



# Spoken Word2Vec: Learning Skipgram Embeddings from Speech

*Mohammad Amaan Sayeed, Hanan Aldarmaki*

Mohamed Bin Zayed University of Artificial Intelligence

mohammad.sayeed@mbzuai.ac.ae, hanan.alldarmaki@mbzuai.ac.ae

## Abstract

Text word embeddings that encode distributional semantics work by modeling contextual similarities of frequently occurring words. Acoustic word embeddings, on the other hand, typically encode low-level phonetic similarities. Semantic embeddings for spoken words have been previously explored using analogous algorithms to Word2Vec, but the resulting vectors still mainly encoded phonetic rather than semantic features. In this paper, we examine the assumptions and architectures used in previous works and show experimentally how shallow skipgram-like algorithms fail to encode distributional semantics when the input units are acoustically correlated. We illustrate the potential of an alternative deep end-to-end variant of the model and examine the effects on the resulting embeddings, showing positive results of semantic relatedness in the embedding space.

**Index Terms:** speech, word2vec, skipgram, self-supervised

## 1. Introduction

Word embeddings are vector representations that encode distributional relatedness among words, which is correlated with semantic relatedness [1, 2, 3]. Word2Vec algorithms, like skipgram with negative sampling (SGNS) [4] and continuous bag-of-words (CBOW) [5], have been influential in shaping recent progress in statistical NLP, enabling generalization and transfer across tasks and languages, and paving the way to more robust and efficient language models. Whereas text consists of discrete and repeating units, speech signals are composed of continuous and variable signals that need to be processed in order to identify repeating units (e.g. phones/syllables) and word boundaries. Mel Frequency Cepstral Coefficients (MFCCs) or self-supervised neural acoustic models like Wav2Vec [6] are commonly used to extract sequences of acoustic features from continuous speech signals to help distinguish phonetic classes, but they also encode non-linguistic features such as speaker characteristics and environmental acoustics. Supervised learning with large training data that include multiple speakers and noise conditions can lead to intermediate representations that better encode phonetic features and discard irrelevant sources of variation. However, unsupervised and transfer learning are more suitable for low-resource languages, and given the success of unsupervised word embeddings in the text modality, attempts at unsupervised acoustic word embeddings have been recently proposed [7, 8, 9, 10, 11, 12]. These acoustic embeddings encode the phonetic content of each segment (e.g. word segment) and can help in textless NLP applications such as spoken term discovery and query by example search [13]; they could also potentially be used for unsupervised translation as demonstrated in text [3]. While phonetic embeddings arise more naturally in speech models, there have been a few recent attempts at un-

supervised semantic embeddings from speech that are meant to encode semantic relatedness in the manner of their text-based counterparts. The first such attempt is Speech2Vec [14], which follows the Word2Vec context prediction paradigm, but using a sequential encoder-decoder network to process and reconstruct acoustic features of context words. However, the reported results could not be replicated in subsequent studies and were shown to be flawed in [15]<sup>1</sup>. Another attempt was proposed in [16], where they construct semantic embeddings from speech in two stages: first, they create phonetic embeddings from the acoustic features of each word segment, then use the resulting frozen embeddings as input to a skipgram-like model. While some semantic similarities were observed, the results demonstrate mostly phonetic rather than semantic features, demonstrating the difficulty of disentangling semantic from phonetic content of spoken words.

In these models, the words are pre-segmented by force-alignment using transcribed data, and the models are evaluated by grouping words of the same type together. These simplifications are employed to minimize the many sources of ambiguities in speech that further complicate the development and evaluation of semantic embedding. Yet, no existing model provides satisfactory results comparable to text-based embeddings. In this work, we examine the capabilities of architectures employed in previous works in modeling semantic relatedness from speech units. Our results indicate that even with simplifying assumptions, a two-stage approach à la [16] is incapable of learning semantic embeddings since the input units are phonetically correlated. We propose an end-to-end and deep variant of the model that shows far more promising results. The main contributions of this paper can be summarized as follows:

1. We experimentally examine the two-stage architecture proposed in previous work in idealized conditions, and propose an end-to-end variant of the model. We find that a two-stage architecture is incapable of learning distributional features due to the phonetic correlations of the input units, while an end-to-end architecture with sufficient scale and depth can potentially learn distributional semantic features from suitable discretized units.
2. We apply frame-level clustering using MFCC, Wav2Vec2.0, and HuBERT as acoustic features, and analyze the resulting semantic embeddings from the solution outlined above. Our results underline the importance of obtaining discrete and coherent units to move beyond low-level acoustics to corpus-level semantics<sup>2</sup>.

<sup>1</sup>Our own attempts to replicate the Speech2Vec model did not result in semantic embeddings either.

<sup>2</sup>All scripts for replicating our experiments can be found at [github.com/rainmaker29/SpokenWord2Vec](https://github.com/rainmaker29/SpokenWord2Vec)

## 2. Related Work

### 2.1. Word embeddings and subword units

Skip-gram with negative sampling (SGNS) [4] is a canonical example of context-predicting word embedding models, which are obtained by training a network to predict neighboring words and have been shown to be superior to the count-based word vectors previously used for text classification [17]. Context prediction is a simple objective that doesn't require more than raw text for training, and it enables unsupervised and efficient encoding of word distributional semantic relatedness by mapping words that appear in similar contexts into neighboring points in a vector space. SGNS is optimized by distinguishing a positive example from  $k$  negative examples using logistic regression, where the positive examples are neighboring context words and negative examples are randomly drawn from the corpus. Clearly, the model relies on the concept of a coherent and distinct word that can be observed and identified repeatedly and modeled independently from other words, which is achieved using one-hot vectors to uniquely encode each word in the vocabulary. Subword models like FastText [18] which composes word embeddings from character  $n$ -grams, still rely on coherent words and character  $n$ -grams as unique and repeated units. Character-based language models have been successfully implemented, but they require larger networks to model long-term dependencies and some reliance on words or  $n$ -grams for the training objective (e.g. consider the modifications for character LMs in [19], the softmax over words in ELMo [20], and Byte-Pair-Encoding or other subword tokenization schemes used in recent large language models [21]).

In the speech domain, recent works explored the potential of unsupervised speech recognition [22] and spoken language models [23, 24]. These models employ clustering of the acoustic features to obtain discrete units: for example, [22] applied K-means clustering of Wav2Vec2.0 frames for phonetic segmentation before mean-pooling and dimensionality reduction. [24] clustered the acoustic features using K-Means and used the centroid cluster IDs as semantic tokens. HuBERT [25] is a self-supervised acoustic model that relies on clustering for the discrete target units and has been shown to be more effective than continuous acoustic models for phonemic discrimination. These results point towards discretization as the most viable option for learning semantic models of speech.

### 2.2. Speech-based word embeddings

[8] describe an "audio version of Word2Vec", which is essentially a denoising sequential auto-encoder using MFCC features as inputs representing pre-segmented words. Given the nature of the model and training objective, the resulting embeddings simply compress the input features and efficiently encode the phonetic content of the spoken words. Speech2Vec was proposed next [14] with an objective that more closely resembles Word2Vec models; it uses a sequential encoder-decoder network to reconstruct the acoustic features (i.e. MFCCs) of neighboring words. The original paper reports positive results with good performance in several word similarity benchmarks. The resulting pre-trained Speech2Vec embeddings were released<sup>3</sup>, but the code to re-train the embeddings was not. In a recent study, [15] examined these pre-trained embeddings by analyzing their properties and vocabulary content, and concluded that they are most likely produced by a text-based model. For example, the released embeddings show little similarity between homophones

<sup>3</sup><https://github.com/iamyuanchung/speech2vec-pretrained-vectors>

(words with identical pronunciation like 'hail' and 'hale'). Our attempts to replicate the Speech2Vec model confirm that the resulting speech embeddings encode phonetic rather than semantic similarity, more or less similar to previously proposed phonetic embeddings. While the objective function seems logically similar to the Continuous Bag-of-Words model (CBOW), we surmise that the reconstruction of the phonetic features as the target objective diverts the model from the high-level semantic task to encode low-level phonetic features instead. [16] proposed a two-stage model that consists of a phonetic module, followed by a linear skipgram-like embedding module. The phonetic module consists of a phonetic encoder and speaker encoder trained in tandem, where the phonetic encoder is optimized by maximizing reconstruction loss and minimizing speaker discriminability. After training the phonetic module, the resulting phonetic embeddings are frozen and used as input in the second stage. The second module is similar to the standard SGNS model except that the input words are given as phonetic embeddings instead of one-hot vectors. Examining the resulting embeddings, we see that they mostly encode phonetic rather than semantic features.

## 3. Architectures & Scale: Sanity Check

In this section, we attempt to evaluate the potential of the model described in [16] for encoding semantic features in a more simplified setting and with various scales. We examine the architecture in an idealized setting where we use character sequences to represent words instead of acoustic features. Characters are coarsely related to phonemes, but are easier to work with in the following ways: while phones vary across time and frequency domains and can be difficult to segment and identify, each character can be uniquely and perfectly identified and encoded independently from other characters using one-hot vectors. This means that any difficulty we face with character inputs is likely to be amplified with acoustic inputs. On the other hand, positive outcomes with characters will not necessarily generalize to acoustic inputs, but may point to possible fruitful directions for future research.

### 3.1. Models

To mimic the phonetic embeddings used in the first stage of [16], we train a bidirectional GRU auto-encoder with reconstruction loss using character sequences of individual words. The resulting embeddings encode orthographic similarities between words. In the second stage, we freeze these orthographic embeddings and use them as input to a model similar to SGNS, which encodes the input words via linear projection layers, followed by logistic regression to distinguish positive from negative pairs (see Figure 2). The second stage is identical to the original SGNS model, but since the input is continuous, we use multiple layers of linear projection. In another setting, we train the model end-to-end with the same SGNS objective: the character sequences are first embedded into fixed-dimensional vectors using the encoder network, followed by linear projection and logistic regression. For each positive context word, we randomly sample 10 negative words from the corpus.

### 3.2. Data, training, and evaluation scheme

We use the text transcriptions of the train-clean-100 subset of LibriSpeech [26] for training. We start with the following hyperparameters: GRU layers = 1, hidden units = 50, and linear projection layers = 1. We scale the model by multiplying these values by the scaling factor  $s$  to observe the effect of scale on performance, with  $s$  ranging from 1 to 4 (e.g. with  $s=4$ , we

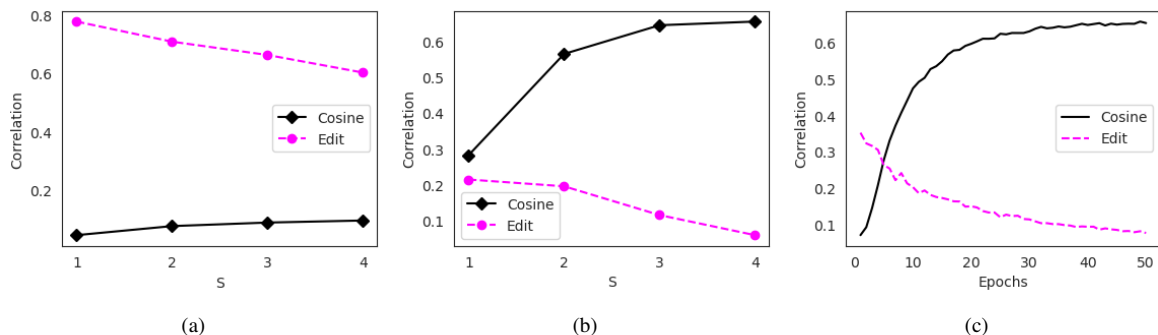


Figure 1: Pearson correlation with target cosine distance and edit distance for different scales ( $s$ ) of the (a) two-stage model vs. (b) end-to-end model. (c) shows the progress of correlation scores throughout the first 50 epochs of training the end-to-end model with  $s=4$ .

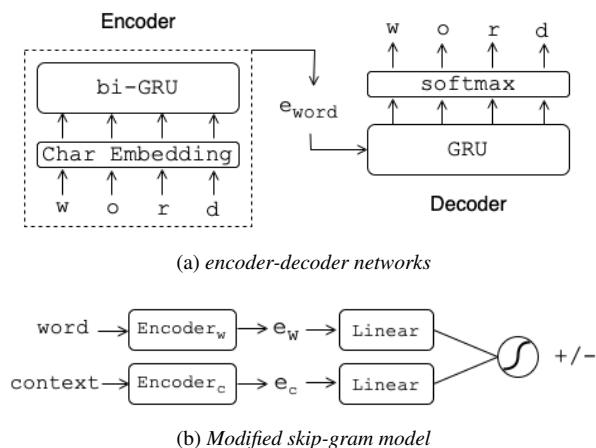


Figure 2: Character-based skip-gram model consisting of GRU character encoder, followed by a multi-layer linear network to handle continuous input. The encoder can be pre-trained using a reconstruction loss with the GRU decoder as in (a), or end-to-end within (b).

have 4 GRU layers, 4 linear layers, and 200 hidden units). For skipgram settings, we used a window of size 5 for the positive examples, and 10 random negative samples without sub-sampling. All models were trained with the Adam optimizer for 100 epochs. To evaluate the resulting embeddings, we train regular SGNS embeddings using the gensim package [27] on the same text, then calculate pairwise cosine distances of a set of word pairs<sup>4</sup>, and report the Pearson correlation coefficient between the distances generated by our models and the target distances, as well as the correlation with Levenshtein edit distance<sup>5</sup>. A high correlation with edit distance indicates that the embeddings encode orthographic (and hence phonetic) features, whereas a high correlation with gensim cosine distances indicate that the model encodes the desired semantic features.

### 3.3. Results

The results are shown in Figure 1 (a) and (b). The final embeddings of a two-stage model (Figure 1 (a)) are highly correlated

<sup>4</sup>We extract a total of 40,000 word pairs based on both semantic and phonetic similarity using a script that will be shared.

<sup>5</sup>We use the Levenshtein python package: <https://pypi.org/project/Levenshtein/>

with edit distance, with almost no correlation with the target gensim embeddings, and we observe little improvement with scale. We conclude that such a two-stage architecture is not suitable for learning semantic embeddings from subword units (e.g. phonetic features) since the frozen embeddings are continuous and highly correlated by their orthographic/phonetic content, which will necessarily be carried forward to the second stage. Training the model end-to-end, on the other hand (Figure 1 (b)), results in much higher correlation with semantic features, and scaling the model results in higher correlation with the target embeddings and lower correlation with edit distance. As seen in Figure 1 (c), which plots the correlation values throughout training epochs for the largest end-to-end model, the model starts with relatively high correlation with edit distances due to the inherent correlations in the input character sequences. However, over time, the model learns to encode the higher-level distributional semantics and ignore low-level similarities. We observe a steady increase in semantic correlation and a decrease in edit distance correlation as long as the training loss is decreasing.

## 4. Deep skipgram model for spoken words

The previous results indicate that an end-to-end approach is more suitable for learning skipgram embeddings from low-level features. However, unlike the discrete character sequences, acoustic sequences are continuous and have their own spurious correlations that can affect the learning process, and may interfere with the ability to learn high-level semantic correlations. To validate these speculations, we trained the same models using acoustic features directly and used them as input to the encoder in place of the character embeddings. In addition, we experimented with frame-level clustering to derive discrete input units.

### 4.1. Acoustic Features

We use 39-dimensional MFCCs (the first 13 DCT coefficients plus their delta and acceleration coefficients), which are commonly used as input in previous works on acoustic word embeddings, but they are known to also encode speaker and environmental acoustics. Pre-trained neural acoustic models, such as Wav2Vec 2.0 and HuBERT are known to be more robust against speaker and environmental acoustics. We use the s3prl package [28] to extract these features from whole utterances. We then segment the feature sequences according to the corresponding word boundaries obtained by force-alignment<sup>6</sup>.

<sup>6</sup>We use the alignments from: <https://github.com/CoerentinJ/librispeech-alignments>

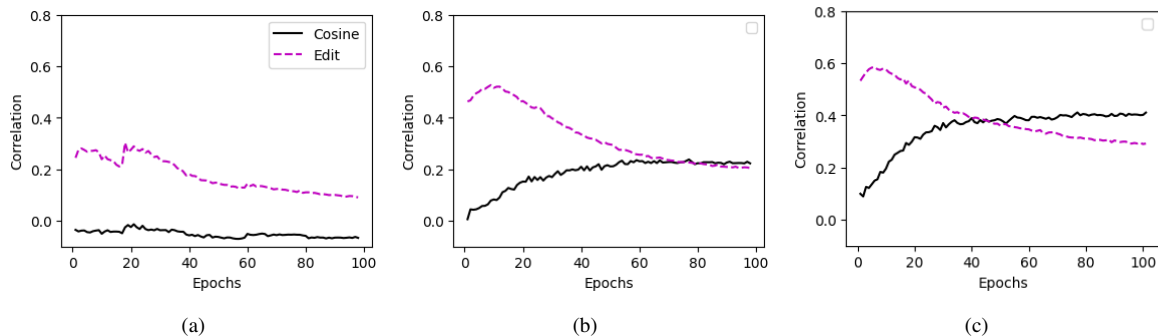


Figure 3: Pearson correlation with target cosine distances and edit distances across the 100 training epochs of the end-to-end model with  $s=4$ . The input features are obtained by clustering acoustic frames with  $K=100$  using (a) MFCC, (b) Wav2Vec2, and (c) HuBERT features.

| Model    | Examples of nearest neighbors   |
|----------|---|
| MFCC     | father, further, however, color, friend, thought, rather, tender, pleasant, beginning, clever         |
|          | eyes, honest, arm, ominous, irony, eyed, always, life, often, offers, onward                          |
|          | exclaimed, excellent, agreed, observed, assured, innocent, escaped, discovered, horrid, heart, closed |
| Wav2Vec2 | father, sister, mother, brother, queer, master, daughter, mary, prison, grandmother, mistress         |
|          | eyes, arms, arm, ice, face, hands, hangs, tears, boughs, hair, nose                                   |
|          | exclaimed, answered, madam, thank, interrupted, asked, dear, remarked, pray, guess, replied           |
| HuBERT   | father, mother, fear, daughter, brother, sister, wife, villefort, friend, husband, grandmother        |
|          | eyes, arms, hands, eyelids, shoulders, tears, hairs, knees, face, ice, weariness                      |
|          | exclaimed, yes, asked, responded, replied, cried, murmured, ah, retorted, quoth, answered             |

Table 1: Examples of nearest neighbors using the acoustic-based embeddings.

## 4.2. Discrete Units

Using the continuous acoustic feature vectors directly as input to the end-to-end model, the resulting embeddings showed almost zero correlation with the target semantic features. It seems crucial to discretize the input in order to minimize the effect of acoustic correlations. Ideally, the resulting discrete units should consistently correspond to target phonemes, and different occurrences of the same words should have minimum variation. To that end, we clustered the acoustic frames using the K-Means algorithm and used the cluster IDs as discrete acoustic units. Following the settings in HuBERT [25], we experimented with  $K=\{100,500\}$ . We then replaced all contiguous occurrences of the same cluster ID with a single unit to reduce the length of the sequence and minimize redundancy. Furthermore, following the observation in [24], we normalized all features before clustering such that each feature dimension has zero mean and unit variance

## 4.3. Results

We trained the end-to-end model using the discrete units described above, and evaluated the quality of the embeddings by mean-pooling the embeddings of all occurrences of each word in the vocabulary, then calculating the Pearson correlation coefficient between the pair-wise similarities of these embeddings and the target model, in addition to the correlation with edit distance, as described in section 3.2. Figure 3 shows the learning curves for the model with the largest scale ( $s=4$ ) for 100 epochs of training. Using MFCC-based embeddings, the cosine correlation remained around 0 throughout training, while edit distance correlation decreased over time. Wav2Vec2 and Hubert based models show an increasing trend of semantic correlation and a decreasing edit distance correlation over time, which is the desired trend for semantic embedding; the highest correlation with the target cosine distances was 0.41, achieved by the

HuBERT-based model, paired with 0.31 correlation with edit distance. Table 1 shows examples of nearest neighbors by the end of training. While both Wav2Vec2 and HuBERT-based models show some semantic qualities, the HuBERT-based embeddings appear more consistent. The embeddings show both semantic (e.g. *eyes* and *tears*) and phonetic qualities (e.g. *eyes* and *ice*).

## 5. Discussion & Conclusions

In this work, we conducted several experiments to understand the mechanism of learning word distributional semantics from low-level continuous speech features. Our findings underscore the importance of the following factors: (1) the type of acoustic features used as input, (2) the scale and depth of the embedding network, and (3) using end-to-end training paradigm. By combining self-supervised pre-trained acoustic features, discrete frame-level units obtained by clustering these features, and deep embedding networks trained end-to-end with a skipgram-like objective, we obtained word embeddings that exhibit semantic relatedness that goes beyond phonetic similarity. Previous works in this vein failed to obtain such results for several factors: they mainly relied on MFCC features, used shallow networks, and/or failed to account for spurious acoustic correlations in the input units. Our experiments show that each of these factors individually diminish the ability of the model to break free from the inherent acoustic correlations of spoken words and learn corpus-level semantic features. In spite of the positive results we obtained, we acknowledge several difficulties that would be encountered in a more realistic unsupervised setting. In particular, unsupervised word segmentation remains a rather challenging task, so alternative subword segments should be considered and examined for a truly self-supervised semantic embedding model.

## 6. References

- [1] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 2013, pp. 746–751.
- [2] G. Finley, S. Farmer, and S. Pakhomov, "What analogies reveal about word vectors and their compositionality," in *Proceedings of the 6th joint conference on lexical and computational semantics (\*SEM 2017)*, 2017, pp. 1–11.
- [3] H. Aldarmaki, M. Mohan, and M. Diab, "Unsupervised word mapping using structural similarities in monolingual embeddings," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 185–196, 2018.
- [4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.
- [5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
- [6] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," 2020.
- [7] K. Levin, K. Henry, A. Jansen, and K. Livescu, "Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings," in *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE, 2013, pp. 410–415.
- [8] Y.-A. Chung, C.-C. Wu, C.-H. Shen, H.-Y. Lee, and L.-S. Lee, "Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," 2016.
- [9] S. Ghannay, Y. Esteve, and N. Camelin, "Acoustic word embeddings for asr error detection." 2016.
- [10] Y. Yuan, C.-C. Leung, L. Xie, H. Chen, B. Ma, and H. Li, "Learning acoustic word embeddings with temporal context for query-by-example speech search," *Proc. Interspeech 2018*, pp. 97–101, 2018.
- [11] P. Peng, H. Kamper, and K. Livescu, "A correspondence variational autoencoder for unsupervised acoustic word embeddings," *Advances in neural information processing systems*, 2020.
- [12] H. Kamper, Y. Matuskevych, and S. Goldwater, "Multilingual acoustic word embedding models for processing zero-resource languages," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6414–6418.
- [13] H. Aldarmaki, A. Ullah, S. Ram, and N. Zaki, "Unsupervised automatic speech recognition: A review," *Speech Communication*, 2022.
- [14] Y.-A. Chung and J. Glass, "Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech," *Proc. Interspeech 2018*, pp. 811–815, 2018.
- [15] G. Chen and Y. Cao, "A reality check and a practical baseline for semantic speech embedding," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [16] Y.-C. Chen, S.-F. Huang, C.-H. Shen, H. yi Lee, and L. shan Lee, "Phonetic-and-semantic embedding of spoken words with applications in spoken content retrieval," 2019.
- [17] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 238–247.
- [18] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the association for computational linguistics*, vol. 5, pp. 135–146, 2017.
- [19] P. Bojanowski, A. Joulin, and T. Mikolov, "Alternative structures for character-level rnn," *arXiv preprint arXiv:1511.06303*, 2015.
- [20] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 2227–2237. [Online]. Available: <https://aclanthology.org/N18-1202>
- [21] K. Bostrom and G. Durrett, "Byte pair encoding is suboptimal for language model pretraining," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 4617–4624.
- [22] A. Baevski, W.-N. Hsu, A. Conneau, and M. Auli, "Unsupervised speech recognition," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 826–27 839, 2021.
- [23] K. Lakhota, E. Kharitonov, W.-N. Hsu, Y. Adi, A. Polyak, B. Bolte, T.-A. Nguyen, J. Copet, A. Baevski, A. Mohamed *et al.*, "On generative spoken language modeling from raw audio," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1336–1354, 2021.
- [24] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi *et al.*, "Audiolm: a language modeling approach to audio generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- [25] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhota, 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, 2021.
- [26] 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, pp. 5206–5210.
- [27] R. Rehurek and P. Sojka, "Gensim–python framework for vector space modelling," *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, vol. 3, no. 2, 2011.
- [28] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhota, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. yi Lee, "SUPERB: Speech Processing Universal PERFORMANCE Benchmark," in *Proc. Interspeech 2021*, 2021, pp. 1194–1198.