Exploring the Effect of Segmentation and Vocabulary Size on Speech Tokenization for Speech Language Models

Shunsuke Kando[1], *Yusuke Miyao*[1], *Shinnosuke Takamichi*[2,1]

[1]Graduate School of Information Science and Technology, The University of Tokyo, Japan
[2]Faculty of Science and Technology, Keio University, Japan
{skando,yusuke}@is.s.u-tokyo.ac.jp, shinnosuke_takamichi@keio.jp

Abstract

The purpose of speech tokenization is to transform a speech signal into a sequence of discrete representations, serving as the foundation for speech language models (SLMs). While speech tokenization has many options, their effect on the performance of SLMs remains unclear. This paper investigates two key aspects of speech tokenization: the segmentation width and the cluster size of discrete units. First, we segment speech signals into fixed/variable widths and pooled representations. We then train K-means models in multiple cluster sizes. Through the evaluation on zero-shot spoken language understanding benchmarks, we find the positive effect of moderately coarse segmentation and bigger cluster size. Notably, among the best-performing models, the most efficient one achieves a 50% reduction in training data and a 70% decrease in training runtime. Our analysis highlights the importance of combining multiple tokens to enhance fine-grained spoken language understanding.

Index Terms: speech language models, spoken language understanding

1. Introduction

With the recent breakthroughs in large language models for textual natural language processing, speech language models (SLMs) have emerged as a new paradigm for spoken language processing [1–4]. SLMs are built by training language models on top of discrete speech representations (called "discrete units"). The process of converting a speech signal into a discrete unit sequence is called "speech tokenization". Speech tokenization is typically performed by quantizing representations obtained from self-supervised learning (SSL) models [5–7]. Leveraging the rich representations of SSL models, SLMs trained on these discrete units have demonstrated strong performance in zero-shot spoken language understanding (SLU) [8], spoken dialogue [9], speech-to-speech translation [10], and other related tasks.

Various speech tokenization techniques have been proposed to enhance SLM performance. Generative Spoken Language Modeling (GSLM) and its variants simply apply K-means clustering to the SSL model representations as is [1,2,11]. However, since SSL model representations typically correspond to approximately 20 ms speech, the resulting discrete unit sequence tends to be long. This severely affects the training of the Transformer-based language model [12] as computation cost increases quadratically with respect to the sequence length. Besides, previous study suggests that speech SSL representations primarily encode phonetic rather than semantic feature [13], which might impair the capability of SLMs on a deeper understanding of spoken language. To address these issues, previous research has invented speech tokenization techniques that segment input speech into fixed or variable-width units before discretization [14–16]. While segmentation reduces sequence length, it might cause a loss of information preserved in the original representation, and there is no clear agreement on the optimal tokenization scheme for this tradeoff and its reason.

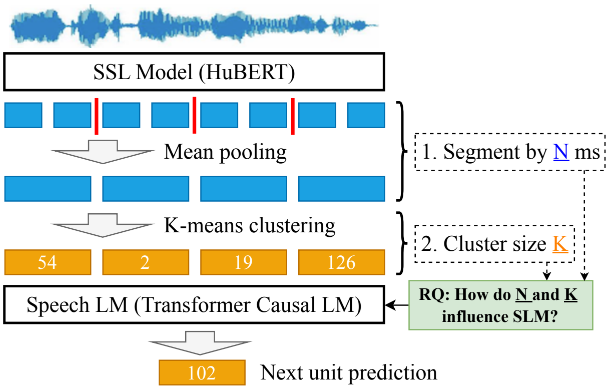


Figure 1: *Overview of our research. First, we extract continuous speech representation from the SSL model. We add segments to the representation sequence by N ms and pooled them. We apply K-means clustering to pooled representations with the cluster size of K. By training SLMs in multiple settings of N and K, we explore the optimal choice for spoken language understanding.*

This paper examines two key aspects of speech tokenization: the segmentation width and the cluster size of discrete units. As depicted in Figure 1, we first segment the SSL representation sequence in a fixed width and pooled features within each segment to obtain coarser representations. Using these pooled representations, we then train K-means models to generate discrete unit sequences. By applying multiple segmentation widths and varying the cluster sizes of the K-means model, we explore the optimal configurations for zero-shot SLU tasks.

Through comparative experiments, we find the positive effect of segmenting by moderately coarse width and making cluster size bigger at the same time. We qualitatively suggest that larger segmentation width requires a larger vocabulary to accurately represent input speech. Notably, a large segmentation setting reduces sequence length, enabling more lightweight training without sacrificing performance. We also observe that specific benchmarks have different optimal settings, highlighting the importance of combining multiple tokens for SLU. Besides fixed-width segmentation, we also investigate variable segmentation based on linguistic units (i.e., phonemes, sylla-

Table 1: *Minimum and maximum number of tokens for every segmentation width. Minimum and maximum values correspond to $K = 2^7$ and $K = 2^{14}$, respectively.*

N	20	40	80	120	160	200	240	280
min	87M	63M	39M	27M	21M	17M	14M	12M
max	127M	77M	42M	28M	22M	17M	14M	12M

bles, and words) and compare their performances. Our results demonstrate that variable segmentation does not show a clear advantage over fixed-width segmentation, suggesting that simpler segmentation methods may be preferable.

The experimental code is made publicly available[1].

2. Tokenization Methods to be Explored

As depicted in Figure 1, we first extract continuous speech representations of input speech using the SSL model. Throughout this study, we used HuBERT [7] as an SSL model and extracted representations from the ninth layer. On top of this representation sequence, we performed speech tokenization in three steps.

1. Segment a sequence by N ms and apply mean pooling.
2. Apply K-means clustering with the cluster size K.
3. Deduplicate units. (e.g. 54 54 54 88 88 3 → 54 88 3)

Since each HuBERT representation corresponds to 20 ms speech, N is chosen as a multiple of 20. We experiment with eight values: {20, 40, 80, 120, 160, 200, 240, 280}, where $N = 20$ corresponds to the original sequence. As N becomes larger, the resulting sequence length becomes shorter by the factor of $N/20$. As for K, the choice of value is not consistent across SLM studies, ranging from {50, 100, 200} [1] to {5k, 10k, 20k} [16]. To comprehensively cover this range, we experiment with eight values of powers of two: from $2^7 = 128$ to $2^{14} = 16384$. In total, we employed $8 \times 8 = 64$ tokenization methods.

Previous studies typically set a smaller cluster size for methods with small segment widths [1,3], while larger segment widths are paired with larger cluster sizes [15,16]. This research supplements the cases of large/small cluster size and small/large segment width, aiming to gain deeper insight into these two aspects.

3. Experimental Setup

3.1. Dataset

As a training set for SLM, we used LibriSpeech [17], a 960-hour English audiobook corpus. Although this dataset is relatively small for SLM studies, our preliminary experiments showed that using a larger dataset (LibriLight [18]; 60k hours audiobook corpus) did not lead to performance improvements. A recent study on SLM based on syllable-level units [16] also supports the use of LibriSpeech, as it reports better performance compared to baselines trained on the larger dataset. Table 1 presents statistics on the training data. As described in Section 2, the sequence length is smaller when N is larger, resulting in a smaller dataset size. We show the minimum and maximum values across K. The smaller K is, the more likely there are repetitions, resulting in fewer tokens after deduplication.

Table 2: *Example pairs from benchmarks for spoken language understanding.*

sBLIMP	(✓ Dogs eat meat, ✗ Dogs eats meat)
sWUGGY	(✓ brick, ✗ blick)
pros-syntax	✓ But in the next breath [PAUSE] he cautioned. ✗ But in the next [PAUSE] breath he cautioned.
pros-lexical	✓ But in the next [PAUSE] breath he cautioned. ✗ But in the next breath he cau [PAUSE] tioned.
tStoryCloze (tSC)	Ana was tanning on the beach. She dozed off in the warm sun. She woke three hours later. Her eyes widened as she looked in the mirror. ✓ Ana was extremely sunburnt. ✗ Michael hoped the new squirrel would fare.

3.2. Model Setup

We trained all K-means models on a 100-hour subset of the LibriSpeech training set. For SLM training, we used OPT [19], a decoder-only Transformer language model. We tuned hyperparameters to match GSLM [1], resulting in 12 layers, 16 attention heads, embedding size of 1024, and FFN size of 4096. To accelerate training, we concatenated all training data and grouped sequences into chunks of 2,048 tokens. Each model was trained for up to 50,000 steps with a batch size of 16 on a single NVIDIA A100 GPU. We applied an early stop when the validation loss did not improve for 1,000 consecutive steps. We report the average scores of SLMs trained with three different random seeds.

3.3. Evaluation

We evaluate SLMs on five types of zero-shot SLU tasks shown in Table 2. Each task consists of pairs of correct and incorrect speech audio samples, and the model is evaluated based on its ability to assign a higher likelihood to the correct sample. The chance rate is 0.5 for all tasks.

sBLIMP [8] assesses the model's grammatical knowledge. Each task is categorized according to 12 types of linguistic phenomena, such as subject-verb agreement or argument structure. **sWUGGY** [8] verifies whether a model has lexical knowledge. It consists of a pair of an existing word and a slightly modified nonce word. **pros-syntax** and **pros-lexical** are from prosaudit benchmark [20], which probes model's capability in handling prosodic information. Stimulus pairs are constructed by inserting a 400 ms pause to the natural and unnatural position within speech. In the pros-syntax task, the correct pause placement corresponds to a prosodic phrase boundary. In the pros-lexical task, the correct placement is at a word boundary, while the incorrect placement is within a word. **Topic SC (tSC)** [2] tests whether a model has commonsense knowledge. This is a spoken version of StoryCloze [21], which rewrites the last sentence of a five-sentence story to produce an incoherent story. Since the original Spoken SC (sSC) dataset is regarded as too challenging [2], we used tSC instead, where the final sentence is randomly chosen from the dataset to generate topically incoherent story[2].

[1] https://github.com/mynlp/speechlm

[2] We also evaluated on sSC but found that all models performed at near-chance rate accuracy.

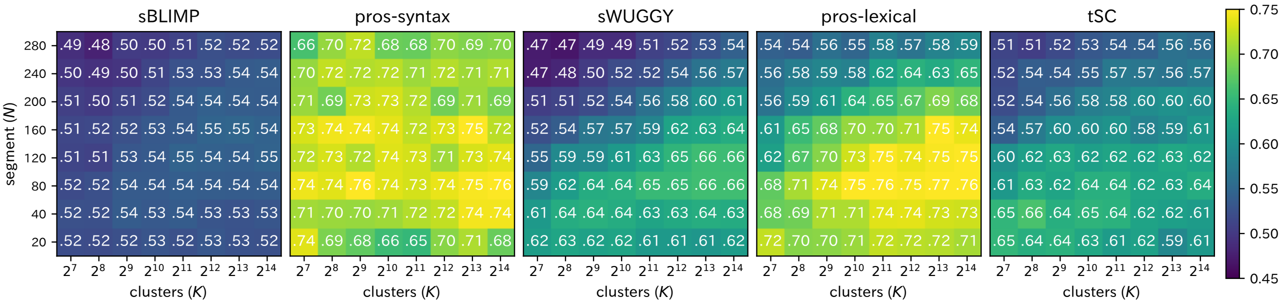


Figure 2: *Main results of SLM performance on zero-shot SLU tasks.*

Table 3: *Best performing K values, average accuracies among five tasks, and training runtimes for* $N = 20, 40, 80, 120$.

N	Best K	Avg. Acc.	Train Runtime
20	2^7 (128)	0.65	12.4 hours
40	2^{13} (8192)	0.65	11.5 hours
80	2^{14} (16384)	0.67	8.3 hours
120	2^{14} (16384)	0.66	6.7 hours

4. Main Result

For simplicity, we denote the configuration with segment width N and cluster size K as (N, K).

Figure 2 shows results on fixed boundary settings. We observe that the best-performing configurations are centered around $(80, 2^{13})$. An exception is tSC, where the optimal setting appears to be around $(40, 2^8)$ (if any), though the differences in accuracy are not significant. For a clear comparison, we identify the best K values based on average accuracy across the five tasks. We focus on relatively small N values $(20, 40, 80, 120)$, as larger N tends to degrade performance. Table 3 shows the summary, including average training runtimes. As we've seen in Table 1, increasing N results in smaller dataset size, which contributes to shorter training runtime. Notably, the best-performing setting $(80, 2^{14})$ reduces the training data by 50% (42M vs. 87M) and the training runtime by 70% (8.3h vs. 12.4h) compared to $(20, 2^7)$ setting.

In terms of benchmarks, while both sBLIMP and pros-syntax are related to syntactic knowledge, the accuracy on pros-syntax is significantly higher than on sBLIMP. This suggests that SLMs have a high capability of handling prosodic features but struggle with a deeper understanding of natural language. For pros-syntax, we observe exceptionally high accuracies even at the largest N values. This may be attributed to the fact that prosaudit inserts a 400 ms pause to stimuli, which is much longer than N. For lexical tasks (sWUGGY and pros-lexical), although pros-lexical shows higher accuracy, both tasks exhibit similar overall trends. On the other hand, tSC results show a slightly different tendency: there seems to be no clear optimal setting. Investigating the underlying factors behind this difference remains for future research.

5. Analysis

5.1. Effect of Larger N and K

Overall, for larger N, accuracy tends to improve with increasing K. This observation is analogous to the relationship between phoneme and morphome: combining a small number of phonemes produces a large number of morphemes [23]. In other words, in smaller N, the model does not require a large vocabulary because there are fewer categories that are essentially distinct. As N increases, the vocabulary size must also be larger to accommodate the growing number of categories.

	0	20	40	60	80	100	120	140	160	180	200	220	240 ... (ms)
"yonder"	**Y**							AA					
"zonder"	**Z**							AA					
✔ $(20, 2^7)$	17	**91**	**77**	**77**	**45**	**45**	118	88	88	100	100	100	
	17	**80**	**40**	**40**	**22**	**118**	118	88	88	100	100	100	
✗ $(80, 2^7)$	**28**				54				32				
	37				54				32				
✔ $(80, 2^{14})$	**10304**				**8642**				**2016**				
	360				**10319**				**10714**				

Figure 3: *An example from sWUGGY where* $(20, 2^7)$ *and* $(80, 2^{14})$ *can solve but* $(80, 2^7)$ *fails. Differences in phoneme or unit are shown in bold. The first row shows the actual stimuli from the dataset and the rest shows unit sequences. Since the dataset does not include phonetic alignments, we annotated them by ourselves using Praat [22].*

To discuss it qualitatively, we extract cases where the combination of (large N, small K) fails but (small N, small K) and (large N, large K) can solve. Figure 3 shows an example from sWUGGY. In this example, SLMs with setting $(20, 2^7)$ and $(80, 2^{14})$ could assign higher likelihood to the existing word "yonder", but $(80, 2^7)$ could not. There is a difference in phoneme up to 140 ms (Y vs. Z), which is captured by settings $(20, 2^7)$ and $(80, 2^{14})$ as the discrete unit sequences differ within this range. However, the setting $(80, 2^7)$ fails to reflect the difference between 80 ms and 140 ms: in this range, it assigns the same unit "54" to both stimuli. This might be attributed to the lack of vocabulary, which is resolved by increasing K from 2^7 to 2^{14}. It would be interesting to investigate whether this effect applies to larger N with much larger K, but that could make training difficult for both the K-means model and SLMs. Future work could investigate on training SLMs with continuous representations, which can be viewed as the limit of discrete representations [24].

5.2. sBLIMP Accuracy Split by Task Type

Figure 2 shows that sBLIMP accuracy is almost chance rate for all settings. This is consistent with findings from previous studies: even much larger SLMs also struggle with sBLIMP [2, 25]. Still, since sBLIMP is a suite of 12 distinct tasks, some of them

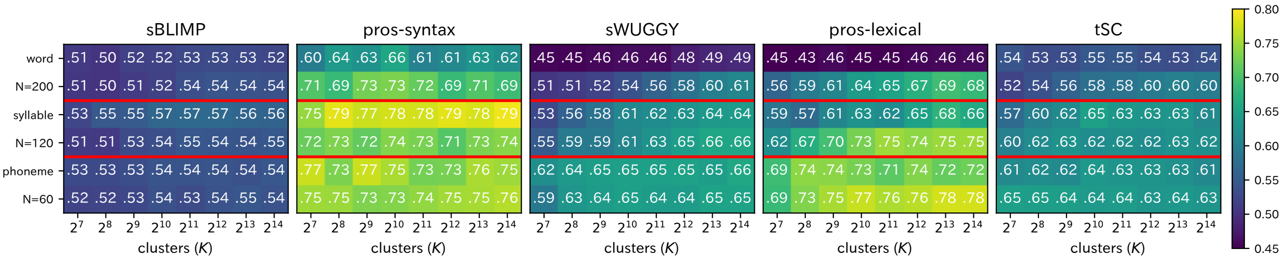
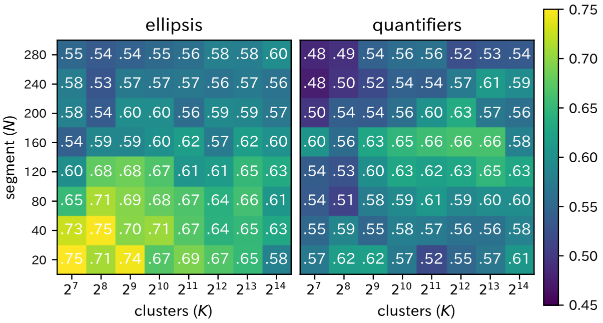


Figure 4: *Results of variable segmentation on phoneme, syllable, and word levels. For comparison, we show the fixed width segmentation results of which N is the same as median of the distribution of variable segmentation.*



Task	Example
ellipsis	✓ Anne's doctor cleans one important book and Stacey cleans a few. ✗ Anne's doctor cleans one book and Stacey cleans a few important.
quantifiers	✓ No boy knew fewer than six guys. ✗ No boy knew at most six guys.

Figure 5: *sBLIMP accuracies split by task type. We show two results (ellipsis and quantifiers) which display unique tendency.*

might be solvable to some extent. We split the accuracy according to the task and found that it is actually the case. We show two examples in Figure 5: "ellipsis" and "quantifiers". "Ellipsis" tests the possibility of omitting expressions from the sentence. "Quantifiers" assesses whether the quantifier is placed in the right position. The results suggest that the best accuracy for both tasks is significantly above chance. Notably, the optimal settings are unique for these tasks: they are located in the vicinity of $(40, 2^8)$ for ellipsis and $(160, 2^{12})$ for quantifiers. This tendency is clearly different from other benchmarks shown in Figure 2. This finding highlights the importance of combining different types of tokens to enhance SLU, which supports previous studies [3, 26].

5.3. Effect of Variable Segmentation Width

While we have discussed the results of fixed-width segmentation, it is natural to segment speech into variable-width segments based on linguistic units, such as phonemes, syllables, and words. Therefore, we trained SLMs on the variable segmentation predicted by unsupervised segmentation methods. Although previous studies have partially attempted this approach [14–16], our goal is to investigate how different levels of linguistic units influence SLM performance under the comparative framework. Also, since variable segmentation poses additional computation costs, we aim to assess whether it is beneficial by comparing it against the fixed-width setting.

We used UnsupSeg [27], Sylber [16], and GradSeg [28] for segmenting speech into phoneme, syllable, and word units, respectively. Similar to the fixed-width segmentation setting, we applied mean pooling to variable-width representations. To compare against fixed-width segmentation settings, we computed medians of each segmentation width distribution[3]. The medians for phoneme, syllable, and word segmentation were 60 ms, 120 ms, and 200 ms, respectively.

Figure 4 shows comparative results of fixed and variable segmentation settings[4]. We observe the positive effect of variable segmentation in limited settings: syllable segmentation on sBLIMP and pros-syntax. Overall, the accuracies were comparable to those of fixed-width segmentation, even significantly impaired in the word segmentation settings. As suggested in [14], inaccurate segmentation may cause performance degradation. Considering the computational cost of unsupervised segmentation, it might be more preferable to use fixed-width segmentation when training SLMs. On the other hand, previous studies suggest learning syllable-level representations and using them instead of raw HuBERT representations for training SLMs, showing impressive performance in SLU tasks [15, 16]. Whether fixed or variable settings, future work could explore the benefits of learning segment-level representations for SLMs within our comparative experimental framework.

6. Conclusion

In this research, we explored the effect of speech tokenization on the SLU capabilities of SLMs. We conducted multiple speech tokenizations based on the combination of the fixed/variable segmentation and the cluster size. Our experiment on fixed-width segmentation suggests the positive effect of moderately coarse segmentation width and bigger cluster size, which contributes to a reduction in both training data size and runtime. We find that the optimal tokenization settings vary across benchmarks, highlighting the importance of combining multiple tokens for further performance of SLMs. We demonstrate that variable-width segmentations basically do not show a clear advantage over fixed-width segmentations. While we conducted a comprehensive set of experiments on speech tokenization, the exact reasons why certain settings are optimal for each benchmark remain unclear. Additionally, our focus was on SLU tasks, and we did not explore other areas such as speech synthesis or speech continuation. We leave these explorations for future work.

[3] Since the distributions of segment width have a long tail, we used median instead of average as a representative value.

[4] We additionally trained SLMs on $N = 60$ setting for comparison.

7. Acknowledgements

This work was supported by JST ACT-X JPMJAX24C9.

8. References

[1] K. Lakhotia, E. Kharitonov, W.-N. Hsu, Y. Adi, A. Polyak, B. Bolte, T.-A. Nguyen, J. Copet, A. Baevski, A. Mohamed, and E. Dupoux, "On Generative Spoken Language Modeling from Raw Audio," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1336–1354, 2021.

[2] M. Hassid, T. Remez, T. A. Nguyen, I. Gat, A. Conneau, F. Kreuk, J. Copet, A. Defossez, G. Synnaeve, E. Dupoux, R. Schwartz, and Y. Adi, "Textually Pretrained Speech Language Models," in *NeurIPS*, Dec. 2023.

[3] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi, and N. Zeghidour, "Audiolm: A language modeling approach to audio generation," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 2523–2533, Jun. 2023.

[4] S. Hu, L. Zhou, S. Liu, S. Chen, L. Meng, H. Hao, J. Pan, X. Liu, J. Li, S. Sivasankaran, L. Liu, and F. Wei, "WavLLM: Towards Robust and Adaptive Speech Large Language Model," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Association for Computational Linguistics, Jan. 2024, pp. 4552–4572.

[5] A. van den Oord, Y. Li, and O. Vinyals, "Representation Learning with Contrastive Predictive Coding," Jan. 2019.

[6] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 12 449–12 460.

[7] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 29, pp. 3451–3460, Oct. 2021.

[8] T. A. Nguyen, M. de Seyssel, P. Rozé, M. Rivière, E. Kharitonov, A. Baevski, E. Dunbar, and E. Dupoux, "The Zero Resource Speech Benchmark 2021: Metrics and baselines for unsupervised spoken language modeling," Dec. 2020.

[9] T. A. Nguyen, E. Kharitonov, J. Copet, Y. Adi, W.-N. Hsu, A. Elkahky, P. Tomasello, R. Algayres, B. Sagot, A. Mohamed, and E. Dupoux, "Generative Spoken Dialogue Language Modeling," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 250–266, 2023.

[10] A. Lee, P.-J. Chen, C. Wang, J. Gu, S. Popuri, X. Ma, A. Polyak, Y. Adi, Q. He, Y. Tang, J. Pino, and W.-N. Hsu, "Direct Speech-to-Speech Translation With Discrete Units," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Association for Computational Linguistics, May 2022, pp. 3327–3339.

[11] E. Kharitonov, A. Lee, A. Polyak, Y. Adi, J. Copet, K. Lakhotia, T. A. Nguyen, M. Rivière, A. Mohamed, E. Dupoux, and W. Hsu, "Text-free prosody-aware generative spoken language modeling," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Association for Computational Linguistics, 2022, pp. 8666–8681.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5998–6008.

[13] K. Choi, A. Pasad, T. Nakamura, S. Fukayama, K. Livescu, and S. Watanabe, "Self-Supervised Speech Representations are More Phonetic than Semantic," in *Proc. Interspeech*, 2024, pp. 4578–4582.

[14] R. Algayres, Y. Adi, T. Nguyen, J. Copet, G. Synnaeve, B. Sagot, and E. Dupoux, "Generative Spoken Language Model based on continuous word-sized audio tokens," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Association for Computational Linguistics, Feb. 2023, pp. 3008–3028.

[15] A. Baade, P. Peng, and D. Harwath, "SyllableLM: Learning Coarse Semantic Units for Speech Language Models," https://arxiv.org/abs/2410.04029v1, Oct. 2024.

[16] C. J. Cho, N. Lee, A. Gupta, D. Agarwal, E. Chen, A. W. Black, and G. K. Anumanchipalli, "Sylber: Syllabic Embedding Representation of Speech from Raw Audio," Oct. 2024.

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015*. IEEE, 2015, pp. 5206–5210.

[18] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, "Libri-light: A benchmark for ASR with limited or no supervision," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*. IEEE, 2020, pp. 7669–7673.

[19] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. T. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, "OPT: open pre-trained transformer language models," *CoRR*, vol. abs/2205.01068, 2022.

[20] M. De Seyssel, M. Lavechin, H. Titeux, A. Thomas, G. Virlet, A. S. Revilla, G. Wisniewski, B. Ludusan, and E. Dupoux, "ProsAudit, a prosodic benchmark for self-supervised speech models," in *INTERSPEECH 2023*. ISCA, Aug. 2023, pp. 2963–2967.

[21] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. F. Allen, "A corpus and cloze evaluation for deeper understanding of commonsense stories," in *NAACL HLT 2016*, K. Knight, A. Nenkova, and O. Rambow, Eds. The Association for Computational Linguistics, 2016, pp. 839–849.

[22] P. Boersma and D. Weenink, "Praat: doing phonetics by computer [computer program]. version 6.4.27," http://www.praat.org/, retrieved 27 January 2025.

[23] A. Martinet, *Elements of General Linguistics*, ser. Phoenix books. University of Chicago Press, 1966.

[24] T. A. Nguyen, B. Sagot, and E. Dupoux, "Are Discrete Units Necessary for Spoken Language Modeling?" *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1415–1423, Oct. 2022.

[25] T. A. Nguyen, B. Muller, B. Yu, M. R. Costa-jussà, M. Elbayad, S. Popuri, P. Duquenne, R. Algayres, R. Mavlyutov, I. Gat, G. Synnaeve, J. Pino, B. Sagot, and E. Dupoux, "Spirit-lm: Interleaved spoken and written language model," *CoRR*, vol. abs/2402.05755, 2024.

[26] J. Shi, H. Inaguma, X. Ma, I. Kulikov, and A. Y. Sun, "Multi-resolution hubert: Multi-resolution speech self-supervised learning with masked unit prediction," in *The Twelfth International Conference on Learning Representations*. OpenReview.net, 2024.

[27] F. Kreuk, J. Keshet, and Y. Adi, "Self-supervised contrastive learning for unsupervised phoneme segmentation," in *Proc. Interspeech*, H. Meng, B. Xu, and T. F. Zheng, Eds. ISCA, 2020, pp. 3700–3704.

[28] T. S. Fuchs and Y. Hoshen, "Unsupervised word segmentation using temporal gradient pseudo-labels," in *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP*. IEEE, 2023, pp. 1–5.