Fast and Speaker-Independent Utterance Selection for ASR-Free CALL Systems of Minority Languages

Takaki Koshikawa*, Akinori Ito*†, Takashi Nose*
* Graduate School of Engineering, Tohoku University, Sendai, Japan
E-mail: {koshikawa.takaki.s3@dc., akinori.ito.a2@, takashi.nose.b7@}tohoku.ac.jp
† Advanced Institute of So-Go-Chi Informatics, Tohoku University, Sendai, Japan

***Abstract*—This paper addresses utterance selection in a spoken dialogue system designed for CALL systems of minority languages, where no ASR or TTS modules are available. To enable speaker-independent and efficient matching, we employ Self-Supervised Learning (SSL) speech representations and introduce a lightweight method called Speech2Token (S2T), which converts feature vectors into discrete token sequences. To reduce the cost of similarity matching, we further apply three compression techniques: Deduplication (Dedup), Optimum Code Summarization (OCS), and Greedy Sequential Optimization (GSO). Experiments on Japanese and Kaqchikel datasets show that using SSL features improves utterance selection accuracy by 20 points compared to conventional features. Furthermore, tokenizing the features and applying Deduplication achieves comparable accuracy while reducing response time by nearly 50%, demonstrating the method's practicality for efficient spoken dialogue systems in minority language settings.**

I. Introduction

There are approximately 7,000 languages spoken worldwide, but nearly half are endangered due to declining speaker populations. These are known as minority or low-resource languages [1]. Since language preserves ethnic identity and cultural heritage [2], its loss threatens cultural diversity and human legacy [3]. In response, "language revitalization" efforts aim not only to archive speech data but also to increase speakers through learning opportunities [4], as seen in work on Komi [5] and Savosavo [6]. Native speaker instruction is ideal, but often unfeasible for minority languages. Thus, computer-assisted language learning (CALL) systems are gaining traction. For example, "Ojibwemodaa" supports pronunciation and grammar training using ASR [7]. Language learning involves reading, writing, listening, and speaking [8], with speaking becoming increasingly important [9], [10]. Yet speaking-oriented systems—spoken dialogue systems—remain limited for minority languages due to the cost of building ASR/TTS. Many also lack sufficient data or writing systems [11].

To address these challenges, the Sample-Based Spoken Dialogue System (SBSDS) has been proposed [12]. It selects responses by matching input speech to pre-recorded samples based on acoustic similarity, without ASR or TTS. Aimed at practicing basic vocabulary and sentence patterns—not open-domain chatting—SBSDS suits low-resource environments. Still, computational cost and robustness to speaker variation remain issues [13].

Recently, self-supervised learning (SSL) has become popular for extracting high-dimensional, context-aware representations from speech [14]–[16]. These features capture various linguistic and paralinguistic information and have shown strong performance in many speech tasks [17], [18]. Meanwhile, discretization of SSL features into token sequences—via vector quantization or clustering—has gained attention for enabling NLP-style modeling in speech processing [19], [20]. Such tokenizers are typically designed for general-purpose tasks such as ASR, TTS, or pretraining of speech models.

In this study, we focus on improving sample-based spoken dialogue systems (SBSDS), which require fast and robust utterance selection in low-resource settings. Specifically, we leverage expressive SSL features and introduce a lightweight tokenization method called Speech2Token, which converts SSL features into tokens using k-means clustering and compresses them for efficient utterance selection. Furthermore, we use Kaqchikel—a Mayan language spoken in Guatemala—as our evaluation target. Although it has around 500,000 speakers and ongoing efforts to standardize and teach it, a rapid shift to Spanish and limited resources make it an ideal language for testing dialogue technologies in minority languages [21], [22].

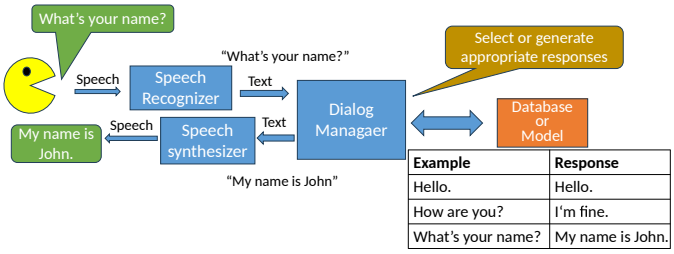
II. Related Work

A. Sample-Based Spoken Dialogue System (SBSDS)

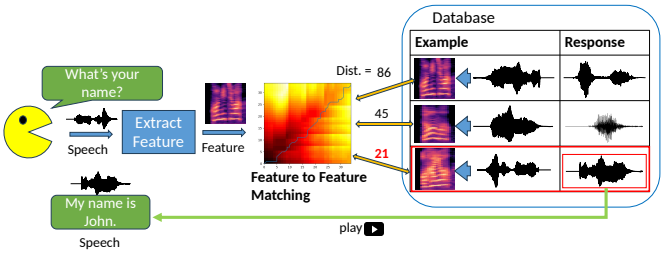
A sample-based spoken dialogue system (SBSDS) is a dialogue system that computes acoustic similarity between the user's input utterance and a database of pre-recorded utterances, and selects the most similar response. Fig. 1a shows the structure of a conventional spoken dialogue system. In such systems, the input speech is first transcribed into text via ASR, followed by response selection or generation by the dialogue manager. Finally, the selected response is synthesized into speech and presented to the user using a TTS system.

In contrast, the structure of an SBSDS is shown in Fig. 1b. The SBSDS database consists of paired samples of example utterances and corresponding response utterances. When a user provides an input utterance, the system matches it to the most similar example utterance and returns the associated response utterance as the system's output. SBSDS are a type of retrieval-based dialogue system, characterized by selecting responses based on similarity rather than language understanding or generation. Because they do not require ASR or TTS mod-

979-8-3315-7206-8/25/$31.00



(a) The framework of ordinary spoken dialogue system



(b) The framework of sample-based spoken dialogue system

Fig. 1: Architectural comparison of spoken dialogue systems: (a) an ordinary system, which relies on ASR and TTS; and (b) a sample-based system, which selects a response from pre-recorded utterance samples without ASR or TTS

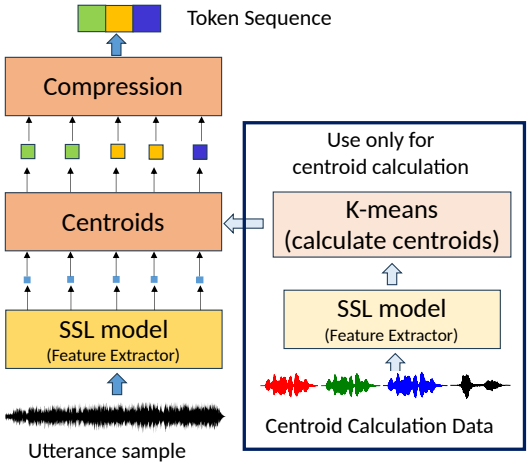


Fig. 2: The overview of Speech2Token

ules, SBSDS are particularly suitable for use in low-resource language environments.

In conventional SBSDS, similarity has typically been computed using acoustic features such as mel-frequency cepstral coefficients (MFCC) and phoneme posteriorgrams (PPG), along with algorithms such as dynamic time warping (DTW) and continuous DP matching (CDP) [12]. These methods demonstrated the feasibility of recognition-free spoken dialogue. However, challenges remain, including long computation times for matching and limited robustness to speaker and language variation.

B. Self-Supervised Speech Representations and Tokenization

Self-supervised learning (SSL) has become a standard approach for speech feature extraction in recent years. SSL models are trained on large-scale speech corpora using pretext tasks and acquire high-dimensional acoustic representations that encode information such as phonemes, vocabulary, speaker identity, and prosody.

For instance, wav2vec 2.0 [14] learns representations by predicting masked regions of latent features using quantized targets. HuBERT [15] extends this by generating pseudo-labels via offline clustering, refined through iterative masked prediction. XLS-R [16] extends wav2vec 2.0 to the multilingual setting, trained on 128 languages and over 4 million hours of speech. In contrast, ContentVec [23] targets speaker-independent, content-only features by using pitch/timbre perturbation and speaker conditioning.

Meanwhile, several studies have proposed discretizing SSL features using vector quantization or clustering to enable NLP-style speech modeling [19], [20]. These tokenizers often employ learned quantizers such as vector-quantized autoencoders or transformer-based modules, and are primarily designed for ASR, TTS, or speech model pretraining.

In this study, we adopt the above SSL models as feature extractors and propose a lightweight tokenization method, Speech2Token, based on k-means clustering. Unlike general-purpose tokenizers, Speech2Token is tailored for fast utterance selection in SBSDS and does not require training or reconstruction objectives.

III. Proposed Method

A. Speech2Token

In this study, we propose a lightweight tokenization method called Speech2Token, which converts continuous representations from SSL models into discrete token sequences for efficient utterance selection. Unlike learned tokenizers based on vector-quantized autoencoders or transformer-based modules [19], [20], Speech2Token uses only k-means clustering.

As shown in Fig. 2, we use the centroids computed by k-means clustering for quantizing the features. Specifically, the centroid vectors are computed using a separate training dataset via k-means. The speech used in the dialogue system is first converted into feature vectors through an SSL model. Each feature vector is mapped to the nearest centroid based on distance, and the corresponding centroid index is used as a token. Subsequently, compression is applied to the token sequence to obtain the final compressed sequence. The compression methods are described in the following subsection.

B. Token Sequence Compression

To compress token sequences and enable fast utterance selection, we propose three compression methods: Deduplication (Dedup), Optimum Code Summarization (OCS), and Greedy Sequential Optimization (GSO).

1) Deduplication: Dedup is the simplest method. It shortens the sequence by collapsing consecutive identical tokens into a single instance. For example, the token sequence $\{0, 0, 1, 1, 1, 3, 3, 2\}$ is transformed into $\{0, 1, 3, 2\}$ through

Deduplication. This method is lightweight, with computational complexity of $O(n)$ for a sequence of length n. It is robust to repeated or elongated sounds but lacks the ability to control the compression rate, limiting its potential for high compression precision.

2) Optimum Code Summarization: OCS compresses a token sequence into a target length m by applying a two-stage dynamic programming approach inspired by continuous speech recognition [24]. Suppose the token sequence $\{q_1, q_2, \ldots, q_n\}$ is obtained via Speech2Token from a sequence of N-dimensional feature vectors $\{x_1, x_2, \ldots, x_n\}$. Each token $q_i \in Q$ corresponds to a centroid vector $c(q_i) \in \mathbb{R}^N$, where $|Q| = K$. The goal is to compress this sequence into a new sequence $R = \{r_1, \ldots, r_m\}$. Each r_i corresponds to a segment $s_i = (b_i, e_i)$ that satisfies the following conditions:

$$b_1 = 1, \quad e_m = n \tag{1}$$

$$e_{i-1} + 1 = b_i \quad (i > 1) \tag{2}$$

The compression error is defined as:

$$\varepsilon(R, S) = \sum_{i=1}^{m} \sum_{j=b_i}^{e_i} \|c(r_i) - c(q_j)\|^2 \tag{3}$$

For each segment $[a, b]$, the optimal representative token $r(a, b)$ is given by:

$$r(a, b) = \arg\min_{r \in Q} \sum_{k=a}^{b} \|c(r) - c(q_k)\|^2 \tag{4}$$

Using dynamic programming, the optimal compressed sequence $(R(i, j), S(i, j))$ for compressing $q_1, \ldots, q_j$ into i tokens is given

$$R(1, j) = \{r(1, j)\}, \quad S(1, j) = \{(1, j)\} \tag{5}$$

$$\hat{k} = \arg\min_{i-1 \le k < j} \varepsilon(R(i-1, k) \cup \{r(k+1, j)\}, S(i-1, k) \cup \{(k+1, j)\}) \tag{6}$$

$$R(i, j) = R(i-1, \hat{k}) \cup \{r(\hat{k}+1, j)\} \tag{7}$$

$$S(i, j) = S(i-1, \hat{k}) \cup \{(\hat{k}+1, j)\} \tag{8}$$

By repeating this procedure for $i = m$ and $j = n$, the optimal compressed sequence $(R(m, n), S(m, n))$ is obtained. The computational complexity is $O(n^2 m)$, but this method enables precise control of compression while preserving information.

3) Greedy Sequential Optimization: GSO is a top-down greedy method that begins with the entire token sequence as a single segment. The algorithm iteratively selects the segment and split point (b, k, e) that yields the largest error reduction when divided into two segments (b, k) and $(k+1, e)$, continuing until the desired number of segments m is reached.

For each candidate split, the error reduction is computed as:

$$\Delta = \varepsilon_{\text{orig}} - (\varepsilon_{\text{left}} + \varepsilon_{\text{right}}) \tag{9}$$

The error ε for a segment (b, e) is defined as the sum of squared distances between each token vector $c(q_j)$ and its representative token $c(r)$, selected as the closest centroid to the segment's mean vector:

$$\varepsilon = \sum_{j=b}^{e} \|c(q_j) - c(r)\|^2 \tag{10}$$

While this method is suboptimal compared to OCS, it achieves relatively good compression with lower computational cost $O(nm)$ by greedily applying local error minimization.

TABLE I: Example of Japanese sentences

No.	ID	Sentence(JP)	Sentence(EN)
001	10001	*Yahhô*	Hey
002	10001	*Yâ*	Hi
...	...	...	...
040	10102	*Anata ha dare*	Who are you?
041	10102	*Anata no namae ha*	What's your name?
...	...	...	...

TABLE II: Voice libraries used in the experiment

Library	Gender
Amehare Hau Meimei Himari Namine Ritz Shikoku Metan	Female
Aoyama Ryusei Kenzaki Mesuo Kurono Takehiro Shirakami Kotaro	Male

IV. Experiments

A. Material

For evaluation of utterance selection, we used speech data in Japanese and Kaqchikel. The Kaqchikel data consisted of actual recorded utterances, as exemplified in Table III. A total of 274 utterances were used, each assigned one of 44 semantic IDs. Multiple utterances with the same or similar meanings were given the same ID.

The Japanese dataset was constructed by synthesizing speech from 482 sentences used in previous research on chat-based dialogue systems [25], using eight different speakers (four male and four female) from the VoiceVox speech synthesis library. Example sentences and speaker information are shown in Tables I and II, respectively. Japanese was selected as the evaluation language because it was also used in prior studies [12], [13], and because all speakers shared the same sentence list, enabling controlled speaker-wise comparison.

B. Experiment Overview

As illustrated in Fig. 3, utterance selection experiments were conducted on both Japanese and Kaqchikel data. For each input utterance, the distance to all database utterances was computed, and the one with the smallest distance was selected as the system response. If the semantic ID of the selected response matched that of the input, it was considered correct.

TABLE III: Example of Kaqchikel sentences

No.	ID	Sentence(Kaq)	Sentence(EN)
001	10001	xsaqär tat	Good morning, sir
002	10001	xsaqär matyox	Good morning
...	...	...	...
034	10016	akuchi' at k'äs wi?	Where do you live?
035	10016	akuchi' aqajon awochoch?	Where are you renting?
...	...	...	...

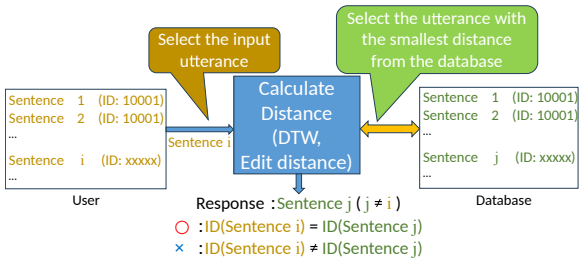


Fig. 3: Experiment overview

For SSL-based features, DTW was used to compute distances, while for token sequences, token error rate (TER) based on edit distance was used.

Note that, in this study, all utterances in the database were treated as example utterances. We did not use example–answer pairs, which are the original structure of SBSDS. This allows us to evaluate the similarity-based matching performance without being influenced by dialogue design or response naturalness. In other words, we focus purely on whether the selected response has the same semantic ID as the input. This design follows the SBSDS framework while enabling reproducible and quantitative evaluation.

For Japanese, all combinations of two out of eight speakers were used to form user–database speaker pairs. The procedure for each pair was as follows: (1) each utterance from the user speaker was treated as input, (2) distances to all utterances of the database speaker were computed, (3) database utterances with exactly the same text as the input were excluded from candidates, and (4) the utterance with the smallest distance was selected. The result was judged correct if the semantic IDs matched. This design choice allows us to better highlight the differences in retrieval performance among various feature types. The final score was computed as the average accuracy across all 56 pairs. For Kaqchikel, the evaluation followed the same procedure, but since the speakers did not share a common sentence list as in Japanese, no speaker-pair comparisons were conducted. Instead, all 274 utterances were used as both inputs and database entries.

To estimate the theoretical upper bound of selection accuracy, we simulated the presence of transcriptions and computed character-level distances using character error rate (CER). The results showed that the theoretical upper-bound accuracy was 71.40% for Japanese and 43.07% for Kaqchikel.

TABLE IV: Details of investigated SSL models

Model	Output dim.	Layer	Corpus (Langs)	Hours
ContentVec[1]	768	12	LibriSpeech (1)	960
HuBERT[2]	768	12	LibriSpeech (1)	960
wav2vec2.0[3]	768	12	LibriSpeech (1)	960
XLS-R[4]	1024	24	VP, MLS, CV, VL, BBL (128)	436K

TABLE V: Accuracy and best layers for each SSL model

Feature type	Kaqchikel		Japanese	
	Layer	Acc. (%)	Layer	Acc. (%)
Theoretical Value	—	43.07	—	71.40
MFCC[12]	—	7.30	—	17.97
PPG[12]	—	15.71	—	34.06
ContentVec	L8	**35.04**	L12	**57.18**
HuBERT	L8	34.31	L12	56.89
wav2vec 2.0	L5	31.39	L4	47.79
XLS-R	L8	32.12	L17	52.37

C. DTW-based Model and Layer Selection

In this experiment, we tested various SSL models and identified the best-performing output layers using DTW. The models used are listed in Table IV. For comparison, conventional MFCC and PPG features were also included. MFCC were computed with 12 static coefficients plus Δ parameters (24 dimensions total), using a sampling rate of 16 kHz, window length of 25 ms, and frame shift of 10 ms. PPG were extracted using phoneme classifiers trained on JNAS [26] and TIMIT [27], and concatenated frame-wise [28].

Table V shows the best-performing layer and corresponding accuracy for each model. In Kaqchikel, ContentVec layer 8 achieved the highest accuracy, while in Japanese, layer 12 performed best. In both languages, ContentVec outperformed conventional features by approximately 20 percentage points, demonstrating substantial improvement in utterance selection accuracy.

XLS-R, despite being trained on multilingual data, did not surpass ContentVec in accuracy. This suggests that the diversity of training data does not always lead to better performance. Fig. 4 shows that ContentVec, designed for speaker-independent representation, achieves more stable accuracy across speakers compared to conventional features, confirming its suitability for SBSDS. Based on these results, subsequent experiments use ContentVec layer 8 for Kaqchikel and layer 12 for Japanese.

D. Evaluation of Centroid Calculation Conditions for Speech2Token

We evaluated the impact of centroid training conditions as shown in Table VI. Token sequences were generated for each condition and used in utterance selection without compression. The best result for Japanese was with 30 minutes of Japanese

[1]https://huggingface.co/lengyue233/content-vec-best
[2]https://huggingface.co/facebook/hubert-base-ls960
[3]https://huggingface.co/facebook/wav2vec2-base-960h
[4]https://huggingface.co/facebook/wav2vec2-xls-r-300m

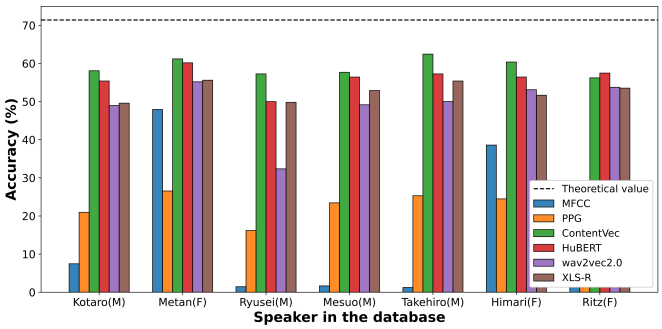


Fig. 4: Accuracy results on Japanese for different SSL models and database speakers (user: Amehare Hau (F))

TABLE VI: Variable settings for centroid calculation conditions

Condition	Value
Language	Japanese[29], English[30], Chinese[31], Spanish[32]
Amount of calculation data	30 min, 300 min
Number of clusters	64, 128, 256, 512

data and 128 clusters; for Kaqchikel, 30 minutes of Spanish data and 256 clusters worked best (Table VII). As a baseline, we also evaluated SpeechTokenizer [20], trained on 960 hours of LibriSpeech with 1024 clusters, which showed notably lower accuracy than S2T in both languages.

These results suggest that using a language related to the target or sharing its socio-linguistic context may improve tokenization accuracy. For Kaqchikel, although not genealogically related to Spanish, strong influence from Spanish vocabulary and grammar may explain its effectiveness. Moreover, ContentVec-based tokenization retained speaker independence, and performance loss due to tokenization was minimal. SpeechTokenizer's lower accuracy may partly stem from its large number of clusters (1024). While S2T uses fewer (128–256), the gap likely also reflects its task-specific SSL features and simple, non-learned clustering, which better support utterance-level similarity in SBSDS.

E. Token Compression Comparison and Final Evaluation

To evaluate the effectiveness of the proposed compression methods (Dedup, OCS, GSO), For OCS and GSO, we conducted utterance selection experiments using token sequences compressed at four different rates: $\rho \in \{0.7, 0.6, 0.5\}$. This allowed us to investigate the trade-off between compression and system performance.

Table VIII shows that response time generally decreased as the compression rate ρ decreased, but selection accuracy also dropped, notably at $\rho = 0.5$. OCS and GSO required additional compression time (1.0 and 0.3 seconds, respectively), while Deduplication incurred almost none and maintained accuracy close to the SSL baseline, reducing response time by nearly 50%. This is likely due to two factors: (1) the computational cost of edit distance scales quadratically with sequence length,

TABLE VII: Best centroid training conditions and accuracy

Method	Data	Cls.	Kaq (%)	Jpn (%)
SpeechTokenizer[20]	LS960h	1024	21.90	37.77
Speech2Token(Ours)	30min (Spa)	256	**32.85**	—
Speech2Token(Ours)	30min (Jpn)	128	—	**60.34**

TABLE VIII: Accuracy and response time for each feature type and compression ratio

Feature type		ρ	Kaqchikel Acc.(%)	Kaqchikel Time(sec)	Japanese Acc.(%)	Japanese Time(sec)
Theoretical Value		—	43.07	—	71.40	—
MFCC[12]		—	7.30	3.70	17.97	8.09
PPG[12]		—	15.71	3.73	34.06	8.21
SSL feats.		—	**35.04**	3.79	57.18	9.29
Token	w/o comp.	1.0	32.85	3.61	60.34	7.88
	Dedup	0.70(Kaq) 0.75(Jpn)	31.75	1.48	**60.79**	4.59
	OCS	0.7	33.94	3.29	58.84	5.15
		0.6	30.29	2.70	54.79	4.00
		0.5	27.37	2.23	48.87	3.54
	GSO	0.7	28.83	2.48	59.06	4.57
		0.6	26.64	1.88	53.46	3.43
		0.5	25.18	**1.41**	51.34	**2.96**

and (2) Dedup achieved compression rates of 0.70 and 0.75 for Kaqchikel and Japanese, respectively, which may lead to a reduction of the overall computational cost to approximately 50% (i.e., $0.7^2 \approx 0.49\times$, $0.75^2 \approx 0.56\times$) of the original.

These results confirm that SSL features improve accuracy over conventional feaatures, and that tokenization followed by Deduplication further reduces response time without sacrificing accuracy

V. Conclusion

We proposed a speaker-independent and fast utterance selection method for SBSDS using tokenized and compressed SSL features. Experiments showed that ContentVec improved selection accuracy by over 20 points compared to conventional features, and that Deduplication maintained this accuracy while reducing response time by nearly 50% with negligible compression cost. By combining tokenization with Deduplication, the method achieves both efficiency and robustness. These results demonstrate its practicality for lightweight dialogue systems in minority settings without the need for ASR or TTS. In the future, we aim to develop a practical CALL system with utterance-response pairs, user studies, and broader evaluations across languages.

VI. Acknowledgement

Part of this work was supported by JSPS KAKENHI JP24H00085, JP21H00895, and JP23K20725.

References

[1] L. Campbell and A. Belew, *Cataloguing the world's endangered languages*. Routledge London, 2018, vol. 711.

[2] S. Chiblow and P. J. Meighan, "Language is land, land is language: The importance of indigenous languages," *Human Geography*, vol. 15, no. 2, pp. 206–210, 2022.

[3] L. A. Grenoble and L. J. Whaley, *Saving languages: An introduction to language revitalization*. Cambridge University Press, 2005.

[4] L. Hinton, *Language revitalization: An overview*. The Green Book of Language Revitalization in Practice, 2001.

[5] C. Gerstenberger *et al.*, "Instant annotations in ELAN corpora of spoken and written Komi, an endangered language of the Barents sea region," in *Proc. ComputEL-2*, 2017, pp. 57–66.

[6] B. Jana *et al.*, "Multimodal language use in savosavo," *Pragmatics*, vol. 27, no. 2, pp. 173–206, 2017.

[7] M. Hermes *et al.*, "Ojibwe language revitalization, multimedia technology, and family language learning," *Language Learning & Technology*, vol. 17, no. 1, pp. 125–144, 2013.

[8] S. Hofstetter, "Vocabulary and the four skills. pedagogy, practice, and implications for teaching vocabulary," *ELT Journal*, vol. 76, no. 2, pp. 297–300, 2022.

[9] N. H. Rahmat *et al.*, "Functions of speaking in english language & speaking anxiety," *European Journal of English Language Teaching*, vol. 6, no. 1, pp. 87–103, 2020.

[10] J. Juangsih *et al.*, "The needs analysis of four primary language skills in developing Japanese teaching materials for tourism purposes," *Jurnal Pendidikan Bahasa dan Sastra*, vol. 20, no. 2, pp. 185–196, 2021.

[11] O. Scharenborg *et al.*, "Speech technology for unwritten languages," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 964–975, 2020.

[12] A. Ito, "Spoken dialogue system development without speech recognition towards language revitalization," in *Proc. IIHMSP*, 2022, pp. 393–404.

[13] A. Ito, "Confidence-based utterance selection for a recognizer-free spoken dialogue system," in *Proc. ICMLC*, 2023, pp. 481–484.

[14] A. Baevski *et al.*, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *IEEE/ACM Trans. Audio, Speech & Language Proc.*, vol. 33, pp. 12 449–12 460, 2020.

[15] W.-N. Hsu *et al.*, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Trans. Audio, Speech & Language Proc.*, vol. 29, pp. 3451–3460, 2021.

[16] A. Babu *et al.*, "XLS-R: Self-supervised cross-lingual speech representation learning at scale," in *Proc. INTERSPEECH*, 2022, pp. 2278–2282.

[17] S.-W. Yang *et al.*, "SUPERB: Speech processing universal performance benchmark," in *Proc. INTERSPEECH*, 2021, pp. 1194–1198.

[18] A. Mohamed *et al.*, "Self-supervised speech representation learning: A review," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1179–1210, 2022.

[19] Z. Huang *et al.*, "RepCodec: A speech representation codec for speech tokenization," in *Proc. ACL*, 2024.

[20] X. Zhang *et al.*, "Speechtokenizer: Unified speech tokenizer for speech language models," in *Proc. ICLR*, 2024.

[21] R. M. Brown *et al.*, *La ütz awäch?: Introduction to Kaqchikel Maya language*. University of Texas Press, 2010.

[22] M. Koizumi *et al.*, "Sentence processing in a verb–object–subject language: Evidence from Kaqchikel Maya," *Language, Cognition and Neuroscience*, vol. 29, no. 7, pp. 877–892, 2014.

[23] K. Qian *et al.*, "ContentVec: An improved self-supervised speech representation by disentangling speakers," in *Proc. ICML*, 2022, pp. 18 003–18 017.

[24] H. Komae, "Speech recognition and dynamic programming," *Journal of the Operations Research Society of Japan*, vol. 24, pp. 324–330, 1985, (in Japanese).

[25] Y. Kageyama *et al.*, "Improving user impression in spoken dialog system with gradual speech form control," in *Proc. SIGdial*, 2018, pp. 235–240.

[26] K. Itou *et al.*, "JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research," *Journal of the Acoustical Society of Japan (E)*, vol. 20, no. 3, pp. 199–206, 1999.

[27] J. Garofolo *et al.*, *TIMIT acoustic-phonetic continuous speech corpus*, 1992.

[28] S. Mizuochi *et al.*, "Spoken term detection of zero-resource language using posteriorgram of multiple languages," *Interdisciplinary information sciences*, vol. 28, no. 1, pp. 1–13, 2022.

[29] R. Sonobe *et al.*, "JSUT corpus: Free large-scale Japanese speech corpus for end-to-end speech synthesis," *arXiv*, 2017. eprint: preprintarXiv:1711.00354.

[30] V. Panayotov *et al.*, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.

[31] H. Bu *et al.*, "AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline," in *Proc. O-COCOSDA*, 2017, pp. 58–62.

[32] V. Pratap *et al.*, "MLS: A large-scale multilingual dataset for speech research," in *Proc. INTERSPEECH*, 2020, pp. 2757–2761.