48	.48	.50	.51	.51	.53	.53	.54
7	.43	.45	.49	.50	.51	.52	.52	.53	.52	.53
8	.43	.46	.48	.49	.51	.51	.51	.53	.52	.53
9	.46	.46	.48	.50	.51	.51	.51	.52	.53	.54

Note that the window length of nine under the *Both* condition corresponds to 3.4 exchanges by rough estimation because one exchange includes 2.6 utterances on average. This indicates that the context of almost three exchanges is a sufficient length for RSR.

Detailed Results of RSR (Table 5)

Table 5 shows the detailed CER improvements of RSR with various window parameters under the *Both+Fwd.* condition. As we can see, the RSR performance usually improved as both window parameters a and b become larger. Therefore, if there is no computational restriction, both parameters a and b should be set to large values.

The CERs on the diagonal were better than the CERs on the edges, which also demonstrates the contribution of subsequent utterances. This is because they show the RSR performances under a

Ground truth	ツムツムとかですかね	It's TsumTsum, maybe.
Utterance-wise: $(a, b) = (0, 0)$	積む積むとかですかね	It's stack, stack, maybe.
Sequential: $(a, b) = (9, 0)$	積む積むとかですかね	It's stack, stack, maybe.
RSR: $(a, b) = (9, 9)$	ツムツムとかですかね	It's TsumTsum, maybe.

Figure 7: Example of recognition results. The ground truth and the results of the sequential setting and the RSR in Japanese and English are shown. Here, "ツムツム" ("TsumTsum") is the name of video game. "積む" means "stack" or "pile up" in Japanese.

Index	ID	Input of large SCT ($\hat{\mathbf{s}}_t$)	Text ($\mathbf{c}_t$)
$t-5$	S30	デワヤッタコトノアルゲームノナカデモーイチドヤリタイモノワアリマスカ	では、やったことのあるゲームの中で、もう一度やりたいものはありますか？
$t-4$	U39	ゲームノナカデベツノ (ゲームノナカデエソノ)	
$t-3$	U40	ケータイノゲームノナカデデスカ	
$t-2$	S31	ハイソーデス	はい、そうです。
$t-1$	U41	ン (ウーン)	
t	**U42**	**ツムツムトカデスカネ**	
$t+1$	S32	ナルホドソーナンデスネ	なるほど、そうなんですね。
$t+2$	U43	エ (ウン)	
$t+3$	S33	コドモカラオトナマデサイキンワスイッチトイウゲームキデゲームオスルノガハヤッテイルミタイデスネ	子供から大人まで最近はスイッチというゲーム機でゲームをするのが流行っているみたいですね。
$t+4$	U44	ソレジャホシガッテマス (スイッチホシガッテマス)	
$t+5$	S34	ダイヒョーテキナモノニスプラトゥーンヤカービイマリオノゲームナドアリマスガアソンダコトワアリマスカ	代表的なものにスプラトゥーンやカービィ、マリオのゲームなどありますが、遊んだことはありますか？

Figure 8: The preceding and subsequent sentences of the user's utterance in Fig. 7. Characters in brackets represent the ground truth of the syllabogram without S-ASR errors. The text $\mathbf{c}_t$ of only system utterances is available. Here, "スイッチ" means "Nintendo Switch", and "マリオ" means "Mario," a character in a Nintendo game.

constant window length, i.e., $a + b = \text{const}$. It is therefore better to use both preceding and subsequent utterances for performance improvement under the constraint of the fixed length $(a + b)$.

Example of RSR results (Figs. 7 and 8) We show an example of the recognition results by utterance-wise ASR, sequential ASR and RSR. The actual recognition results and the preceding and subsequent context are shown in different figures. Note that the example here shows that RSR succeeded in utilizing subsequent context while there are other kinds of correctly recognized patterns.

Fig. 7 shows that only RSR estimated the correct characters of the user's utterance while utterance-wise and sequential ASR failed. Here, the pronunciation of "ツムツム" and "積む積む" is the same, but their meanings are different. Since "積む" is a general verb or noun in Japanese and it appears in text resources more frequently than "ツムツム", it is no surprise that utterance-wise ASR failed.

The preceding and subsequent utterances of the user utterance in Fig. 7 are shown in Fig. 8. As we can see, "ツムツム" was recognized correctly when the utterance of S34 ($t + 5$) was utilized in SCT. At a glance, it seems there are no words that co-occur with "ツムツム". The game "ツムツム (TsumTsum)" was also launched as a Nintendo Switch software, so the words "スイッチ (Switch)" and "マリオ (Mario)" might enhance the co-occurrence of "ツムツム" under this context. Of course, a more accurate SCT model may estimate "ツムツム" by utilizing only preceding utterances that include the word "ゲーム (game)".

5 Related Work

There are several related works in the spoken dialogue system and ASR areas. Note that LM- or text-level methods can also be incorporated into the STC model in our approach.

Previous studies in spoken dialogue system (Lee et al., 2024b) are usually based on *sequential* setting, and their focus is ASR error robustness. The language context, i.e., the preceding system and user utterances, is encoded into a vector and exploited in the speech decoder. The context robustness was improved by introducing noise representation learning. In our study, the robustness of SCT against ASR error was also improved by data augmentation.

The neural architecture for large/long context ASR has also typically been developed under sequential, real-time, or no-system-utterances settings (Masumura et al., 2021; Gong et al., 2023, 2024). Therefore, the processing speed and incremental processing based on RNN-T are weighted in the ASR area. There is usually no assumption of semi-supervised (exploiting system utterances) and RSR settings. Although cross-context (preceding and subsequent utterances) is exploited in the LM score computation for CTC-based ASR (Flynn and Ragni, 2023), system utterances were not considered. Note that this LM score framework can also be applied in our SCT process by LM fusion. Other approaches have utilized the past speech signals to obtain better encoded features by applying sliding window processing (Hori et al., 2020).

Error correction or LM-based rescoring methods in the ASR area (Lee et al., 2024a; Sun et al., 2020) are partially related to our settings because the models usually utilize whole recognition results but do not usually assume system utterances. For example, the cross-context from the English ASR results of audio recoding were exploited in T5 translation model to correct ASR errors of an utterance (Lee et al., 2024a). Since error correction is performed completely as a post-processing operation after ASR, we can incorporate error correction methods into our RSR results.

6 Limitations

While we have demonstrated the effectiveness of RSR through experiments in this work, there are limitations to its application in real spoken dialogue systems. These limitations are divided into two main aspects: the computational cost of RSR itself and the design of a dialogue system assuming RSR.

In the future, the specialization of model architecture and decoding algorithm of RSR will reduce the fundamental computational cost. This is because the current implementation of the RSR model is just based on the general framework of neural translation (model and decoding algorithm). The semi-supervised situation will lead to a more efficient architecture and decoding algorithm from this general framework. In addition, an SCT model that can utilize the N-best results of S-ASR is also desirable in terms of S-ASR error robustness. Knowledge distillation techniques (Gou et al., 2021) will also contribute to making the models lighter.

Moreover, the estimation of utterances that require RSR will reduce the computational load in an actual system. If we can detect the potential misrecognized utterances, the number of times RSR needs to be applied will be reduced. The confidence score of utterance-wise ASR can be a criterion for such detection. From the view-point of dialogue systems, the potential importance of the utterance in the current context will also help to make RSR more meaningful.

The dialogue management for RSR should also be designed to more fully utilize the RSR function (self-correction ability). When the RSR output differs from the sequential ASR result, it indicates that the system has misunderstood a user utterance. In such a case, the system may need to change the flow of dialogue or update the system's belief state or dialogue context. We should also consider the timing or scheduling of RSR and its related modifications.

7 Conclusion

In this study, we examined how subsequent utterances affect the ASR performance in terms of the re-recognition function by investigating the impact of the subsequent utterances on the basis of the syllable-based ASR (S-ASR) and syllable-to-character translation (SCT) processes. Experimental results utilizing dialogue speech demonstrated the positive contribution of the subsequent utterances to ASR performance.

Future work will involve a detailed modeling and practical implementation of the re-recognition function, including an improvement of the formulation and model, the development of a faster inference algorithm, and an efficient implementation in the real spoken dialogue system. We also need to investigate the impact of RSR on downstream dialogue system behavior.

Acknowledgments

This work was partly supported by JSPS KAKENHI Grant Numbers JP23K28147 and JP22H00536, and JST Moonshot R&D Grant Number JPMJM2011, Japan.

References

Maximilian Bisani and Hermann Ney. 2004. Bootstrap estimates for confidence intervals in ASR performance evaluation. In *Proc. of ICASSP*, pages 409–409.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Commun.*, 50(5):434–451.

Robert Flynn and Anton Ragni. 2023. Leveraging cross-utterance context for asr decoding. In *Proc. of Interspeech*, pages 1359–1363.

Xun Gong, Yu Wu, Jinyu Li, Shujie Liu, Rui Zhao, Xie Chen, and Yanmin Qian. 2023. LongFNT: Long-form speech recognition with factorized neural transducer. In *Proc. of ICASSP*, pages 1–5.

Xun Gong, Yu Wu, Jinyu Li, Shujie Liu, Rui Zhao, Xie Chen, and Yanmin Qian. 2024. Advanced long-content speech recognition with factorized neural transducer. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 32:1803–1815.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *Int. J. Comput. Vision*, 129(6):1789–1819.

Mandy Guo, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. 2020. Wiki-40B: Multilingual language model dataset. In *Proc. of LREC*, pages 2431–2439.

Yukiya Hono, Koh Mitsuda, et al. Rinna/Nue-ASR.

Yukiya Hono, Koh Mitsuda, et al. 2024. Integrating pre-trained speech and language models for end-to-end speech recognition. In *Proc. of Findings of ACL*, pages 13289–13305.

Takaaki Hori, Niko Moritz, Chiori Hori, and Jonathan Le Roux. 2020. Transformer-based long-context end-to-end speech recognition. In *Proc. of Interspeech*, pages 5011–5015.

Maekawa Kikuo, Makoto Yamazaki, et al. 2014. Balanced corpus of contemporary written Japanese. *Language Resources and Evaluation*, (48):345–371.

Kazunori Komatani, Ryu Takeda, and Shogo Okada. 2023. Analyzing differences in subjective annotations by participants and third-party annotators in multimodal dialogue corpus. In *Proc. of SIGDIAL*, pages 104–113.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proc. of EMNLP*, pages 230–237.

Seongmin Lee, Kohki Tamura, Tomoaki Nakamura, and Naoki Yoshinaga. 2024a. Can noisy cross-utterance contexts help speech-recognition error correction? In *Proc. of IWSDS*.

Wonjun Lee, San Kim, and Gary Geunbae Lee. 2024b. Enhancing dialogue speech recognition with robust contextual awareness via noise representation learning. In *Proc. of SIGDIAL*, pages 333–343, Kyoto, Japan. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *Proc. of ICLR*.

Kikuo Maekawa. 2003. Corpus of spontaneous Japanese: Its design and evaluation. In *Proc. of ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*.

Ryo Masumura, Naoki Makishima, Mana Ihori, Akihiko Takashima, Tomohiro Tanaka, and Shota Orihashi. 2021. Hierarchical transformer-based large-context end-to-end asr with large-context knowledge distillation. In *Proc. of ICASSP*, pages 5879–5883.

Toshinobu Ogiso, Mamoru Komachi, Yasuharu Den, and Yuji Matsumoto. 2012. UniDic for early middle Japanese: a dictionary for morphological analysis of classical Japanese. In *Proc. of LREC*, pages 911–915.

Daniel Povey, Arnab Ghoshal, et al. 2011. The kaldi speech recognition toolkit. In *Proc. of SLT*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *Preprint*, arXiv:2212.04356.

Alec Radford, Jong Wook Kim, et al. 2023. Robust speech recognition via large-scale weak supervision. In *Proc. of ICML*.

Colin Raffel, Noam Shazeer, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(1).

Toshinori Sato, Taiichi Hashimoto, and Manabu Okumura. 2017. Implementation of a word segmentation dictionary called mecab-ipadic-NEologd and study on how to use it effectively for information retrieval (in Japanese). In *Proc. of Annual Meeting of the Association for NLP*, pages NLP2017–B6–1.

David Snyder, Guoguo Chen, and Daniel Povey. 2015. MUSAN: A Music, Speech, and Noise Corpus. ArXiv:1510.08484v1.

G. Sun, C. Zhang, and P. C. Woodland. 2020. Cross-utterance language models with acoustic error sampling. *Preprint*, arXiv:2009.01008.

Shinnosuke Takamichi, Kentaro Mitsui, et al. 2019. JVS corpus: free Japanese multi-speaker voice corpus. *Preprint*, arXiv:1908.06248.

Ryu Takeda and Kazunori Komatani. 2025. Reducing orthographic dependency on paired data by probabilistic integration via syllabogram for japanese dialogue speech recognition. In *Proc. of APSIPA ASC*, pages 549–554.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, et al. 2018. ESPnet: End-to-end speech processing toolkit. In *Proc. of Interspeech*, pages 2207–2211.

Gordon Wichern et al. 2019. WHAM!: Extending speech separation to noisy environments. In *Proc. of Interspeech*, pages 1368–1372.

Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. 2019. Transformer based grapheme-to-phoneme conversion. In *Proc. of Interspeech*, pages 2095–2099.

Table 6: CER (↓) in % for Hazumi corpus

Condition		CER
Open ASR model	Reazon v2	13.06
	Whisper large-v3	16.41
	Nue	22.29
	ESPnet (CSJ recipe)	22.46
Ours	Character-ASR (Utt.-wise)	12.20
	S-ASR + SCT	
	Utt.-wise: $(a, b) = (0, 0)$	11.57
	Sequential: $(a, b) = (9, 0)$	11.12
	RSR: $(a, b) = (9, 9)$	11.03
	No. of characters	228,242

A Comparison with Open ASR Models

We demonstrate here that the CERs of our utterance-wise setting offer a reasonably better performance as baselines through comparison with other open ASR models. The CERs by Reazonspeech ESPnet v2[5], Whisper large v3 (Radford et al., 2023), and Rinna Nue (Hono et al., 2024; Hono et al.) for Hazumi were compared under the default settings and utterance-wise situation (Takeda and Komatani, 2025). In addition, we provided the performance of ESPnet-based character ASR (C-ASR) trained by our training set with the same configuration of S-ASR.

As shown in Table 6, our models including C-ASR outperformed the open ASR models even under the utterance-wise setting. This is mainly because 1) speech in the Hazumi set is a little noisy and reverberated, and 2) many proper nous are included in Hazumi. Moreover, deletion errors of fillers and some content words often occurred with these open ASR models. Although Whisper can exploit preceding utterances, we can not expect a dramatic improvement even if we exploit the preceding language context.

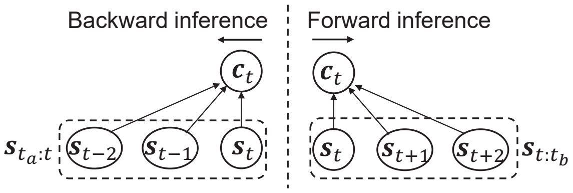

Figure 9: Simplified models in SCT process under sequential condition

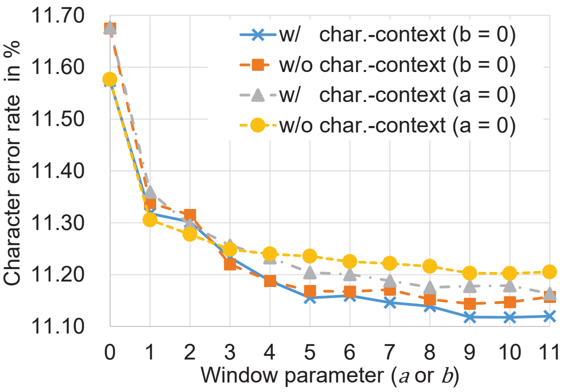

Figure 10: CER (↓) vs. window length under sequential condition: $a = 0$ or $b = 0$.

B Impact of Character-level Context in SCT Process

Our question here is whether it is necessary to estimate the latent variables $\mathbf{c}_{t'} (t' \neq t)$ for $\mathbf{c}_t$. The answer can be obtained by comparing the performance of the models shown in Fig. 4 and by Fig. 9 under sequential ASR condition. The estimation using the models in Fig. 9 corresponds to the following SCT process:

$$\hat{\mathbf{c}}_t = \mathrm{argmax}_{\mathbf{c}_t} p(\mathbf{c}_t | \mathbf{s}_{t_a:t}), \tag{13}$$

$$\hat{\mathbf{c}}_t = \mathrm{argmax}_{\mathbf{c}_t} p(\mathbf{c}_t | \mathbf{s}_{t:t_b}). \tag{14}$$

While the former conditional PDF can be implemented by a reversed SCT model, the latter can be implemented by a standard SCT model. Neighboring character-level contexts are not exploited in these models.

We found that utilizing character-level context improved the CERs under the same conditions ($a = 0$ or $b = 0$), as shown in Fig. 10. Both system and user utterances were used in this evaluation. Here, *w/ char.-context* and *w/o char.-context* correspond to the models shown in Fig. 4 and Fig. 9,

[5] https://huggingface.co/reazon-research/reazonspeech-espnet-v2

Ground truth	うん コブクロ とか 行きました	Yes, I've been to Kobukuro's (concert)
Utterance-wise: $(a, b) = (0, 0)$	うん コブクロ とか 行きました	Yes, I've been to Kobukuro's
Sequential: $(a, b) = (9, 0)$	うん こ袋 とか 行きました	Yes, I've been to Ko-bag
RSR: $(a, b) = (9, 9)$	うん コブクロ とか 行きました	Yes, I've been to Kobukuro's

Figure 11: Example of recognition results. Top table shows the ground truth and the results of the sequential setting and the RSR in Japanese and English. Here, "コブクロ" ("Kobukuro") is the name of a Japanese band. "こ袋" ("Ko-bag") is a meaningless word.

Index	ID	Input of large SCT ($\hat{\mathbf{s}}_t$)	Text ($\mathbf{c}_t$)
$t-5$	S57	コンサートトカニワイキマスカ koNsa:totokaniwaikimasuka	コンサートとかには行きますか？ Do you go to concerts?
$t-4$	U78	ニジマデイキマス (イキマスイキマス) nijimadeikimasu (ikimasuikimasu)	
$t-3$	S58	タトエバ tatoeba	たとえば For example?
$t-2$	U79	タトエバ tatoeba	
$t-1$	U80	エーチョッキンデ (ウーンチョッキンデ) e:choqkinde (u:Nchoqkinde)	
t	U81	ウンコブクロトカイキマシタ (イッタノウンコブクロトカイキマシタ) uNkobukurotokaikimashita (iqtanouNkobukurotokaikimashita)	
$t+1$	S59	ナマデエンソーオキクトハクリョクガアッテカンドースルミタイデスネ namadeeNso:okikutohakuryokugaaqte kando:surumitaidesune	なまで演奏をきくと、迫力があって、感動するみたいですね！ Hearing the performance live is so powerful and moving!
$t+2$	U82	ソーデスネゼンゼンチガイマスネ so:desunezeNzeNchigaimasune	
$t+3$	U83	ミンナファンナノデノリモイーデス miNnafaNnanodenorimoi:desu	
$t+4$	S60	タノシソーデスネ tanoshiso:desune	楽しそうですね！ That looks fun!
$t+5$	U84	ウンタノシーデスヨ uNtanoshi:desuyo	
$t+6$	S61	ワタシモジッサイニエンソーオキイテミタイモノデス watashimojiqsainieNso:oki:temitaimonodesu	わたしも実際に、演奏をきいてみたいものです I'd like to hear the performance

Figure 12: The preceding and subsequent sentences from Fig. 11 in Japanese and English. Characters in brackets represent the ground truth of the syllabogram without S-ASR errors. The text $\mathbf{c}_t$ of only system utterances is available.

respectively. The CERs of *w/ char.-context* outperformed those of *w/o char.-context* by 0.02 points when the window parameter was set to 9. The difference between the two increases as the window length increases, which demonstrates that character-level contextual information enhances RSR performance with longer contexts.

C Example of RSR

Figs. 11 and 12 show the recognition results and context utterances of a different utterance from Fig. 11. In this example, the preceding utterances degraded the recognition performance.

The recognition by sequential ASR failed and output "こ袋" by utilizing preceding utterances, while it was recognized correctly by utterance-wise ASR as "コブクロ". The subsequent utterances were utilized to recover from the failure. Since "コブクロ (Kobukuro)" refers to a Japanese band, the words "演奏 (performance)" and "ファン (fan)" might re-enhance the relationship.