Reducing Orthographic Dependency on Paired Data by Probabilistic Integration via Syllabogram for Japanese Dialogue Speech Recognition

Ryu Takeda* and Kazunori Komatani*
*SANKEN, The University of Osaka, Japan
E-mail: {rtakeda, komatani}@sanken.osaka-u.ac.jp Tel: +81-6-6879-8416

***Abstract*—Character end-to-end ASR (C-ASR) models fail to recognize the spelling of words that are absent from paired data (audio and text). This orthographic dependency on the paired data can be resolved by using a cascade of syllable ASR (S-ASR) and syllable-to-character translation (SCT) models once the pronunciations of the text are given as syllabograms in model training. However, this method's performance suffers from errors that propagate from each model. We propose a probabilistic integration of the two models and robust training of the SCT model. The former provides a global score for better estimations based on S-ASR and SCT models during the block-wise beam search. The latter introduces a pronunciation error generator using a mask token to increase robustness against S-ASR errors. Experimental results using real Japanese dialogue corpora showed that our model outperformed the naive cascading with no probability, C-ASR, and other open models. We also showed that the integration of recognition results from C-ASR and S-ASR with SCT outperformed each isolated model.**

I. Introduction

A. Background and Motivation

Our purpose is to develop a *vocabulary-portable* automatic speech recognition (ASR) for spoken dialogue systems. Such an ASR should recognize not only well-known words but also specialized words, such as technical terms, local in-group words and new words, in order to be portable across various scenarios and domains. Several open neural ASR models have shown high performance in general, but they often ignore fillers and hesitations – key cues in dialogue that reflect user traits, internal states, and turn-taking behavior [1]–[4]. It is partly due to the use of rough labels (i.e., inaccurate transcriptions) during training.

Character end-to-end ASR (C-ASR) models often fail to recognize the spellings of words that are absent from paired data (audio and text), which reduces portability. Although they can be augmented with a language model trained on a large amount of unpaired text, recognition still often fails due to the direct and strong association of acoustic features with spelling characters learned from the paired data. This issue is caused by the *orthographic dependency* on the paired data: For example, a word "YouTube/ユーチューブ" that appears only in the unpaired text may be misrecognized as frequent characters in the paired data "優中部" because their pronunciations are the same (Fig. 1 left). Such dependency becomes serious in several languages, like Japanese, whose writing system includes

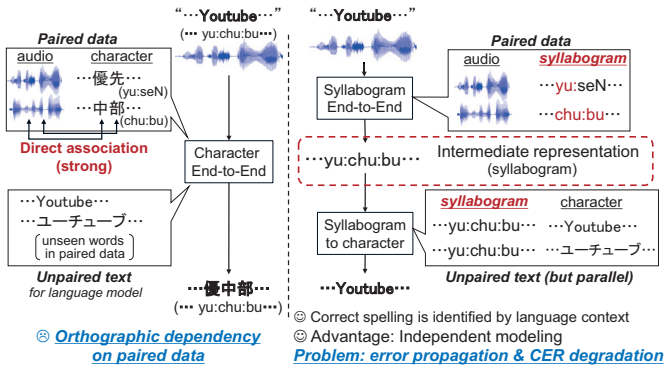

Fig. 1. Orthographic dependency (left) and our focus (right)

ideograms (Kanji), phonograms (Alphabet), and syllabograms (Hiragana/Katakana). The variety of meaningful characters results in many homophones (words with different orthographies / meanings), which are often related to named entities [5]. Thus, misrecognized text have *different meanings from the user's intentions*. Context information or domain knowledge could resolve such ambiguities, and could be given as unpaired text information.

One promising solution to the *orthographic dependency* problem is to cascade the syllable ASR (S-ASR) and syllable-to-character translation (SCT) models. In this approach, the pronunciations of unpaired texts are represented as a sequence of *interpretable* syllabograms (pronunciation symbols). The S-ASR model first recognizes a sequence of syllabograms from the entire audio utterance, and then the SCT model "translates" the sequence into corresponding spelling characters (Fig. 1 right). The advantage of this role-specific approach is that each model can be designed and trained independently without encountering out-of-character issues, except for the syllabary. S-ASR depends only on a pronunciation system and is independent of any target domain or spelling, which enhances ASR portability. We can flexibly design, fine tune, or replace the SCT model for each application domain by utilizing the latest neural architectures, token units and small/large amounts of unpaired text.

However, such an approach based on two separate models usually suffers from error propagation between the models. Since the output of S-ASR usually includes recognition errors,

979-8-3315-7206-8/25/$31.00 ©2025 IEEE

the SCT model may translate several misrecognized syllabograms into incorrect spelling characters. In addition, the local best estimation made by each model is not always the globally optimal estimation for the entire process.

We propose 1) a probabilistic model to integrate the S-ASR and SCT models and 2) a robust training method for the SCT model. The former provides a global score for better estimations on the basis of two-layer probabilistic models, i.e., S-ASR and SCT, during the block-wise beam search. The score from a language model (LM) is also added (shallow fusion) to assess the validity of the estimated characters by the SCT model. The latter introduces a pronunciation error generator that modifies the input syllabograms of training data by simulating deletion, insertion, and substitution errors of S-ASR, which may improve the robustness of the SCT model. The effect of replacing such errors with a mask token is demonstrated through experiments. We also integrate the ASR results from C-ASR and our approach to improve the performance further on the basis that C-ASR works well for words in paired data and our approach works well for words only in unpaired text.

The following are our contributions:

- We confirmed the effectiveness of a probabilistic integration via syllabograms for Japanese dialogue speech to resolve the orthographic dependency on the paired data
- We investigated the impact of LM-fusion and mask token training of the SCT model on the character error rate (CER).
- We performed a comprehensive evaluation of our and open ASR models over open Japanese corpora. Our model parameters have been publicly released[1].

B. Related Work

The studies on utilizing unpaired text are categorized into LM fusion, speech generation, context biasing, and neural architecture extension. These techniques do not compete with our work; instead, they can further enhance the performance of our approach. There are several LM fusions depending on the network connectivity [6]: shallow fusion [7], deep fusion [8], and cold fusion [9]. Text-to-speech technologies have been used to generate speech signals from unpaired text data to increase paired data [10], [11]. Specific architectures for vocabulary extension [5], [12] and context bias [13]–[16] have also been studied.

Several studies have explored two-layer approaches using intermediate representations in cross-lingual ASR [17], [18] and streaming Chinese ASR [19], not Japanese ASR. In those studies, error propagation was mitigated by using the confidence vector of each phoneme from phoneme ASR [17] or error simulation training of the syllable-to-character conversion model [19]. Recognition methods such as these are typically based on a 1-best search (cascade / end-to-end), and their neural architectures also differ from ours, e.g., RNN-T vs. transformer encoder-decoder with external LM fusion. In addition, our experiments additionally examined the utilization of the mask-token in the SCT model training.

[1]examples – https://github.com/ouktlab/espnet_asr_models, and model parameters – https://huggingface.co/ouktlab

II. Baselines

A. Character End-to-End ASR with LM Shallow Fusion

Character end-to-end ASR (C-ASR) directly estimates a character sequence $c_{1:L} = [c_1, ..., c_L]$ of length L from a sequence of speech feature vectors $\mathbf{x}_{1:T} = [\mathbf{x}_1, ..., \mathbf{x}_T]$ of length T. Each feature vector at frame t, $\mathbf{x}_t \in \mathbb{R}^D$, is a D-dimensional vector. Neural networks are used to model the posterior probability $p(c_{1:L}|\mathbf{x}_{1:T})$ through supervised training using paired data.

The character-level LM $p_{\text{c}}(c_{1:L})$ is fused during decoding to utilize unpaired text resources. In the hybrid CTC/attention model [20], the following score function J_{w2c} is evaluated in order to obtain the best estimation $\hat{c}_{1:L}$

$$\hat{c}_{1:L} = \text{argmax}_{c_{1:L}} J_{\text{w2c}}(c_{1:L}|\mathbf{x}_{1:T}), \text{and} \quad (1)$$

$$J_{\text{w2c}}(c_{1:L}|\mathbf{x}_{1:T}) = w_{\text{c},1} \ln p_{\text{ctc}}(c_{1:L}|\mathbf{x}_{1:T}) + w_{\text{c},2} \ln p_{\text{att}}(c_{1:L}|\mathbf{x}_{1:T}) + w_{\text{c},3} \ln p_{\text{c}}(c_{1:L}), \quad (2)$$

where p_{ctc}, p_{att}, and p_{c} represent the probabilities from the CTC model, attention model, and character-level LM, respectively. $w_{\text{c},i} (i = 1, 2, 3)$ represents the mixture weight, and their sum equals 1.

B. Syllable ASR and Syllable-to-Character Translation

The cascading method estimates $c_{1:L}$ from $\mathbf{x}_{1:T}$ via a syllabogram sequence $s_{1:M} = [s_1, ..., s_M]$ of length M as an intermediate representation. It requires two models: an S-ASR model $p(s_{1:M}|\mathbf{x}_{1:T})$ and an SCT model $p(c_{1:L}|s_{1:M})$. Since the lengths M and L are usually different, the SCT model is assumed to be an encoder-decoder architecture in this paper.

A simple decoding can be achieved through the following cascaded estimation:

$$\hat{s}_{1:M} = \text{argmax}_{s_{1:M}} J_{\text{w2s}}(s_{1:M}|\mathbf{x}_{1:T}), \text{and} \quad (3)$$

$$\hat{c}_{1:L} = \text{argmax}_{c_{1:L}} J_{\text{s2c}}(c_{1:M}|\hat{s}_{1:M}), \quad (4)$$

where J_{w2s} and J_{s2c} represent the actual score functions of S-ASR and SCT, respectively. Here, we assume that J_{w2s} has the same form as J_{w2c} except that the mixture weights in J_{w2s} are denoted as $w_{\text{s},i} (i = 1, 2, 3)$, and a syllabogram-level LM p_{s} is used instead of p_{c}. As for J_{s2c}, the score function is defined as the log probability, i.e., $\ln p(c_{1:L}|s_{1:M})$.

III. Proposed Method

A. Probabilistic Integration: Formulation

S-ASR and SCT models are integrated via syllabograms, which are considered to be latent variables from a *probabilistic* perspective to clarify the global score function. Here, the conditional joint probability $p(c_{1:L}, s_{1:M}|\mathbf{x}_{1:T})$ is the ideal score function. Since we are interested in not only $c_{1:L}$ but also $s_{1:M}$ in terms of pronunciation recognition for the dialogue system, we estimate both $c_{1:L}$ and $s_{1:M}$ from $\mathbf{x}_{1:T}$ based on the joint probability.

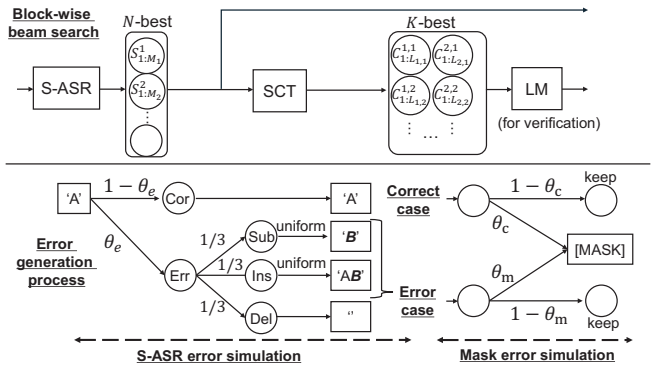

Fig. 2. Block-wise beam search (top) and error generation process for SCT training (bottom)

We decompose the joint probability by considering the recognition process as follows:

$$p(c_{1:L}, s_{1:M}|\mathbf{x}_{1:T}) = p(c_{1:L}|s_{1:M})p(s_{1:M}|\mathbf{x}_{1:T}) \quad (5)$$

where $p(s_{1:M}|\mathbf{x}_{1:T})$ and $p(c_{1:L}|s_{1:M})$ represent the S-ASR and SCT models, respectively. Since this decomposition still considers several possibilities of $s_{1:M}$ using the same S-ASR and SCT models, its recognition performance is expected to be better than that of the cascading method.

B. Block-wise Beam Search and Score Function

We use a block-wise beam search to find an approximate estimation of $c_{1:L}$ because it is difficult to evaluate the marginalized probability in Eq. (5) over the latent variables. Here, $s_{1:M}$ and $c_{1:L}$ are estimated sequentially in a block (utterance)-wise manner by applying an N-best beam search in each stage. This process is illustrated in Fig. 2 (top). For example, the N-best hypotheses, $\{s^n_{1:M_n}\}^N_{n=1}$, are estimated in the S-ASR stage. Then, K-best hypotheses, $\{c^{k,n}_{1:L_{k,n}}\}^K_{k=1}$, are estimated in the SCT stage for each hypothesis $s^n_{1:M_n}$. Finally, the hypothesis with the highest score is selected as the best estimation, $\hat{c}_{1:L}$.

Our score function based on Eq. (5) with character-level LM shallow fusion becomes

$$\begin{aligned} J_{\text{w2sc}}(c_{1:L}, s_{1:M}|\mathbf{x}_{1:T}) &= v_1 J_{\text{w2s}}(s_{1:M}|\mathbf{x}_{1:T}) \\ &+ v_2 J_{\text{s2c}}(c_{1:L}|s_{1:M}) + v_3 \ln p(c_{1:L}), \end{aligned} \quad (6)$$

where v_i $(i = 1, 2, 3)$ are weights and their sum is 1. Note that the cascade approach corresponds to the setting of performing a 1-best search in each stage.

C. Training Strategy for Robust SCT Model

The SCT training uses a *parallel* corpus that includes pairs of character sequences and corresponding syllabogram sequences. Adding *errors* to the input sequences makes the SCT model robust, as it enables the model to learn from erroneous inputs in advance. Our main concern is the impact of simulation strategies and mask-token replacement on ASR performance.

We introduce an error generator that consists of an S-ASR error simulator and a masking error simulator, and we apply it to each syllabogram in the input, as shown in Fig.2 (bottom). The former generates {deletion, insertion, and substitution} errors, while the latter replaces several tokens with a mask token. Three parameters control the error properties. The parameter θ_{e} controls the simulated error rate of S-ASR. For substitution and insertion errors, the corresponding token is replaced with a different syllabogram. While θ_{c} represents the probability that a correct syllabogram is incorrectly masked, θ_{m} represents the probability that an erroneous syllabogram is correctly masked. We can disable the masking error simulator by setting $\theta_{\text{c}} = 0$ and $\theta_{\text{m}} = 0$. All errors from the S-ASR error simulator, except deletion errors, are correctly masked by setting $\theta_{\text{m}} = 1$; i.e., it is equivalent to full *mask-token replacement*.

D. Integration of ASR results from C-ASR and S-ASR with SCT

The integration of the ASR results from C-ASR and S-ASR with SCT is expected to improve performance even further. This is because C-ASR works well for words in the paired data while S-ASR with SCT works well for words only in unpaired text. We focused on their recognition scores because they are correlated with the uncertainty of the recognition results to some extent.

Our criterion is based on the following score difference between the two models

$$g = \frac{J_{\text{w2c}}(\hat{c}^c_{1:L_c}|\mathbf{x}_{1:T})}{L_c} - \alpha \frac{J_{\text{w2sc}}(\hat{c}^{sc}_{1:L_{sc}}, \hat{s}_{1:M}|\mathbf{x}_{1:T})}{L_{sc}} - \beta, \quad (7)$$

where $\hat{c}^c_{1:L_c}$, $\hat{c}^{sc}_{1:L_{sc}}$, and $\hat{s}_{1:M}$ represent the recognition results with top score from C-ASR and S-ASR with SCT models, respectively. α and β are parameters for score-range and offset adjustment, and they are determined by using the development set. Each model's score is normalized by the recognized character length L_c or L_{sc}. $\hat{c}^c_{1:L_c}$ is accepted as a final recognition result if $g > 0$, and $\hat{c}^{sc}_{1:L_{sc}}$ is accepted otherwise. Note that each length-normalized recognition score is less than or equal to 0 because it is basically the log probability of a discrete random variable.

IV. Experiment

The experiment investigated the performance of the following methods on human-machine dialogue speech recorded in real environments: our approach and C-ASR trained under the same conditions including training set, each process in our approach, and open ASR models. The open ASR models were Reazonspeech ESPnet v2[2], Whisper large v3 [21], and Rinna Nue [22], [23] with the default settings. These models were trained on over 10,000 hours of audio data.

A. Data Set

Training paired data for C-ASR and S-ASR: The training speech data (approximately 12,500 hours in total) mainly consisted of seed paired corpora and their augmented versions (Table I). The seed corpora, totaling 900 hours, mainly

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

TABLE I
TRAINING SET FOR EACH MODEL

Model	C-ASR, S-ASR	LM, SCT
Data	Paired data (audio & text)	Text in paired data + unpaired text
Size	Over 12,500 hours (audio)	Over 400 million characters
Corpus	10 corpora (over 900 hours)	10 corpora + BCCWJ, Wiki40b-ja
Others	+ Data augmentation	+ Text standardization

consisted of ten *public* Japanese speech corpora with transcriptions: CSJ [24], S-JNAS, TWM, JEIDA-JCSD, ETL-WD, RIKEN-DLG[3], APP, APPDIC[4], SLC-3[5], and JVS [25]. The augmented data were generated by applying transformations to the seed corpora, such as speech-rate perturbation, convolution with impulse responses, and addition of non-speech signals to improve robustness to noise. Impulse responses that were measured at 540 positions in a real room (RT_{20} 640 ms) were used. The non-speech signal data primarily consisted of MUSAN [26], WHAM! (train set) [27], and the ProSoundEffects corpus[6]. We also included 20 hours of additional pure tone signals, white/brown/pink noise, babble noise, and recorded environmental noises. The signal-to-noise ratio (SNR) was randomly selected from $-10, -5, 0, 5, 10$, and 20 dB.

Training text data for SCT and LMs: We used transcriptions from the paired data, BCCWJ text [28], unpaired Wiki-40B (ja) text [29] and Wikipedia title data (Tab. I). The syllabograms (Katakana) of the wiki text were obtained using the Japanese morphological analyzer Mecab [30] with the NEologd [31] and UniDic [32] dictionaries. The spellings of words and representations of numbers were standardized to some extent in accordance with the transcription rule of CSJ. For example, alphabetical words were represented by Katakana for LM of C-ASR.

Development set (dev set): The *dev set* consisted of *dialogue recordings in CSJ* [24] that were not used for model training to determine the stopping epoch for SCT model training and some weight parameters of each score function. We selected eight stereo recordings, totaling 90 minutes of speech involving 16 persons.

Test set: Our main *open* test sets consisted of four Hazumi{1712, 1902, 1911, 2105} [33] corpora (11.5 hours in total), as they are real recordings of human-machine interactions in a spoken dialogue system. The system provides topics, and users talk about a TV show, manga, singers, and other subjects on the basis of *their own experiences*. **Users uttered words that were not in our training paired data (*mismatched*), making these corpora reasonable for evaluating our approach to the orthographic dependency problem.** In addition, ASR models had to be robust since reverberations and background noises were present in the audio signals. We also used a situated spoken dialogue corpus[1] (PASD-1), and clean and noisy eval sets from the CSJ [24] for comparison. These sets also included words that were matched or mismatched to the training paired data or unpaired text. The properties of each data-set are summarized in Table II. The noisy CSJ sets were generated by convolving impulse responses from RWCP-SSD [34] with the CSJ *eval sets* and adding non-speech signals from ESC-50 [35] and WHAM! (test set) [27] at a SNR of 20 dB. Note that the speech signals in Hazumi, CSJ, and noisy CSJ were segmented into utterances, while those in PASD-1 were not. We eliminated the *tsu* and *uec* sets from PASD-1 due to strong channel interference.

[3]https://research.nii.ac.jp/src/list.html
[4]https://www.atr-p.com/products/sdb.html
[5]https://alaginrc.nict.go.jp/slc-outline.html
[6]Pro Sound Effects Library. http://www.prosoundeffects.com

B. Configuration of Model

C-ASR and S-ASR of the baseline and proposed method: The loss function and architectures of C-ASR and S-ASR were identical, following the ESPnet CSJ recipe with transformer ASR and LM [36]. This is because conformer ASR was not robust against unseen noises [37]. The vocabulary sizes, including silence and special symbols, were approximately 3,700 for C-ASR and 170 for S-ASR. *SpecAug* were used to improve robustness. The number of training epochs was set to 30 with a default scheduler, and the models were then tuned with fixed learning rates of 2.0×10^{-5} or/and 2.0×10^{-6}. The final parameters were obtained by averaging parameters over several epochs from each learning rate. The numbers of parameters of ASR and LM were 97M and 50M, respectively.

SCT in the proposed method: T5 for conditional generation [38] 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 [39]. We experimented with 6 and 12 layers and selected 12 for its the better performance. The number of parameters was 110M. The number of training epochs was set to 10 with fixed learning rates of first 10^{-4} and then 10^{-6}, and the final parameters were obtained by averaging parameters over five epochs. The vocabulary size was approximately 11,100 Japanese characters, defined by JIS X 0213 (Japanese Industrial Standard for coded character sets). The other configurations were the default ones.

Other parameters in the proposed method: The error simulator parameter, θ_{e}, was set to 0.15, similarly to masked LM settings [40]. If the mask error simulator was enabled, θ_{c} was set to 0.05 and $[0.15, 0.50, 0.85, 1.00]$ were tested as θ_{m}. The LM architecture in Eq. (6) and its tokenizer followed the GPT-NeoX small model [41], and the LM was trained from scratch using the same settings as SCT model training. The number of training epochs was set to 10 with fixed learning rates of first 1.0×10^{-4} and then 1.0×10^{-6}, and the final parameters were also obtained by averaging parameters over five epochs. The number of parameters was 153M. α in Eq. (7) was set to 3.17 to minimize the deletion error for the dev set with $\beta = 0$.

Decoding and other parameters: The relative weights in C-ASR and S-ASR were determined using the CER for the dev set; $(w_{c,1}, w_{c,3}) = (0.21, 0.30)$ and $(w_{s,1}, w_{s,3}) = (0.19, 0.35)$. By minimizing the deletion error for the dev set, the SCT model trained with $\theta_{\mathrm{m}} = 1.00$ was selected, and the SCT v_2 and LM weights v_3 were set to 0.75 and 0.02, respectively. The beam size was set to 40 in C-ASR and S-ASR

TABLE II
FEATURES OF OPEN TEST SETS

Hazumi: *Our target* - real spoken dialogue data	
Mismatched with the paired training data	
Style	Dialogue (Human-Machine)
Type	*Non-task*
Topic	User's experience: travel, TV, manga, singer, ...
CSJ eval sets: standard Japanese ASR evaluation data	
Matched with the paired training data	
Style	Solo speech (not dialogue)
Type	Technical research presentation
Topic	Acoustic signal and language processing
PASD-1: other situated dialogue data	
Mismatched with the paired training data	
Style	Dialogue (Human-Human)
Type	Task specific
Topic	Schedule managing, travel guide, ...

TABLE III
MAIN RESULTS: CHARACTER ERROR RATE (%) UNDER SAME TRAINING DATA CONDITION.

Corpus		Hazumi				
	Set	1712	1902	1911	2105	Total
Baseline	B1: C-ASR (with transformer LM)	14.6	12.2	10.7	11.6	12.15
	B2: Cascade: S-ASR(1-best) + SCT(1-best)	13.5	12.1	10.7	11.6	11.88
Proposed	P1: S-ASR(5-best) + SCT(1-best)	13.5	12.2	10.6	11.3	11.80
	P2‡: S-ASR(5-best) + randSCT(1-best)	13.4	11.6	10.2	11.0	11.44
	P3‡: S-ASR(5-best) + maskSCT(1-best)	13.3	11.7	10.3	10.9	11.45
	P4†: S-ASR(5-best)+maskSCT(1-best) + LM	13.1	11.6	10.3	10.8	11.37
	P5†: S-ASR(5-best)+maskSCT(5-best) + LM	13.1	11.6	10.2	10.8	11.36
	P6: Integration (B1+P5) $(\alpha, \beta) = (3.17, 0)$	**12.7**	**11.1**	**9.6**	**10.4**	**10.84**
	C1: Integration (B1+P5) $(\alpha, \beta) = (1, 0.095)$	13.7	11.7	10.2	10.9	11.51
# of characters						227,526

TABLE IV
COMPARISON WITH OPEN END-TO-END ASR MODELS: DELETION ERROR RATE / CHARACTER ERROR RATE IN %

Corpus	Hazumi	CSJ	Noisy CSJ	PASD-1
O1: Reazon v2	8.79 / 13.06	9.83 / 14.53	11.09 / 18.99	11.7 / 17.7
O2: Whisper [21]	9.84 / 16.41	9.56 / 13.35	11.45 / 18.58	17.7 / 24.0
O3: Nue [22]	17.33 / 22.29	19.86/ 25.85	22.20 / 33.70	34.5 / 44.1
O4: ESPnet (CSJ)	7.32 / 22.46	1.31 / 4.06	8.40 / 27.77	5.6 / 26.8
B1: C-ASR	4.78 / 12.15	1.31 / 3.72	1.90 / 5.93	4.3 / 11.3
B2: Cascade	4.50 / 11.88	1.41 / 4.25	2.06 / 6.56	4.5 / 11.6
P5: S-ASR with SCT	4.17 / 11.36	1.38 / 4.02	1.95 / 6.19	4.5 / 11.4
P6: Integration	**3.67 / 10.84**	**1.16 / 3.58**	**1.70 / 5.75**	**3.9 / 11.3**
# of characters	227,526	115,742	115,742	19,230

and 15 in SCT. The experiments were conducted on Nvidia RTX A6000 GPUs.

C. Results

Table III presents the results under the same training conditions, and Table IV lists the open ASR model conditions[7]. The statistical significance of the CER differences between two methods was assessed using the probability of improvement (POI) in % via the bootstrap method [42] in the Kaldi toolkit [43] under the 95% confidence interval settings.

Table III shows that our probabilistic approach (P1) outperformed the baselines: C-ASR (B1) and cascade 1-best search (B2). The total CER of 5-best search (P1) was 0.35 and 0.08 points lower than that of C-ASR (B1) and cascade (B2), with POIs of 99 % and 95%, respectively. Thus, simple cascade approach falls into local estimation.

Table III also shows experimental comparisons of different implementations of mask tokens (P1, P2, and P3). Here, *SCT*, *randSCT*, and *maskSCT* denote the SCT models trained with no error simulator, only the S-ASR simulator, and full mask-token replacement, respectively. randSCT (P2) also outperformed P1 by 0.36 points with a POI of 100%. Since no significant difference was observed between randSCT (P2) and maskSCT (P3) with a POI of 56% (‡), full mask-token replacement is sufficient and simple for robust SCT model training.

The effectiveness of LM-fusion (P4, P5) is shown in the middle part of Table III. The LM-fusion (P4) further improved CER over that of P3, with a POI of 100%, and the additional K-best search in SCT (P5) improved CER over 1-best search in SCT (P4) with a POI of 90% (†). The probabilistic propagation for SCT process slightly contributes to CER.

Finally, the integration of S-ASR with C-ASR (P6) outperformed the C-ASR (B1) and S-ASR with SCT approach (P5) from the bottom part of Table III. The CER of the Integration (P6) improved by 0.52 and 1.04 points in total compared with C-ASR (B1) or S-ASR with SCT (P5) alone, respectively. These results indicate that the two approaches complement each other. We found that the integration based on the simple score difference (C1) setting $(\alpha, \beta) = (1, 0.095)$ degraded performance, where $\beta = 0.095$ minimized the deletion error for the dev set with $\alpha = 0$. Therefore, the parameter settings (α, β) and the score criterion itself were important to improve CER further.

From Table IV, we found that the open models had many deletion errors and our approach (P5 and P6) outperformed them. In particular, our integration (P6) performed best, and the performance on even PASD-1 was not degraded compared with C-ASR (B1) or S-ASR with SCT (P5) alone. The deletion errors of Reazon (O1), Whisper (O2), and Nue (O3) were generally above 8%, with several non-filler words misrecognized (deleted). These experiments did not allow for context information over several utterances, which may have negatively affected the performance of Whisper and Nue. Note that *Reazon was able to recognize several words that our model missed, suggesting that a combination of these models could be beneficial in practice.*

Table IV also shows the noise robustness of our model (P5) by comparing its performance with that of ESPnet with the CSJ recipe (O4) for CSJ and noisy CSJ sets. On the other hand, S-ASR with SCT (P5) performed similarly to or slightly worse than C-ASR (B1), likely due to the lower accuracy of the SCT model. This performance gap could be reduced by having a deeper SCT model correctly learn technical terms from the CSJ corpus, as the CER (Katakana) of S-ASR on the noisy CSJ sets was below 3.0%.

The limitations of our method are that 1) the evaluation was limited to a specific language (Japanese) and 2) the

[7]All recognition results of each ASR model were standardized with best effort by unifying the representations of numbers and by adjusting the orthographic variants to the transcriptions of each set.

network architectures were not optimized. Regarding language dependency, we believe that research like ours is essential for *language localization*. As for a practical implementation as spoken dialogue systems, integration with streaming ASR will be effective in terms of the tradeoff between the recognition accuracy and the processing latency.

V. Conclusion

We proposed a probabilistic integration of S-ASR and SCT models via syllabograms to mitigate orthographic dependency. Experimental results on Japanese dialogue sets demonstrated the effectiveness of our approach compared with C-ASRs. As for future work, we aim to replace unreliable recognized syllabograms from S-ASR with mask tokens to propagate the uncertainty to the SCT model. We will also investigate the ASR performance for spoken dialogue data recorded in the real babble noise environment.

Acknowledgment

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

References

[1] M. Goto, K. Itou, and S. Hayamizu, "A real-time filled pause detection system for spontaneous speech recognition," in *Proc. of Eurospeech*, 1999, pp. 227–230. DOI: 10.21437/Eurospeech.1999-60.

[2] M. Watanabe, K. Hirose, *et al.*, "Filled pauses as cues to the complexity of upcoming phrases for native and non-native listeners," *Speech Communication*, vol. 50, no. 2, pp. 81–94, 2008.

[3] M. Lease, M. Johnson, and E. Charniak, "Recognizing disfluencies in conversational speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1566–1573, 2006. DOI: 10.1109/TASL.2006.878269.

[4] K. Hara, K. Inoue, *et al.*, "Prediction of turn-taking using multitask learning with prediction of backchannels and fillers," in *Proc. of Interspeech*, 2018, pp. 991–995.

[5] Y. Sudo, K. Hata, *et al.*, "Retraining-free customized ASR for enharmonic words based on a named-entity-aware model and phoneme similarity estimation," in *Proc. of Interspeech*, 2023, pp. 491–495.

[6] H. Inaguma, J. Cho, *et al.*, "Transfer learning of language-independent end-to-end ASR with language model fusion," in *Proc. of ICASSP*, 2019, pp. 6096–6100. DOI: 10.1109/ICASSP.2019.8682918.

[7] D. Zhao, T. N. Sainath, *et al.*, "Shallow-fusion end-to-end contextual biasing," in *Proc. of Interspeech*, 2019, pp. 1418–1422.

[8] C. Gulcehre, O. Firat, *et al.*, "On integrating a language model into neural machine translation," *Computer Speech & Language*, vol. 45, pp. 137–148, 2017, ISSN: 0885-2308.

[9] A. Sriram, H. Jun, *et al.*, "Cold fusion: Training Seq2Seq models together with language models," in *Proc. of Interspeech*, 2018, pp. 387–391. DOI: 10.21437/Interspeech.2018-1392.

[10] T. Hayashi, S. Watanabe, *et al.*, "Back-translation-style data augmentation for end-to-end ASR," in *Proc. of SLT*, 2018, pp. 426–433.

[11] L. Ye, G. Cheng, *et al.*, "Improving recognition of out-of-vocabulary words in E2E code-switching ASR by fusing speech generation methods," in *Proc. of Interspeech*, 2022, pp. 3163–3167.

[12] M.-T. Nguyen, D. P. Nguyen, *et al.*, "Improving speech recognition with jargon injection," in *Proc. of SIGDIAL*, 2024, pp. 490–499.

[13] J. Suh, I. Na, and W. Jung, "Improving domain-specific ASR with LLM-generated contextual descriptions," in *Proc. of Interspeech*, 2024, pp. 1255–1259. DOI: 10.21437/Interspeech.2024-377.

[14] M. Bhattacharjee, I. Nigmatulina, *et al.*, "Contextual biasing methods for improving rare word detection in automatic speech recognition," in *Proc. of ICASSP*, 2024, pp. 12 652–12 656.

[15] X. Gong, A. Lv, *et al.*, "Contextual biasing speech recognition in speech-enhanced large language model," in *Proc. of Interspeech*, 2024, pp. 257–261. DOI: 10.21437/Interspeech.2024-965.

[16] R. Huang, M. Yarmohammadi, *et al.*, "Improving neural biasing for contextual speech recognition by early context injection and text perturbation," in *Proc. of Interspeech*, 2024, pp. 752–756.

[17] H. Xue, Q. Shao, *et al.*, "TranUSR: Phoneme-to-word transcoder based unified speech representation learning for cross-lingual speech recognition," in *Proc. of Interspeech*, 2023, pp. 216–220.

[18] S. Feng, M. Tu, *et al.*, "Language-universal phonetic representation in multilingual speech pretraining for low-resource speech recognition," in *Proc. of Interspeech*, 2023, pp. 1384–1388.

[19] X. Wang, Z. Yao, *et al.*, "Cascade RNN-transducer: Syllable based streaming on-device mandarin speech recognition with a syllable-to-character converter," in *Proc. of SLT*, 2021, pp. 15–21.

[20] S. Watanabe, T. Hori, *et al.*, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[21] A. Radford, J. W. Kim, *et al.*, "Robust speech recognition via large-scale weak supervision," in *Proc. of ICML*, 2023.

[22] Y. Hono, K. Mitsuda, *et al.*, "Integrating pre-trained speech and language models for end-to-end speech recognition," in *Proc. of Findings of ACL*, 2024, pp. 13 289–13 305.

[23] ——, *Rinna/Nue-ASR*. [Online]. Available: https://huggingface.co/rinna/nue-asr.

[24] K. Maekawa, "Corpus of spontaneous Japanese: Its design and evaluation," in *Proc. of ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[25] S. Takamichi, K. Mitsui, *et al.*, *JVS corpus: Free Japanese multi-speaker voice corpus*, 2019. arXiv: 1908.06248 [cs.SD].

[26] D. Snyder, G. Chen, and D. Povey, *MUSAN: A Music, Speech, and Noise Corpus*, arXiv:1510.08484v1, 2015. eprint: 1510.08484.

[27] G. Wichern *et al.*, "WHAM!: Extending speech separation to noisy environments," in *Proc. of Interspeech*, 2019, pp. 1368–1372.

[28] M. Kikuo, M. Yamazaki, *et al.*, "Balanced corpus of contemporary written Japanese," *Language Resources and Evaluation*, no. 48, pp. 345–371, 2014.

[29] M. Guo, Z. Dai, *et al.*, "Wiki-40B: Multilingual language model dataset," in *Proc. of LREC*, 2020, pp. 2431–2439.

[30] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to Japanese morphological analysis," in *Proc. of EMNLP*, 2004, pp. 230–237.

[31] T. Sato *et al.*, "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*, 2017, NLP2017-B6–1.

[32] T. Ogiso, M. Komachi, *et al.*, "UniDic for early middle Japanese: A dictionary for morphological analysis of classical Japanese," in *Proc. of LREC*, 2012, pp. 911–915.

[33] K. Komatani *et al.*, "Analyzing differences in subjective annotations by participants and third-party annotators in multimodal dialogue corpus," in *Proc. of SIGDIAL*, 2023, pp. 104–113.

[34] S. Nakamura, K. Hiyane, *et al.*, "Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition," in *Proc. of LREC*, 2000.

[35] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proc. of ACM-MM*, Oct. 13, 2015, pp. 1015–1018.

[36] S. Watanabe, T. Hori, S. Karita, *et al.*, "ESPnet: End-to-end speech processing toolkit," in *Proc. of Interspeech*, 2018, pp. 2207–2211.

[37] R. Takeda *et al.*, "Flexible evidence model to reduce uncertainty mismatch between speech enhancement and asr based on encoder-decoder architecture," in *Proc. of APSIPA ASC*, 2023, pp. 1830–1837.

[38] C. Raffel, N. Shazeer, *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 1, Jan. 2020, ISSN: 1532-4435.

[39] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. of ICLR*, 2018.

[40] J. Devlin, M.-W. Chang, *et al.*, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of NAACL*, 2019, pp. 4171–4186.

[41] T. Zhao and K. Sawada, *Rinna/japanese-gpt-neox-small*. [Online]. Available: https://huggingface.co/rinna/japanese-gpt-neox-small.

[42] M. Bisani and H. Ney, "Bootstrap estimates for confidence intervals in ASR performance evaluation," in *Proc. of ICASSP*, 2004, pp. 409–409.

[43] D. Povey, A. Ghoshal, *et al.*, "The kaldi speech recognition toolkit," in *Proc. of SLT*, 2011.