TU Delft

Delft University of Technology

The effectiveness of unsupervised subword modeling with autoregressive and cross-lingual phone-aware networks

Feng, Siyuan; Scharenborg, Odette

DOI
10.1109/OJSP.2021.3076914

Publication date
2021

Document Version
Accepted author manuscript

Published in
IEEE Open Journal of Signal Processing

Citation (APA)
Feng, S., & Scharenborg, O. (2021). The effectiveness of unsupervised subword modeling with autoregressive and cross-lingual phone-aware networks. *IEEE Open Journal of Signal Processing*, *2*, 230 - 247. Article 9420327. https://doi.org/10.1109/OJSP.2021.3076914

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

The effectiveness of unsupervised subword modeling with autoregressive and cross-lingual phone-aware networks

Siyuan Feng and Odette Scharenborg, *Senior Member, IEEE*

This study addresses unsupervised subword modeling, i.e., learning acoustic feature representations that can distinguish between subword units of a language. We propose a two-stage learning framework that combines self-supervised learning and cross-lingual knowledge transfer. The framework consists of autoregressive predictive coding (APC) as the front-end and a cross-lingual deep neural network (DNN) as the back-end. Experiments on the ABX subword discriminability task conducted with the Libri-light and ZeroSpeech 2017 databases showed that our approach is competitive or superior to state-of-the-art studies. Comprehensive and systematic analyses at the phoneme- and articulatory feature (AF)-level showed that our approach was better at capturing diphthong than monophthong vowel information, while also differences in the amount of information captured for different types of consonants were observed. Moreover, a positive correlation was found between the effectiveness of the back-end in capturing a phoneme's information and the quality of the cross-lingual phone labels assigned to the phoneme. The AF-level analysis together with t-SNE visualization results showed that the proposed approach is better than MFCC and APC features in capturing manner and place of articulation information, vowel height, and backness information. Taken together, the analyses showed that the two stages in our approach are both effective in capturing phoneme and AF information. Nevertheless, monophthong vowel information is less well captured than consonant information, which suggests that future research should focus on improving capturing monophthong vowel information.

***Index Terms*—Unsupervised subword modeling, zero-resource, cross-lingual modeling, phoneme analysis, articulatory feature analysis.**

I. INTRODUCTION

There are around $7,000$ spoken languages in the world [1]. For most of them, the amount of transcribed speech data resources is very limited, or even non-existent [2]. Many of these low-resource languages, such as ethnic minority languages in China and languages in Africa, may have never been formally studied. In addition to the lack of enough transcribed speech data, linguistic knowledge about such languages is incomplete, or may even be entirely lacking. Conventional supervised acoustic modeling [3], [4] can therefore not be applied directly. This leads to the current situation that high-performance ASR systems are only available for a small number of major languages, e.g., English, Mandarin, French. To facilitate ASR technology for low-resource languages, investigation of unsupervised acoustic modeling (UAM) methods is necessary, which aims to find and model a set of basic speech units that represents all the sounds in the language of interest, i.e., the low-resource, target language.

Recently, there has been a growing research interest in UAM [5]–[10]. A strict assumption of UAM is that for the target language only raw speech data is available, while the transcriptions, phoneme inventory (and its size) and pronunciation lexicon are unknown. This is known as the *zero-resource* assumption [11]. There are two main research strands in UAM. The first strand formulates the problem as discovering a finite set of phoneme-like speech units [5], [6], [12], [13]. This is often referred to as *acoustic unit/model discovery* (AUD) [5], [8]. The second strand formulates the problem as learning acoustic feature representations that can distinguish subword (phoneme) units of the target language, and is robust to linguistically-irrelevant factors, such as speaker [14]–[16]. This is often referred to as *unsupervised subword modeling* [11], [16], [17]. In essence, the second strand is focused on learning an intermediate representation towards the ultimate goal of UAM, while the first strand aims directly at the ultimate goal. These two strands are closely connected and can benefit from each other; for instance, a good subword-discriminative feature representation has been shown beneficial to AUD [18], [19], while conversely, discovered speech units with good consistency with true phonemes are helpful to learning subword-discriminative acoustic feature representations [14], [16].

This study addresses unsupervised subword modeling in UAM. Learning subword-discriminative feature representations in the zero-resource scenario has been shown to be a non-trivial task [11], [17]. The major difficulty is the separation of linguistic information (e.g., phoneme information) from non-linguistic information (e.g., speaker information). For instance, a speech sound such as [æ][1] produced by different speakers might be mistakenly modeled as different speech units [20].

There are many interesting attempts to unsupervised subword modeling [7], [9], [14]–[16], [21]. One typical research direction is to leverage purely unsupervised learning techniques. One method is the clustering of speech sounds that have acoustically similar patterns and that potentially correspond to the same subword units [7], [22], which results in phoneme-like pseudo transcriptions that can be used to facilitate subword-discriminative feature learning [7], [14]. Unsupervised and self-supervised representation learning algorithms are applied to learn, without using external supervision, speech features that retain the linguistic content in the original data while ignoring linguistically-irrelevant information, particularly speaker variation [15], [23]–[26].

A second research direction to unsupervised subword mod-

The authors are with Multimedia Computing Group, Delft University of Technology, Delft, the Netherlands (e-mail: S.Feng@tudelft.nl; O.E.Scharenborg@tudelft.nl).

Manuscript received.

[1]International Phonetic Alphabet (IPA) symbol.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

eling is to exploit cross-lingual knowledge [27], [28]. Speech and text resources from out-of-domain (OOD) resource-rich languages have been shown beneficial to modeling subword units of in-domain low-resource languages. For instance, [27], [28] used an OOD AM to extract cross-lingual bottleneck features (BNFs), and [28] also used an OOD ASR to generate cross-lingual phone labels.

This study adopts a two-stage learning framework which combines both research directions within the area of unsupervised subword modeling (the second research strand in UAM). At the first stage, the front-end, a self-supervised representation learning model named autoregressive predictive coding (APC) [29] is trained. APC preserves phonetic (subword) and speaker information from the original speech signal, but makes the two information types more separable [29].

At the second stage, the back-end, a cross-lingual, OOD DNN model with a bottleneck layer (DNN-BNF) is trained using the APC pretrained features as the input features to create the missing (due to the zero-resource assumption) frame labels. This system framework was proposed in our recent study [30], and showed state-of-the-art performances on the subword discriminability task on two databases in UAM: ZeroSpeech 2017 [17] and Libri-light [21].

In this work, we expand and extend the work in [30]. Specifically, we (1) compare the proposed approach to a supervised topline system that is trained on transcribed data of the target language; (2) compare the proposed approach with another cross-lingual knowledge transfer method [27]; (3) investigate the potential of our approach in relation to the amount of unlabeled training material by varying the data between 500 hours (as used in [30]) and $5,000$ hours, and compare the models' performance to the topline model. Throughout our experiments, English is chosen as the target low-resource language. Its phoneme inventory and transcriptions are assumed unavailable during system development. Dutch and Mandarin are chosen as the two OOD languages for which phoneme inventories and transcriptions are available.

Unsupervised subword modeling is typically evaluated using overall performance measures, such as ABX [11], [17], purity [6], normalized mutual information (NMI) [13]. These metrics, however, do not provide insights on the approaches' ability of modeling individual phonemes or phoneme categories. As the ultimate goal beyond unsupervised subword modeling is to discover basic speech units that have a good consistency with the true phonemes of the target language, we, to the best of our knowledge for the first time in the literature, additionally present detailed analyses that explore the question of the effectiveness of the proposed approach to capturing phoneme and articulatory feature (AF) information of the target language. The analyses are based on the standard ABX error rate evaluation [11], which we adapted for this work (see Section IV), and consist of two parts, i.e., an analysis at the phoneme level and at the AF level. The analyses are aimed at investigating what phoneme and AF information is (not) captured by the learned subword-discriminative feature representation, which can be used to guide future research to improve unsupervised subword modeling as well as AUD. Moreover, we correlate the phoneme-level ABX error rates and the quality of the cross-lingual phone labels which are used to train our back-end DNN-BNF model in order to study why the proposed approach performs differently in capturing different target phonemes' information, and how the performance is affected by the quality of cross-lingual phone labels.

The remainder of this paper is organized as follows. Section II provides a review of related works on the unsupervised subword modeling task. In Section III, we provide a detailed description of the proposed approach to unsupervised subword modeling, and introduce comparative approaches to compare against our approach. Section IV describes the methodology used for the phoneme-level and AF-level analyses. Section V introduces the experimental design of this study, while Section VI reports the results. Section VII describes the setup for conducting the phoneme- and AF-level analyses, and discusses the results of the analyses. Finally, Section VIII draws the conclusions.

II. Related works

A. Unsupervised learning techniques

Clustering algorithms and self-supervised learning techniques are widely applied in zero-resource speech modeling. For instance, the clustering approach using the Dirichlet process Gaussian mixture model (DPGMM) [31] has shown to outperform all other competitors [7] in the Zero Resource Speech Challenge (ZeroSpeech) 2015 [11]. Follow-up studies focused on improving speaker invariance of the input features to DPGMM [14] (best-performing in ZeroSpeech 2017 [17]), [28], [32], exploring multilingual DPGMM [33], [34], and alleviating the "over-fragment" nature, i.e., DPGMM's shortcoming in producing an excessive number of fine-grained clusters [16], [35]. K-means and HMMs were also investigated for frame clustering [36], [37].

In self-supervised learning for zero-resource speech modeling [15], [23], [24], [29], [38], [39], targets that a model is trained to predict are computed from the data itself [40]. A typical self-supervised representation learning model is the vector-quantized variational autoencoder (VQ-VAE) [15], which achieved a fairly good performance in ZeroSpeech 2017 [41] and 2019 [9], and has become more widely adopted [42]–[44] in the latest ZeroSpeech 2020 challenge [45]. Other self-supervised learning algorithms such as factorized hierarchical VAE (FHVAE) [46], contrastive predictive coding (CPC) [23] and APC [29] were also extensively investigated in unsupervised subword modeling [30], [42], [47], [48] as well as in a relevant zero-resource word discrimination task [49].

B. Cross-lingual knowledge transfer

Transcribed speech data for OOD languages [50], [51] can be exploited in various ways to boost zero-resource subword modeling for low-resource languages. In [27], [28], a DNN AM trained with OOD languages was used to extract cross-lingual phone posteriorgrams [27] or cross-lingual BNFs [27], [28] of a target low-resource language as the learned feature representation. In [16], [28], an OOD ASR system was used to generate phone labels for the target language speech. These

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

cross-lingual labels served as supervision for training a DNN-BNF model. This idea is applied in our present study.

There is evidence that the above-mentioned cross-lingual phone labels are complementary to labels obtained with unsupervised learning [16], [28]. Specifically, cross-lingual phone labels and DPGMM clustering labels for the target language's speech data were jointly used to train a DNN-BNF. The resulting BNF representation performed better than that extracted by a DNN-BNF trained using either type of the labels. The work in [52] adopted another way of combining unsupervised and cross-lingual learning strategies; they used cross-lingual BNFs as input to train a correspondence autoencoder (cAE). In our present study, the combination of unsupervised learning techniques and cross-lingual knowledge transfer is done in a different way by adopting a self-supervised learning front-end followed by a cross-lingual phone-aware DNN-BNF back-end.

C. Analysis of unsupervisedly discovered speech units

Few analyses on the effectiveness of subword modeling at the phoneme level or of the linguistic relevance of the speech units learned using AUD exist [53]–[55]. In [53], an analysis on the consistency between individual discovered speech units from an unknown language and the language's true phoneme inventories showed that while the learned speech units had a good coverage of the phoneme inventories, some phonemes with rapidly changing acoustics (e.g., diphthongs) could not be well discovered. In contrast to [53], our analysis study is based on the subword-discriminative feature representation instead of based on a discrete set of learned units. Moreover, our study also carries out performance analysis at the AF level in addition to the phoneme level. A recent study [55] carried out an analysis of the subword-discriminative feature representations, similar to our present study, but with the purpose of comparing the unsupervised learning approaches to infant phonetic perception. A major difference between [55] and ours is that [55] selected only three phone contrasts from a target language, whereas our study conducts analysis on the complete phoneme inventories. Finally, an analysis by [54] showed that a visually-grounded model (not requiring transcribed data) learns subword units that carry AF information, such as vowel backness or stop voicing.

III. Proposed Approach to Unsupervised Subword Modeling

The general framework of the proposed approach to unsupervised subword modeling is illustrated in Fig. 1. In the training phase, for a target language with a certain amount of untranscribed speech training data (marked in pink in Fig. 1), an APC model as the front-end is trained with target untranscribed speech in a self-supervised manner, and used to extract pretrained features for the target speech. Next, at the back-end, an OOD ASR system assigns one phone label to every frame of the target language's speech. This OOD ASR is trained on a language different from the target language (marked in blue). With APC pretrained features for the target untranscribed speech as input features and cross-lingual phone labels as their corresponding target labels, a DNN-BNF model

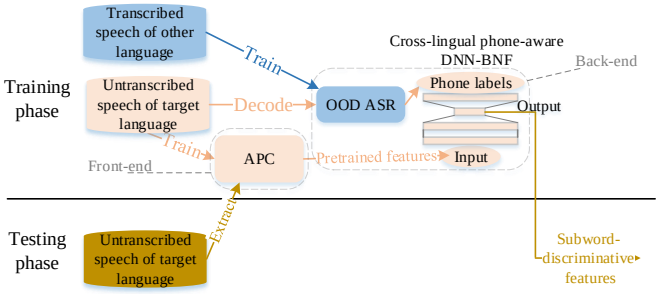

Fig. 1: General framework of the proposed approach. The two colors in the training phase represent the data sets used to train each model.

is trained, in order to learn subword-discriminative features for the target speech. In the testing phase, the DNN-BNF model is used to extract BNF representations for test data of the target language (marked in yellow) as the subword-discriminative feature representation of the target language's speech.

This study compares the proposed approach with other, state-of-the-art approaches (see Section III-C for details). The front-end APC pretraining method is compared with another self-supervised learning method, i.e., FHVAE [24]. Their comparison is made at front-end level. The whole pipeline of the proposed approach is compared with a system consisting of only the back-end DNN-BNF model (without an APC front-end), a CPC approach [23] and a transfer learning BNF system based on a cross-lingual AM [27]. Moreover, two different languages will be used to train two different OOD ASR systems for comparison.

A. APC pretraining

In previous studies, feature representation learning techniques were often adopted in order to suppress speaker variation while retaining linguistic information [14], [16], [28]. In contrast, APC adopted in the proposed approach is aimed at learning a frame-level feature representation that retains both phonetic and speaker information from the speech signal, while making the phonetic information and speaker information more separable for downstream phone or speaker classification tasks, comparing to when spectral features are used as frame-level representations [29]. In such a way, the learned representation is considered to be less at risk of losing phonetic information compared to that learned by methods in [14], [16], [28].

Let $\{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_T}\}$ denote d-dimensional frame-wise features for a set of untranscribed speech data for APC training, where T is the total number of speech frames. At each time step t, the encoder of APC, denoted as $\mathrm{Enc}(\cdot)$, reads as input a feature vector $\boldsymbol{x_t}$, and generates a d-dimensional output feature vector $\hat{\boldsymbol{x}}_{\boldsymbol{t}}$ based on previous input features $\boldsymbol{x_{1:t}} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_t}\}$,

$$\hat{\boldsymbol{x}}_{\boldsymbol{t}} = \mathrm{Enc}(\boldsymbol{x_{1:t}}). \tag{1}$$

The goal of APC is to let $\hat{\boldsymbol{x}}_{\boldsymbol{t}}$ be as close to $\boldsymbol{x_{t+n}}$ as possible, where n is a pre-defined constant non-negative integer, known

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

as the *prediction step*. The loss function for APC training is defined as,

$$\text{Loss} = \sum_{t=1}^{T-n} |\hat{\boldsymbol{x}}_{\boldsymbol{t}} - \boldsymbol{x}_{\boldsymbol{t+n}}| \,. \tag{2}$$

With this loss function, the APC encoder learns information that is relevant to predicting future frames. Intuitively, a large n encourages the APC model to encode global characteristics in the original speech signal, while a small n lets the model focus mainly on local smoothness in speech.

The APC encoder is implemented as a long short-term memory (LSTM) RNN structure [56], as was done in [29]. Let L denote the number of LSTM layers, Equation (1) is formulated as,

$$\boldsymbol{h_0} = \boldsymbol{x_{1:t}}, \tag{3}$$

$$\boldsymbol{h_l} = \text{LSTM}^l(\boldsymbol{h_{l-1}}), l \in \{1, 2, \ldots, L\}, \tag{4}$$

$$\hat{\boldsymbol{x}}_{\boldsymbol{t}} = \boldsymbol{W}\boldsymbol{h_L}, \tag{5}$$

where $\boldsymbol{W}$ is a trainable projection matrix. The equations that form LSTM$(\cdot)$ can be found in [56].

After APC training, $\boldsymbol{h^L} = \{\boldsymbol{h^L_{1:T}}\}$ is extracted as the learned acoustic representation of the original speech, and is henceforth referred to as the *APC feature*. In principle, $\boldsymbol{h^l}$ of an arbitrary layer could be extracted as the learned representation. We follow [29] in using the representation from the top layer as they showed that this gave the optimal performance on phone classification tasks.

B. Cross-lingual phone-aware DNN-BNF

As shown in Fig. 1, the back-end of the proposed approach is a DNN model with a low-dimensional intermediate hidden layer, also known as the bottleneck layer. To train such a DNN-BNF model, cross-lingual phone labels are obtained beforehand. Specifically, an OOD ASR system is applied to decode speech utterances of the target language's training data. The decoding results, i.e. hypothesized transcripts are generated for speech utterances. By applying forced alignment to the hypothesized transcripts (i.e. OOD phone sequences as labels) using the AM of the OOD ASR system, each frame of the training utterance is assigned a phone symbol label modeled by the OOD ASR. In this work the OOD ASR is realized as a hybrid DNN-HMM architecture. As a result, the cross-lingual phone labels are triphone HMM states[2] modeled by the hybrid DNN-HMM. In principle, ASR systems with other architectures could also be applied to generate decoding-based labels for target unlabeled speech, such as connectionist temporal classification (CTC) [57] and attention-based models [58].

The DNN-BNF model is then trained with the speech data of the target language, using the pretrained APC features as input features and the cross-lingual labels as target labels. The training is done by minimizing cross-lingual phone prediction error by using the lattice-free maximum mutual information (LF-MMI) criterion [59]. After training, the DNN-BNF model is used to extract the BNF representations of the test data as the desired subword-discriminative feature representation. The BNF representation is essentially the output of the bottleneck layer, as shown in Fig. 1.

[2]In the literature, they are also referred to as *senones*.

C. Comparative approaches

1) FHVAE

FHVAE is a self-supervised representation learning model which does not need transcribed data in model training [24]. It disentangles phonetic and speaker information by capturing the two types of information with latent sequence variables $\boldsymbol{z_1}$ and latent segment variables $\boldsymbol{z_2}$ respectively. The $\boldsymbol{z_1}$ representation from a well-trained FHVAE is extracted as the desired speaker-invariant phonetic representation for unsupervised subword modeling. The FHVAE model was applied in [10] and achieved good performance in the ZeroSpeech 2019 Challenge [60], which is why we compare the APC model against FHVAE in this study. Details of the FHVAE model description is provided in supplementary material (see Section S1-A).

2) Cross-lingual AM based BNF

Learning subword-discriminative feature representations by exploiting cross-lingual knowledge transfer could be realized in a different way than the back-end of our proposed approach discussed in Section III-B. In our back-end, the DNN-BNF model trained using unlabeled speech data of the target language and cross-lingual phone labels is used to extract the BNF representation. Alternatively, a DNN-BNF model trained using labeled speech data of an OOD language can be leveraged to extract the BNF representation for the target language [27], [61]. In essence, the method in [27], [61] leverages a well-trained cross-lingual AM for transfer learning. Thus this method is denoted as the *cross-lingual AM based BNF*, to be distinguished from our back-end cross-lingual phone labeling based method. The cross-lingual AM based BNF method does not rely on audio data of the target language for training, which makes it fast in system development. Moreover, this method is feasible in a stricter zero-resource scenario where unlabeled audio data of the target language is unavailable for training. We will compare our entire approach pipeline against the cross-lingual AM based BNF method.

3) CPC

CPC is a self-supervised representation learning model which does not require transcribed data in model training [23]. By using a contrastive loss, the model is trained to distinguish future speech frames from a set of negative examples. The CPC model is able to capture phonetic information in speech while suppressing noise and speaker variation [21], and achieves good performance in unsupervised subword modeling [21], [48], [62]. This is why we compare our approach to CPC.

IV. Methods to analyze the effectiveness of the proposed approach

This section describes our phoneme-level and articulatory feature-level methods to analyze the effectiveness of the proposed approach to unsupervised subword modeling. Both methods are based on the ABX test [63]. Section IV-A briefly introduces the ABX test and its application as an overall

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

performance measure in unsupervised subword modeling. Sections IV-B and IV-C discuss the proposed phoneme- and AF-level analyses methods respectively.

A. ABX subword discriminability task

The ABX test was first proposed in [63] as as a human auditory test. A, B and X are three audio signals. A listener is presented with A, B and X, and is asked to make judgement about whether X is more similar to A or to B.

The ABX test has recently been adopted as one of the standard evaluation metrics in unsupervised subword modeling [11], [17], [60]. Specifically, A, B and X are feature representations of three speech sounds. A and B contain triphone sequences that differ only in the central phone, e.g. "a-p-i" versus "a-t-i". X contains either "a-p-i" or "a-t-i". The ABX error rate of a pair of triphone sequences x and y is defined as,

$$\epsilon(x, y) = \frac{1}{2}[\eta(x \to y) + \eta(y \to x)], \tag{6}$$

where

$$\eta(x \to y) = \frac{1}{|S(x)|(|S(x)|-1)|S(y)|} \sum_{A \in S(x)} \sum_{B \in S(y)} \sum_{X \in S(x) \setminus \{A\}} (\mathbb{1}_{d(A,X) > d(B,X)} + \frac{1}{2}\mathbb{1}_{d(A,X)=d(B,X)}). \tag{7}$$

Here $\mathbb{1}$ is the indicator function, $S(x)$ and $S(y)$ denote two sets of speech sounds containing x and y, $d(\cdot, \cdot)$ denotes dynamic time warping (DTW) based dissimilarity between two speech sounds. Frame-level dissimilarity measure for DTW scoring can be the cosine distance, Kullback–Leibler (KL) divergence, etc. Here, cosine distance is used throughout this paper. By taking average of $\epsilon(x, y)$ over all possible triphone sequences x and y that share context phones and contrast the same central phone pair, the phone **pairwise ABX error rate** is calculated. By further taking the average of the phone pairwise ABX error rates over all possible pairs of phones, the **overall ABX error rate** is calculated.

B. Phoneme-level analysis

We define **phoneme-level ABX error rate** as follows. Let ω be a phoneme in phoneme inventory Ω of a target language. The phoneme-level ABX error rate of ω is then calculated as,

$$\xi(\omega) = \frac{1}{|\Omega| - 1} \sum_{\omega' \in \Omega \setminus \{\omega\}} \epsilon(\omega, \omega'), \tag{8}$$

where $\epsilon(\omega, \omega')$ is the pairwise ABX error rate calculated as mentioned in IV-A. $\xi(\cdot)$ is used as the measure of a subword-discriminative feature representation's effectiveness towards each individual phoneme in a target language.

The present study also investigates the correlation between phoneme-level ABX error rate improvement achieved by the back-end of the proposed approach and the quality of the cross-lingual phone labels used in the back-end. Let $\Psi = \{\psi_1, \ldots, \psi_M\}$ denote M cross-lingual phones modeled by an OOD ASR system, and $\{l_t \in \Psi | t = 1, 2, \ldots, T\}$ denote cross-lingual phone labels for a certain amount of target speech data generated by the OOD ASR. Let $\{g_t \in \Omega | t = 1, 2, \ldots, T\}$ denote the true phoneme labels for the same speech data, where $\Omega = \{\omega_1, \ldots, \omega_N\}$ is the set of N phonemes in the target language. An N-by-M confusion matrix $\boldsymbol{E}$ can be constructed, with its element e_{ij} defined as,

$$e_{ij} = \frac{\sum_{t=1}^{T} \mathbb{1}(g_t = \omega_i, l_t = \psi_j)}{\sum_{t=1}^{T} \mathbb{1}(g_t = \omega_i)}. \tag{9}$$

Conceptually, $e_{ij} \in [0, 1]$ represents the percentage of target speech frames of true phoneme ω_i that are labeled as cross-lingual phone ψ_j. As seen from Equation (9), elements in every row of $\boldsymbol{E}$ sum up to 1. Ideally the row vector $[e_{i,1}, e_{i,2}, \ldots, e_{i,M}]$ is a (quasi) one-hot vector, in which case there exists a cross-lingual phone ψ_{j*} so that e_{ij*} is close to 1. A large e_{ij*} indicates high consistency between ω_i and ψ_{j*}, and is desired in the cross-lingual DNN-BNF model training, as the DNN-BNF model can be trained to map acoustic features representing different speech realizations of ω_i to very close representations in the cross-lingual phonetic space. It is worth noting that the consistency being discussed here measures to which extent speech frames of ω_i get the same cross-lingual phone labels irrespective of the labels' symbol.

Mathematically, $j* = \text{argmax}_{j=1}^{M} e_{ij}$, and the co-occurrence probability between ω_i and ψ_{j*}, denoted as $p_{co}(\omega_i)$, is defined as,

$$p_{co}(\omega_i) = \max_{j=1}^{M} e_{ij}. \tag{10}$$

In this study, $p_{co}(\omega_i)$ is utilized as a measure of the cross-lingual phone label quality of ω_i.

C. AF-level analysis

An AF-level analysis is carried out using the proposed *ABX AF discriminability task* in order to evaluate how well a speech feature representation is capable of distinguishing one AF attribute from another. Analogous to the ABX subword discriminability task as introduced in Section IV-A, let A and B contain triphone sequences that differ only in the central phone. Here the central phones of A and B are set to belong to different **AF attributes**. For instance, take manner of articulation (MoA): A could contain a *stop* in the center (e.g., A could be /a p i/), while B contains a *fricative* (e.g., /a f i/). X contains a triphone sequence with its central phone being either a stop or a fricative, but not necessarily "p" or "f". The context phones of X are "/a/" and "/i/". Possible realizations of X could be "a g i", "a z i", etc. Using Equations (6) and (7), followed by taking the average over all possible triphone sequences that share the context phones and contrast the stop and fricative attributes in the central phones, the **pairwise ABX AF error rate** between stops and fricatives is calculated. The **attribute-level ABX AF error rate** (e.g. the stop attribute) can then be calculated by taking the average of all pairwise ABX AF error rates involving that AF attribute.

The present study carries out AF-level analysis on two AFs for consonants, i.e., MoA and place of articulation (PoA), and on two AFs for vowels, i.e., tongue height and tongue backness. The mappings from an English phoneme to its

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

TABLE I: MoA (rows) and PoA (columns) for each English consonant. PoA abbreviations from left to right: Bilabial, Labiodental, Dental, Alveolar, Postalveolar, Palatal, Velar, Glottal. MoA abbreviations from top to bottom: Affricate, Approximant, Fricative, Stop, Nasal.

Consonants	BI	LA	DA	AL	PO	PA	VE	GL
Af					CH, JH			
Ap	W			L	R	Y		
Fr		F, V	TH, DH	S, Z	SH, ZH			HH
St	P, B			T, D			K, G	
Na	M			N			NG	

TABLE II: Tongue height (rows) and backness (columns) for each English monophthong vowel. Diphthongs are excluded in this study.

Vowels	Front	Central	Back
Close	IY, IH		UW, UH
Mid	EH	ER, AH	AO
Open	AE	AA	

TABLE III: Details of two training sets in Libri-light.

	#utterances	#speakers	#hours
unlab-600	35,229	489	526
unlab-6K	362,816	1,742	5,273

AF attributes are shown in Tables I (consonants) and II (vowels). In these two tables, each phoneme is represented as its ARPABET symbol [64]. For the analyses of tongue height and backness, all diphthongs are excluded, because it does not have a stable tongue height or backness attribute but rather during its production the articulators move from one position to another.

V. Experimental setup

A. Databases and evaluation metric

Training data for APC and DNN-BNF models are taken from Libri-light [21], a recently developed and freely-available English database to support unsupervised subword modeling. The *unlab-600* and *unlab-6K* sets from Libri-light are adopted. Details of the two training sets are listed in Table III.

Dutch and Mandarin training data for the development of two OOD ASR systems are the Corpus Gesproken Nederlands (CGN) [65] and Aidatatang_200zh (ADT) [51] respectively. The training-test partition of CGN follows those in [66]. Its training data contains 483 hours of speech covering conversational and read speech and broadcast news. ADT is a read speech corpus. Its training data consists of 140 hours of speech.

Evaluation data are taken from Libri-light and ZeroSpeech 2017 Track 1 [17]. The Libri-light evaluation data consists of 4 sets: *dev-clean, dev-other, test-clean, test-other*. Speech in dev-clean and test-clean have higher recording quality and accents closer to US English than that in dev-other and test-other.

The ZeroSpeech 2017 evaluation data consists of three languages, i.e., English, French and Mandarin [17]. In this study, the English evaluation data is chosen. The data are organized into subsets of differing lengths (1s, 10s and 120s).

The learned BNF representations, as well as APC pretrained feature representations are evaluated in terms of the overall ABX error rate (see Section IV-A), for *within-speaker* (X and A/B are spoken by the same speaker) and *across-speaker* (X and A/B are spoken by different speakers) separately. The across-speaker ABX task is more challenging, and is particularly focused on measuring the robustness of learned feature representations towards speaker variation. The within-speaker task serves as a benchmark for the across-speaker task by showing to what extent the ABX error in the across-speaker performance is caused by speaker variation.

B. Baselines and toplines

The official baseline systems of Libri-light and ZeroSpeech 2017 [17], [21] both use raw MFCC features. The official topline of ZeroSpeech 2017 is a supervised system which uses English labeled data to train an ASR system, followed by generating phone posteriorgram as the learned feature representation [17].

There is no official topline system provided for the Libri-light database. In this work we created a supervised system and used it as the topline of Libri-light. This system uses 960 hours of English labeled data from Librispeech [50] to train a time-delay NN (TDNN) AM, from which the output of the top TDNN layer is used as the learned representation. Training of the TDNN AM is implemented based on the Kaldi Librispeech recipe[3] without modifying any parameters. The TDNN AM consists of five 650-dimensional hidden layers. The input to the TDNN consists of 40-dimensional high-resolution MFCC features (HR MFCCs) appended by 100-dimensional i-vectors. Frame labels needed to train the TDNN AM are obtained by forced-alignment with a GMM-HMM AM trained beforehand, also using Librispeech.

C. Front-end implementation

The front-end APC model is implemented as a multi-layer LSTM network. Residual connections are made between two consecutive layers. Each layer has 100 dimensions. In order to find the optimal number of LSTM layers and prediction step n, layer numbers ranging in $\{3, 4, 5, 6\}$, with n ranging in $\{1, 2, 3, 4, 5\}$[4] are tested when the model is trained with unlab-600. These results are reported in supplementary material (see Section S1-B). From Table S1 the optimal parameters of LSTM layer number and n can be determined as 5 and 5 respectively. The two parameters are then fixed and the APC model is trained with the unlab-6K set from scratch. The input features to APC are 13-dimension MFCCs with cepstral mean normalization (CMN). The APC models are trained using an open-source tool by [29] for a fixed 100 epochs throughout our experiments, with the Adam optimizer [67], an initial learning rate of 0.0001, and a batch size of 32. After training, output of the top LSTM layer is extracted as the APC feature representation.

[3] `s5/local/nnet3/run_tdnn.sh`.

[4] Increasing n to larger than 5 was found leading to rapid ABX error rate degradation in preliminary experiments.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

D. Back-end implementation

1) OOD ASR systems

Two OOD ASR systems were developed, i.e., a Dutch ASR and a Mandarin ASR. Both OOD ASR systems use a chain TDNN AM trained in Kaldi [68] and a tri-gram LM trained using SRILM toolkit [69]. The TDNN AM contains 7-layers, including a 40-dimension bottleneck layer at the 6-th layer. Three-way speed perturbation [70] is applied to the Dutch (CGN) and Mandarin (ADT) training data respectively, before they are used to train the TDNN AM. The model is trained based on the lattice-free maximum mutual information (LF-MMI) criterion [59]. For Dutch, the input features are 40-dimension HR MFCCs. For Mandarin, the input features consist of HR MFCCs appended by pitch features (MFCC+P) [71]. Frame labels for TDNN model training are obtained by forced alignment using a GMM-HMM AM trained beforehand. The numbers of modeled phones and triphone HMM states in the Dutch TDNN AM are 81 and $3,361$, respectively. The number of modeled phones in the Mandarin TDNN AM is 119, including tone-dependent phones that are used to describe the four tones plus a neutral tone in Mandarin. The number of triphone HMM states in the Mandarin TDNN AM is $3,536$.

The Dutch ASR obtained a word error rate (WER) of 8.98% on the CGN broadcast test set. (This WER could be improved upon by integrating an RNN LM. As Dutch ASR performance is not the focus in this work, an RNN LM is not applied). The Mandarin ASR obtained a character error rate (CER) of 6.37% on the ADT test set. The two ASR systems are used to generate cross-lingual phone labels for the speech frames in Libri-light unlab-600 and unlab-6K sets.

2) DNN-BNF model

Two DNN-BNF models are trained, one taking the Dutch phone labels as training labels and one taking the Mandarin phone labels as training labels.

The DNN-BNF model consists of 7 feed-forward layers. Each layer has 450 dimensions except a 40-dimensional bottleneck layer, which is located below the top layer. The DNN-BNF model is trained based on the LF-MMI criterion. The inputs to the DNN-BNF are the APC feature with its neighboring frames, ranging from -3 to $+3$. After training, 40-dimensional BNFs are extracted, and are evaluated in terms of overall ABX error rate.. The BNF representations are named as **A-BNF-Du** and **A-BNF-Ma** henceforth, where "-Du" and "-Ma" denote using the Dutch and Mandarin training labels respectively, and "A-" denotes APC features as input features.

For comparison, two other DNN-BNF models (one using the Dutch phone labels as training labels and one using the Mandarin phone labels) are trained using HR MFCCs as input features. Other training and model parameter settings are unchanged. After training, again the BNFs are extracted and evaluated. The BNF representations are henceforth named as**M-BNF-Du** and **M-BNF-Ma**, where "M-" stands for taking MFCC features as input features.

TABLE IV: Configurations of the BNF representations implemented in the experiments.

	M-BNF-Du	M-BNF-Ma	A-BNF-Du	A-BNF-Ma	C-BNF-Du	C-BNF-Ma
Method	cross-lingual phone labeling				cross-lingual AM	
Acoustic	Libri-light				CGN	ADT
Input	MFCC		APC		MFCC	MFCC+P
Label	Du. ASR	Ma. ASR	Du. ASR	Ma. ASR	CGN	ADT

E. Implementation of the comparative approaches

1) FHVAE

Model parameters of FHVAE are chosen by referring to [47]. The FHVAE encoder and decoder are implemented as 2-layer LSTM networks. Each LSTM layer has 256 dimensions. Latent segment variable $\boldsymbol{z_1}$ and latent sequence variable $\boldsymbol{z_2}$ are both 32 dimensional. The inputs to FHVAE are fixed-length speech segments, each of which consists of 20 frames. Frame-level input features are MFCC with CMN. The FHVAE models are trained using an open-source tool by [24], with the Adam optimizer [67]. The model is trained with unlab-600 in Libri-light. A 10% subset of the training data is randomly selected for cross-validation (CV). The training procedure terminates if FHVAE's lower bound on the CV set does not improve for 40 epochs. After training, $\boldsymbol{z_1}$ is extracted.

2) Cross-lingual AM based BNF

The cross-lingual AM based BNF representation is generated based on the TDNN AM of the OOD ASR systems described in Section V-D1. To that end, speech features of the English evaluation data are fed to the OOD (Dutch or Mandarin) TDNN AM till its bottleneck layer. In this way, two BNF representations are extracted, one from the Dutch TDNN AM, and one from the Mandarin TDNN AM. The two BNF representations are named **C-BNF-Du** and **C-BNF-Ma**, respectively, where "C-" stands for the comparative approach.

To give an explicit comparison between the cross-lingual AM based BNFs and BNFs from our proposed approach (in Section V-D2), Table IV lists their configurations. The row "Acoustic" denotes the acoustic training data, the rows "Input" and "Label" denote input feature representation and frame labels used to train the respective systems.

VI. Experimental results

A. Effectiveness of the front-end APC pretraining

First, the APC and FHVAE methods are compared at the front-end, i.e., without using the DNN-BNF back-end. Overall across-speaker and within-speaker ABX error rates ($\%$) of the APC features, the FHVAE features, and the official MFCC baseline [21] on the four Libri-light evaluation sets are listed in Table V separately and averaged over all evaluation sets (rightmost column). The APC models in this table are trained with unlab-600[5] and unlab-6K respectively, and FHVAE is trained with unlab-600.

The results show that when trained with unlab-600, APC outperforms both the FHVAE feature representations and the MFCC baseline. The lower across-speaker ABX error rate results for the APC features imply that they are more

[5]APC trained with unlab-600 was also published in our recent study [30].

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

TABLE V: Overall ABX error rates (%) of the APC features, FHVAE features, and the official MFCC baseline on Libri-light. Bold indicates the best result. Data within brackets indicates the training set for each APC and FHVAE model.

System	Across-speaker dev-clean	dev-other	test-clean	test-other	Avg.
APC (unlab-600) [30]	12.64	19.00	12.19	18.75	15.65
APC (unlab-6K)	10.79	16.87	10.33	16.79	**13.70**
MFCC baseline [21]	20.94	29.41	20.45	28.50	24.83
FHVAE (unlab-600)	18.41	25.66	17.79	25.65	21.88
Within-speaker					
APC (unlab-600) [30]	8.83	11.07	8.36	11.48	9.94
APC (unlab-6K)	7.49	9.99	7.05	10.11	**8.66**
MFCC baseline [21]	10.95	13.55	10.58	13.60	12.17
FHVAE (unlab-600)	10.30	13.15	10.21	13.16	11.71

speaker invariant than the FHVAE features, even though the APC model is not explicitly suppressing speaker variation as FHVAE is.

Table V shows for the APC model, when the amount of training data is increased ten-fold by training on unlab-6K, the APC feature representation further improves by relative ABX error rate reduction of 12.5% in the across-speaker condition and 12.9% in the within-speaker condition.

B. Effectiveness of the proposed approach

Next, we compare the overall ABX error rates (%) of the proposed approach, comparative approaches, and our created supervised topline on the Libri-light evaluation sets. Table VI shows the performances of the different approaches evaluated on the four Libri-light evaluation sets separately and averaged over all evaluations sets (right-most column), for the across-speaker condition (top half of of Table VI) and within-speaker condition (bottom half of Table VI). The systems CPC [21] and CPC+DA [62] adopt the CPC model, and CPC+DA additionally applies data augmentation. Since CPC+DA [62] used augmented audio data derived from Librispeech, the performance of this system is listed under "Training set NOT in Libri-light". The topline system, C-BNF-Du and C-BNF-Ma that are trained with Librispeech, CGN and ADT respectively (see Table IV) are also listed under "NOT in Libri-light". Several observations can be made from this table:

(1) The cross-lingual phone-aware DNN-BNF methods that use the APC features as input features (A-BNF-Du and A-BNF-Ma) consistently outperform systems with MFCC input features (M-BNF-Du and M-BNF-Ma) on all evaluation sets. This demonstrates the effectiveness of the front-end APC pretraining in our proposed two-stage system framework. This observation confirms our earlier findings in recent work [30], in which the training set for APC was unlab-600 (526 hours). The present study shows that when the amount of training material is scaled up to unlab-6K (5,273 hours), APC pretraining brings even greater relative ABX error rate reduction than when trained on unlab-600: the across- and within-speaker relative error rate reductions from M-BNF-Du to A-BNF-Du are 9.5% and 5.5%, respectively, when trained with unlab-6K, while they are 7.6% and 4.8% when trained with unlab-600. Similarly, the across- and within-speaker relative error rate reductions from M-BNF-Ma to A-BNF-Ma are 15.0% and 15.4%, respectively, when trained with unlab-6K, while 10.0% and 9.3% when trained with unlab-600.

TABLE VI: Overall ABX error rates (%) of the proposed cross-lingual phone-aware BNFs, comparative approaches, and supervised topline on Libri-light. Systems listed under "NOT in Libri-light" used various different datasets other than unlab-600 or unlab-6K.

System	Across-speaker dev-clean	dev-other	test-clean	test-other	Avg.
Training set: unlab-600					
M-BNF-Du	6.67	11.65	6.64	12.00	9.24
M-BNF-Ma	7.92	12.71	7.74	13.23	10.40
A-BNF-Du	**6.18**	**11.02**	**6.03**	**10.94**	**8.54**
A-BNF-Ma	7.00	11.80	6.84	11.81	9.36
APC	12.64	19.00	12.19	18.75	15.65
CPC [21]	9.58	14.67	9.00	15.10	12.09
Training set: unlab-6K					
M-BNF-Du	6.06	11.30	6.12	11.35	8.71
M-BNF-Ma	7.80	12.39	7.60	12.95	10.19
A-BNF-Du	**5.70**	**10.08**	**5.70**	**10.02**	**7.88**
A-BNF-Ma	6.38	11.02	6.23	11.02	8.66
APC	10.79	16.87	10.33	16.79	13.70
CPC [21]	8.48	13.39	8.05	13.81	10.93
*Training set **NOT** in Libri-light*					
Topline	5.30	9.58	5.47	9.64	7.50
CPC+DA [62]	6.62	10.60	5.90	10.95	8.52
C-BNF-Du	7.17	11.20	6.89	11.40	9.17
C-BNF-Ma	9.92	14.91	9.83	15.34	12.50
Within-speaker					
Training set: unlab-600					
M-BNF-Du	4.97	6.94	4.73	6.86	5.88
M-BNF-Ma	6.06	7.71	5.62	7.82	6.80
A-BNF-Du	**4.77**	**6.69**	**4.49**	**6.43**	**5.60**
A-BNF-Ma	5.25	7.14	5.21	7.09	6.17
APC	8.83	11.07	8.36	11.48	9.94
CPC [21]	7.36	9.39	6.90	9.59	8.31
Training set: unlab-6K					
M-BNF-Du	4.70	6.58	4.36	6.37	5.50
M-BNF-Ma	5.94	7.65	5.69	7.77	6.76
A-BNF-Du	**4.48**	**6.15**	**4.24**	**5.91**	**5.20**
A-BNF-Ma	5.03	6.77	4.65	6.42	5.72
APC	7.49	9.99	7.05	10.11	8.66
CPC [21]	6.51	8.42	6.22	8.55	7.43
*Training set **NOT** in Libri-light*					
Topline	4.36	6.16	4.22	5.87	5.15
CPC+DA [62]	4.66	5.81	4.46	6.56	5.37
C-BNF-Du	5.50	7.50	5.27	6.86	6.28
C-BNF-Ma	7.63	9.30	7.28	9.51	8.43

(2) The proposed A-BNF-Du system trained with unlab-600 is comparable to the state-of-the-art CPC+DA system [62][6]. When A-BNF-Du is trained with the larger, unlab-6K set, it outperforms CPC+DA. Both A-BNF-Du and A-BNF-Ma outperform CPC [21] trained on unlab-600 and unlab-6K. It should be noted that in contrast to our approach, the CPC and CPC+DA systems do not require transcribed data from OOD languages for training. The huge advancement of CPC+DA [62] over CPC [21] indicates the effectiveness of adopting data augmentation techniques in the concerned task. It would thus be interesting to investigate the efficacy of integrating data augmentation and CPC with our current system framework. We leave it for future study.

(3) The best performance achieved by our proposed approach (A-BNF-Du) trained on unlab-6K is only slightly inferior to the supervised topline system (0.38% across-speaker and 0.05% within-speaker absolutely). This is an encouraging finding, as it indicates that on the task of discriminating between a pair of subword units of an unknown language, with

[6]A more strict comparison between A-BNF-Du and CPC+DA should be made under the identical training material setting, however performance of CPC+DA trained with Libri-light datasets was not reported in [65].

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

a sufficient amount of unlabeled data, our approach which does not require any linguistic knowledge of the target language, could perform on par with a supervised AM using transcribed training data of that language.

(4) In both the cross-lingual phone labeling method and the cross-lingual AM based BNF method, using Dutch data as OOD resources results in better performance than using Mandarin data. For A-BNF-Du and A-BNF-Ma that adopt cross-lingual phone labeling, the superiority of using Dutch data over using Mandarin data was found in a recent work [30]. The present study demonstrates the same finding in the cross-lingual AM based method by comparing C-BNF-Du and C-BNF-Ma.

(5) With 600-hour (or more) unlabeled training data of the target language available, the cross-lingual phone labeling method outperforms the cross-lingual AM based BNF method which does not rely on target unlabeled speech data but relies on OOD labeled speech data. This can be observed by, for instance, comparing A-BNF-Du with C-BNF-Du, or by comparing A-BNF-Ma with C-BNF-Ma. The superiority of cross-lingual phone labeling is consistent over all the evaluation sets and both the across- and within-speaker conditions. This superiority can be partially explained as the first method leverages both OOD transcribed data and in-domain unlabeled data in system development, while the second method leverages OOD transcribed data only[7].

Interestingly, the superiority of cross-lingual phone labeling over cross-lingual AM based BNF is more prominent when Mandarin data is chosen as OOD resource, compared to when Dutch data is chosen. For instance, in the across-speaker condition, the relative performance increase from C-BNF-Du to A-BNF-Du (trained with unlab-6K) is 14.1% relatively, while for the Mandarin models this relative increase is 30.7%. A possible explanation is that the cross-lingual AM based BNF method has a language mismatch between training and test acoustic data, while for the cross-lingual phone labeling method, the acoustic data during training and test are both from the target language. A larger language mismatch between the OOD language and the target language, e.g. Mandarin-English has a larger mismatch than Dutch-English, leads to larger negative effects on ABX performance.

C. ZeroSpeech 2017 evaluation results

Finally, we tested our approach on the ZeroSpeech 2017 data. Overall ABX error rates ($\%$) of the proposed approach (A-BNF-Du and A-BNF-Ma), the cross-lingual AM based BNF approach (C-BNF-Du and C-BNF-Ma), and the front-end APC representation on the ZeroSpeech 2017 English 1s, 10s and 120s evaluation sets are listed in Table VII separately and averaged over all evaluation sets. The official baseline and topline of ZeroSpeech 2017, systems *HE* [14], *CH* [41], *SH* [27] that are at the top of the ZeroSpeech 2017 leaderboard and the system *CPC+DA* [62] (same as CPC+DA discussed in Section VI-B) are listed in the table for comparison. The systems *HE* and *CH* utilized untranscribed speech data from the ZeroSpeech 2017 training sets only. The system *SH* utilized over $1,300$ hours of OOD transcribed data including 80 hours of English data during training, so *SH* is a supervised system. Please note that only the results on the $10s$ evaluation set of the CPC+DA approach are reported in [62].

TABLE VII: Overall ABX error rates ($\%$) of A-BNF-Du, A-BNF-Ma, the official MFCC baseline, the supervised topline, and state-of-the-art systems on the ZeroSpeech 2017 English evaluation sets. The topline and *SH* that are trained with supervised English data are marked with †.

	Across-speaker				Within-speaker			
	1s	10s	120s	Avg.	1s	10s	120s	Avg.
Training set: unlab-600								
A-BNF-Du	7.65	6.69	6.66	7.00	5.52	4.77	4.68	4.99
A-BNF-Ma	8.19	7.33	7.30	7.61	5.97	5.39	5.37	5.58
APC	14.36	12.59	12.49	13.15	9.80	8.28	8.26	8.78
Training set: unlab-6K								
A-BNF-Du	**6.91**	6.28	**6.26**	**6.48**	**4.96**	**4.53**	**4.54**	**4.68**
A-BNF-Ma	7.47	6.93	6.89	7.10	5.38	5.03	5.00	5.14
APC	12.00	10.79	10.70	11.16	8.32	7.28	7.21	7.60
Training set: ZeroSpeech 2017 (45 hours)								
CH [41]	8.1	8.0	8.0	8.03	5.6	5.5	5.5	5.53
HE [14]	10.1	8.7	8.5	9.10	6.9	6.2	6.0	6.37
*Training set **NOT** in Libri-light or ZeroSpeech 2017*								
Baseline [17]	23.4	23.4	23.4	23.4	12.0	12.1	12.1	12.1
†Topline [17]	8.6	6.9	6.7	7.40	6.5	5.3	5.1	5.63
†SH [27]	7.9	7.4	6.9	7.40	5.5	5.2	4.9	5.20
CPC+DA [62]	-	**5.8**	-	-	-	4.6	-	-
C-BNF-Du	7.81	7.79	7.78	7.79	5.59	5.58	5.60	5.59
C-BNF-Ma	10.60	10.62	10.57	10.60	7.67	7.58	7.56	7.60

From Table VII it can be seen that when trained with unlab-600, A-BNF-Du outperforms the supervised topline system. When trained with the larger, unlab-6K set, both A-BNF-Du and A-BNF-Ma outperform the topline. A-BNF-Du and A-BNF-Ma perform better than the unsupervised systems *CH, HE* and the supervised system *SH* both when trained with unlab-600 and with unlab-6K. The state-of-the-art system CPC+DA performs better than our best system (A-BNF-Du, trained with unlab-6K) in the across-speaker condition on the 10s evaluation set, while ours is better in the within-speaker condition. A-BNF-Du and A-BNF-Ma both perform better than APC in the unlab-600 and unlab-6K training data settings.

Table VII shows A-BNF-Du and A-BNF-Ma representations perform better than C-BNF-Du and C-BNF-Ma representations. It further confirms our finding in the previous section that of the two cross-lingual knowledge transfer methods, the cross-lingual phone labeling method (in the back-end of A-BNF-Du and A-BNF-Ma) performs better. The superiority of the cross-lingual phone labeling method over the cross-lingual AM based BNF method is more prominent when Mandarin data is chosen as the OOD resource, compared to when Dutch data is chosen. This is again in line with the finding in the previous section.

VII. In-depth analyses of the effectiveness of the proposed approach

The in-depth analyses of the performance of our proposed approach is conducted at two broad levels: at the phoneme level, using the phoneme-level ABX error rate (see Section IV-B); and at the AF level, using the pairwise and attribute-level ABX AF error rates (see Section IV-C). These analyses

[7]We do not claim that our method always outperforms the second one. In our unpublished results, A-BNF-Du trained with a 13-hour subset of unlab-600 performed worse than C-BNF-Du, which implies that large amounts of unlabeled target training data are required in order to get a good performance with our method.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

aim to uncover what phoneme and AF information is (not) captured by the learned representation from our proposed approach, which can be used to guide future research in further improving the proposed approach.

A. Setup

The in-depth analyses of the performance of our proposed approach are conducted on the dev-clean set from Libri-light. This set contains $2,703$ utterances summing up to 5.4 hours. The total number of frames is $1,934,785$.

The A-BNF-Du and A-BNF-Ma representations generated by our proposed approach trained using unlab-600 are chosen for the analyses. Moreover, the front-end APC features trained with unlab-600 and the official MFCC features are chosen for analyses in order to compare against A-BNF-Du and A-BNF-Ma.

The phoneme-level analysis uses the 39 English phonemes in the CMU Dictionary [72]: these are 10 monophthongs, 5 diphthongs, and 24 consonants. Calculation of $p_{co}(\omega_i)$ (see Equation (10)) depends on the ground-truth English phoneme labels and the cross-lingual phone labels. The English true phoneme labels for dev-clean are obtained by carrying out a forced alignment using the English TDNN AM that is described in Section V-B. Please note that during calculation of $p_{co}(\omega_i)$ using Mandarin cross-lingual phone labels, tone symbols are removed, in order to make a fair comparison of $p_{co}(\omega_i)$ between Dutch and Mandarin phone labels.

The AF-level analysis focuses on four AFs: MoA and PoA for consonants, and height and backness for vowels. In the analysis of each AF, t-SNE [73] is adopted to visualize the distribution of the learned representations with respect to the different attributes in an AF (e.g. fricative is an attribute in MoA). The number of speech frames per AF for the visualizations is around $2,400$ for all four AFs as a trade-off between computational complexity and avoiding sparsity of the visualization plots. The number of speech frames per AF attribute is equal for all attributes for the same AF, i.e., 500 for MoA, 300 for PoA, and 800 for vowel height and backness. The speech frames are randomly chosen from dev-clean.

B. Phoneme-level analysis results

Phoneme-level ABX error rates (%) of A-BNF-Du, A-BNF-Ma, APC, and MFCC representations are illustrated in Fig. 2 for the within-speaker (top panel) and across-speaker (bottom panel) conditions separately. The phonemes are sorted (left to right) based on the within-speaker relative error rate reduction from MFCC to A-BNF-Du in descending order. The distribution of the phoneme-level relative ABX error rate reductions (%) from MFCC to APC feature representations aggregated for monophthongs, diphthongs, and consonants for the within-speaker (left panel) and across-speaker (right panel) conditions are shown in Fig. 3, and from APC to A-BNF-Du (red boxes) and to A-BNF-Ma (black boxes) in Fig. 4. Figs. 3 and 4 reflect the phoneme-level ABX error reduction achieved by the front-end and the back-end respectively. In Figs. 3 and 4, each triangle point represents an individual phoneme. The horizontal dash-dotted line in each subfigure of Figs. 3 and

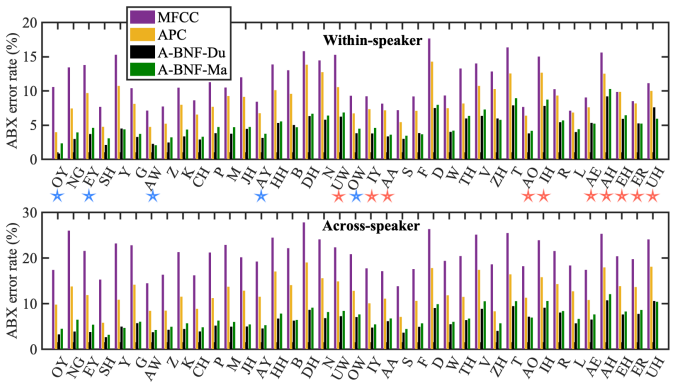

Fig. 2: Phoneme-level ABX error rates (%) of MFCC, APC, A-BNF-Du and A-BNF-Ma (lower is better). Phonemes are sorted (left to right) based on within-speaker relative error rate reduction from MFCC to A-BNF-Du in descending order. Orange and blue "☆" symbols denote monophthongs and diphthongs respectively, the rest are consonants.

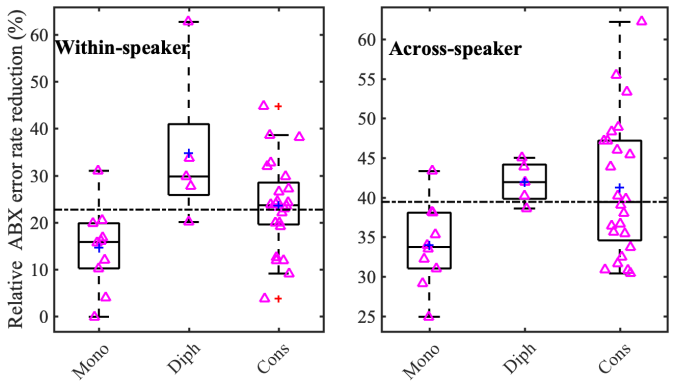

Fig. 3: Distribution of phoneme-level relative ABX error rate reduction (%) from MFCC to APC as the effect of front-end of the proposed approach (higher is better). "Mono", "Diph" and "Cons" are abbreviations of Monophthongs, Diphthongs and Consonants.

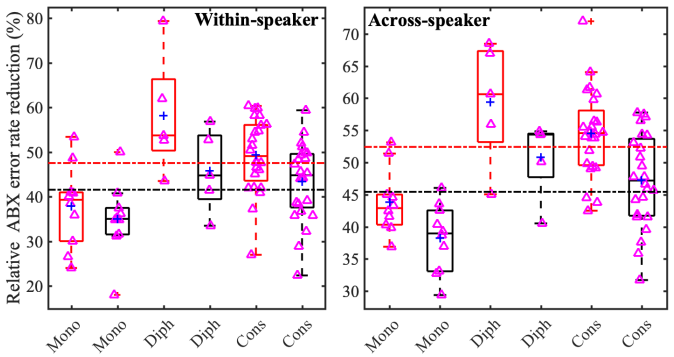

Fig. 4: Distribution of phoneme-level relative ABX error rate reduction (%) from APC to A-BNF-Du (red) and from APC to A-BNF-Ma (black) as the effect of back-end of the proposed approach (higher is better).

4 denotes the average of the phoneme-level relative error rate reduction over all phonemes. The "+" inside each box marks

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

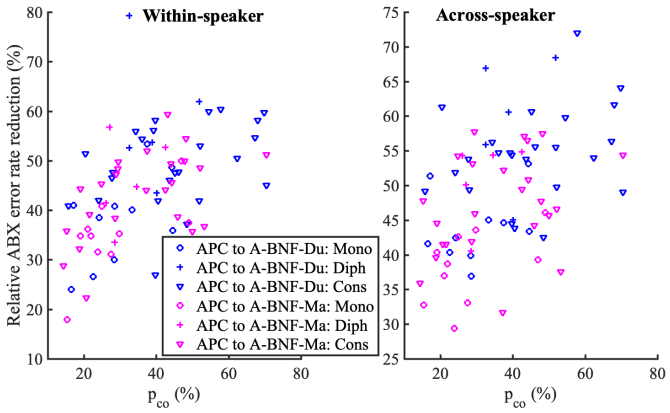

Fig. 5: Correlation between phoneme-level ABX error rate reduction (%) from APC to A-BNF-Du (blue)/A-BNF-Ma (pink) and $p_{co}(\%)$ for each English phoneme.

the mean value of all phonemes in that box.

Fig. 2 shows that diphthongs /OY/, /EY/, /AW/ and consonants /NG/, /SH/, /Y/, /G/ benefit the most from the learned A-BNF-Du representation of all 39 phonemes. In contrast, monophthongs /UH/, /ER/, /EH/, /AH/, /AE/ and consonants /R/ and /L/ benefit the least. Interestingly, phonemes with the largest error rate reductions (left-most in Fig. 2) are not only those having the highest error rates by the MFCC representation (e.g. /Y/ has a high error rate but /SH/ does not). In general, performance improvements obtained from MFCC to A-BNF-Du are larger for diphthongs than for monophthongs. This can be clearly observed when comparing the orange (denote monophthongs) and blue (denote diphthongs) stars in Fig. 2. For consonants, the improvements vary greatly. As is seen from Fig. 2, some consonants (/NG/ and /SH/) are among the phonemes with the largest improvements, while some (/L/ and /R/) are among the least.

Fig. 3 shows that with front-end APC pretraining, all phonemes show a positive error rate reduction (except the monophthong vowel /EH/ in the within-speaker condition). This demonstrates APC is effective not only from the perspective of overall ABX performance, which is shown in the previous section, but also towards each individual phoneme. Nevertheless, the gains differ for the different phonemes: one diphthong /OY/ gains a huge improvement (over 60%) when using APC in the within-speaker condition. while the biggest improvements in the across-speaker condition were found for three consonants: over 50% relative error rate reductions for /Y/, /SH/ and /ZH/. At the same time, the two monophthongs /EH/, /ER/ and consonant /L/ benefit little from APC pretraining in the within-speaker condition (less than 5%).

Fig. 4 shows that irrespective of using Dutch or Mandarin labels as cross-lingual labels, diphthongs benefit the most from the back-end cross-lingual BNF learning, followed by consonants, while monophthongs benefit the least. Moreover, using Dutch labels results in larger performance improvements than using Mandarin labels on all three phoneme categories. The advantage of the Dutch labels over the Mandarin ones is also illustrated in Fig. 2, where the majority of the phoneme-level ABX error rates by A-BNF-Du are lower than those by A-BNF-Ma (/Y/ and /UH/ are two exceptions in both the within- and the across-speaker condition).

From Fig. 3 and Fig. 4, it can be observed that both the front-end and the back-end of the proposed approach bring larger performance improvements for diphthongs than monophthongs. Recall in Fig. 2 we saw that A-BNF-Du achieved larger performance improvements over MFCC for diphthongs than for monophthongs, we conclude that the superiority of modeling diphthongs over monophthongs by A-BNF-Du results from a combined effect of APC pretraining and cross-lingual phone-aware DNN-BNF model training. The larger reduction in error rates for diphthongs than for monophthongs in the APC front-end can possibly be attributed to APC pretraining being better at modeling duration or at least longer sounds: diphthongs, which are long sounds, and long monophthongs benefit the most from APC pretraining in the within-speaker condition, i.e., /UW/ and /IY/ (not explicitly marked in Fig. 3), are both long monophthong vowels, while short monophthong vowels such as /UH/, /IH/ and /EH/ are among the monophthongs that benefit the least. Moreover, consonants that benefit the most from APC in the within-speaker condition are /NG/, /TH/, /SH/, /Z/. None of them are *stops*, which have a short time duration.

To gain deeper insights on why the back-end DNN-BNF model performs differently in capturing different target phonemes' information, Fig. 5 plots the correlation between the phoneme-level ABX error rate reduction (%) from APC to A-BNF-Du/-Ma and p_{co} (%) for English phonemes. In line with our expectation, a clear positive correlation for both the A-BNF-Du (blue) and the A-BNF-Ma (pink) representations and for both the within-speaker and the across-speaker conditions can be observed: a high $p_{co}(\omega)$ of an English phoneme ω indicates a high consistency of cross-lingual phone labels that are assigned to frames of phoneme ω. This means that frames of the same sounds in the target language are consistently assigned the same English label (irrespective of whether it is the *correct* English label), thus ensuring reliable frame label supervision for training the back-end DNN-BNF model to create a subword-discriminative feature representation of ω.

Furthermore, from Fig. 5, we can see that using Dutch phone labels (blue marks) results in more English phonemes getting a high p_{co} than Mandarin phone labels (pink marks). Specifically, in the case of using Mandarin labels, only for /B/ p_{co} is larger than 55% (see horizontal axis of either one subfigure in Fig. 5), while this is the case for 7 English phonemes (/S/, /M/, /K/, /N/, /P/, /NG/, /G/) when using Dutch labels. Interestingly, all of these are consonants. For monophthongs and diphthongs, regardless of using Dutch or Mandarin labels, their respective p_{co} rarely exceeds 50% (/EY/ is the only exception when using Dutch labels). This indicates that both monophthongs and diphthongs are less likely to obtain highly consistent cross-lingual phone labels than consonants, which explains why monophthongs benefit less from back-end cross-lingual DNN-BNF model training than consonants (see Fig. 4): the frame label supervision for training the DNN-BNF model is of less quality for monophthongs than for consonants.

It is worth noting that diphthongs were substantially benefiting from the back-end DNN-BNF model (see Fig. 4).

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

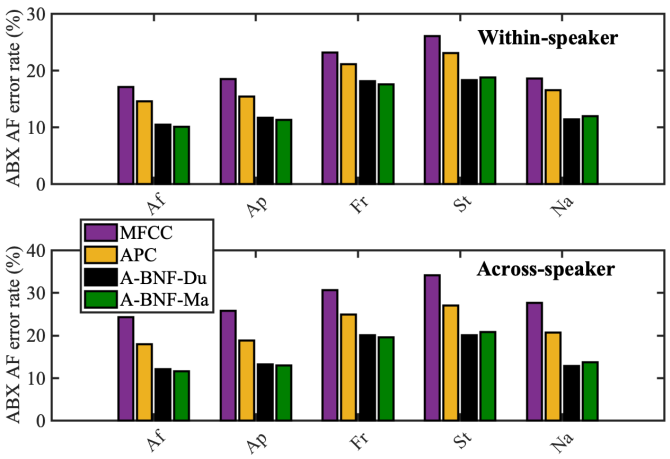

Fig. 6: MoA attribute-level ABX AF error rate (%) of MFCC, APC, A-BNF-Du, and A-BNF-Ma.

However, as mentioned above, diphthongs are less likely to obtain highly consistent cross-lingual phone labels than consonants, which appears to contrary to Fig. 4. Our explanation is as follows: p_{co} does not take into account the dynamic characteristics of diphthongs. Specifically, when cross-lingual phone labels are being generated by an OOD ASR system, it might well be that the first half and second half of the speech frames of a diphthong are labeled as two different cross-lingual phones. This would result in a low p_{co}, but does little to no harm to training the back-end DNN-BNF model to learn the representation of such a diphthong, because the DNN-BNF would learn to represent the diphthong as a sequence of two consecutive phones. An example is the diphthong /OY/ modeled by A-BNF-Du, which is marked as the top "+" in the left half of Fig. 5: while /OY/ gains the largest within-speaker ABX error rate reduction (79.4%), $p_{co}(/\mathrm{OY}/)$ is moderate (32.5%). Among all the speech frames of /OY/, the Dutch phone [o][8] occupies 32.5% of the frame labels, the Dutch phone [j][9] occupies 21.8%, and the rest are labeled as other Dutch phones with smaller percentages.

C. AF-level analysis results: Manner of articulation

Manner of articulation (MoA) attribute-level ABX AF error rates (%) of the MFCC, APC, A-BNF-Du, and A-BNF-Ma representations are shown in Fig. 6. MoA pairwise ABX AF error rates (%) of A-BNF-Du and A-BNF-Ma representations are listed in Table VIII.

Fig. 6 shows that both A-BNF-Du and A-BNF-Ma representations outperform MFCC and APC in capturing manner of articulation information. This shows that our approach proposed for modeling subword units implicitly learns information regarding manner of articulation of phonemes. A-BNF-Du better captures stop and nasal information than A-BNF-Ma, while A-BNF-Ma better captures affricate, approximant and fricative information. The comparison on the average MoA pairwise ABX AF error rates for A-BNF-Du and A-BNF-Ma (right-most column and bottom row in Table VIII)

[8]IPA symbol is [o:]. The vowel in *oost* (English translation: *east*).
[9]IPA symbol is [j]. The vowel in *ja* (English translation: *yes*).

TABLE VIII: MoA pairwise ABX AF error rates (%) of A-BNF-Du and A-BNF-Ma. **Pink** numbers denote across-speaker error rates, **black** numbers denote within-speaker error rates.

(a) A-BNF-Du

	Af	Ap	Fr	St	Na	avg.
Af	-	2.96	19.69	16.76	2.43	14.01
Ap	3.98	-	15.04	14.28	14.51	
Fr	22.65	16.44	-	25.70	12.04	
St	18.69	15.88	27.48	-	16.70	
Na	3.05	16.59	13.47	18.15	-	
avg.	15.72					

(b) A-BNF-Ma

	Af	Ap	Fr	St	Na	avg.
Af	-	2.62	18.54	17.01	2.23	13.96
Ap	3.26	-	14.21	13.76	14.74	
Fr	20.68	15.68	-	25.61	12.08	
St	19.22	15.36	28.13	-	18.82	
Na	3.07	17.53	13.77	20.36	-	
avg.	15.64					

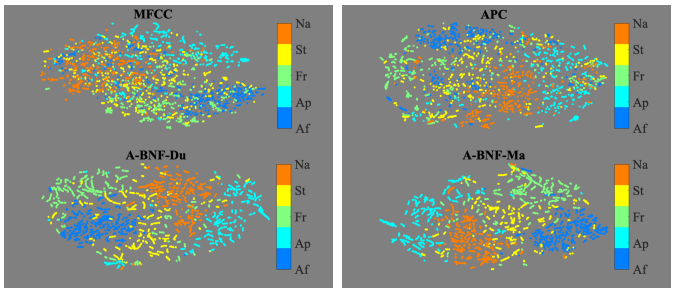

Fig. 7: T-SNE visualization of the frame-level MFCC, APC, A-BNF-Du and A-BNF-Ma representations. Each color denotes a different MoA attribute.

shows slightly lower average error rates for A-BNF-Ma than for A-BNF-Du in both the within-speaker and across-speaker conditions. This is in contrast to the phoneme-level results which showed consistently better results for A-BNF-Du over A-BNF-Ma. Taken together, this suggests that manner of articulation information is less language-dependent than phoneme information.

Fig. 6 furthermore shows that stops and fricatives have consistently higher ABX AF error rates than the other three MoA attributes for MFCC, APC, A-BNF-Du, and A-BNF-Ma representations. Table VIII shows that fricatives are often confused with affricates and stops while stops are often confused with fricatives. From an articulatory-acoustic point of view this can be explained by stops being very short and highly dynamic sounds. They start with a stretch of silence or low noise (in the case of voiced stops) when the vocal tract is fully closed, followed by a release that consists of noise caused by turbulence of the air in the vocal tract. A fricative solely consists of this noise, while an affricate can be seen as a concatenation of a short stop and a short fricative, so has acoustic characteristics of both.

Fig. 7 shows the t-SNE visualizations of the speech frames when using the MFCC, APC, A-BNF-Du, and A-BNF-Ma feature representations and labeling the frames with their MoA attributes. Each color represents a different MoA attribute,

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

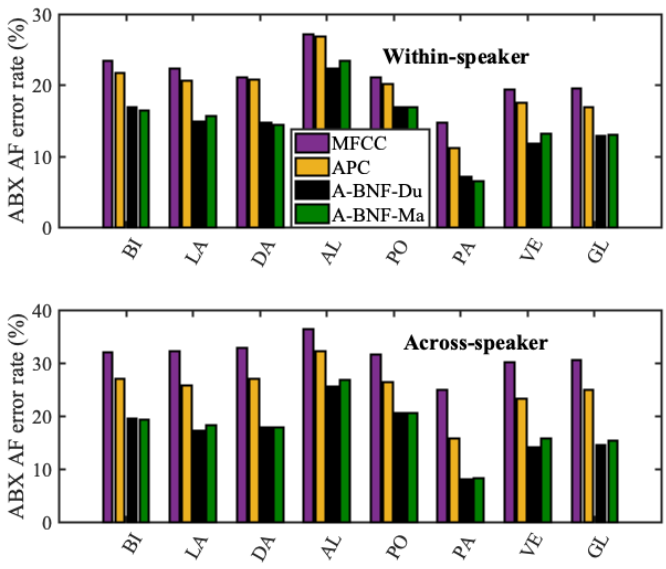

Fig. 8: PoA attribute-level ABX AF error rate (%) of MFCC, APC, A-BNF-Du and A-BNF-Ma.

and each sample point stands for a speech frame. This figure clearly demonstrates that using A-BNF-Du and A-BNF-Ma as in our proposed approach results in MoA attributes that form more explicit attribute-specific patterns than with MFCC and APC (compare bottom two panels with the top two panels).

For A-BNF-Du and A-BNF-Ma, the clusters of affricates, approximants and nasals are more coherent than those of fricatives and stops. This is in agreement with the relatively higher ABX AF error rates of fricative and stop as shown in Fig. 6. Moreover, for the A-BNF-Du and A-BNF-Ma representations, affricates generally separate well from approximants and nasals, and are less separable from fricatives and from stops, which is in line with the articulatory-acoustic properties of affricates, fricatives, and stops. Table VIII further confirms this finding: MoA pairwise error rates of affricate-fricative and affricate-stop are much higher than those of affricate-approximant and affricate-nasal. By comparing visualizations of A-BNF-Du and A-BNF-Ma with MoA attributes, no noticeable difference is found.

D. AF-level analysis results: Place of articulation

Place of articulation (PoA) attribute-level ABX AF error rates (%) of MFCC, APC, A-BNF-Du, and A-BNF-Ma representations are shown in Fig. 8. PoA pairwise ABX AF error rates (%) of A-BNF-Du and A-BNF-Ma representations are listed in Table IX.

Fig. 8 shows that the A-BNF-Du and A-BNF-Ma representations consistently capture place of articulation information better than MFCC and APC for all PoA attributes and for both the within- and across speaker conditions. This demonstrates the effectiveness of our approach in capturing information that distinguishes PoA attributes. A-BNF-Du performs the best in capturing labiodental, alveolar and velar information in both within- and across-speaker conditions, while A-BNF-Ma performs the best in capturing bilabial, dental and palatal

TABLE IX: PoA pairwise ABX AF error rates (%) of A-BNF-Du and A-BNF-Ma. **Pink** numbers denote across-speaker error rates, **black** numbers denote within-speaker error rates.

(a) A-BNF-Du

	BI	LA	DA	AL	PO	PA	VE	GL	avg.
BI	-	20.01	18.58	25.50	20.60	5.66	15.72	12.70	
LA	23.05	-	16.72	25.93	18.46	3.34	10.45	9.86	
DA	21.94	20.81	-	27.72	14.42	4.82	9.19	11.39	
AL	28.45	29.09	32.66	-	23.97	13.61	21.17	18.50	14.74
PO	23.60	21.29	19.90	28.11	-	10.39	13.25	17.70	
PA	6.23	3.16	3.61	15.01	16.26	-	2.93	9.77	
VE	18.48	12.41	11.56	23.84	16.98	4.11	-	10.50	
GL	15.09	11.31	14.27	21.67	18.87	8.74	11.58	-	
avg.				17.22					

(b) A-BNF-Ma

	BI	LA	DA	AL	PO	PA	VE	GL	avg.
BI	-	20.23	17.96	25.25	18.48	4.14	16.48	12.40	
LA	23.15	-	17.52	27.77	17.93	3.38	12.52	10.09	
DA	21.50	22.18	-	27.22	14.16	3.13	9.21	11.90	
AL	28.47	31.33	31.88	-	24.83	14.17	24.51	19.81	14.98
PO	21.90	21.29	18.48	29.36	-	10.99	14.20	17.74	
PA	5.70	2.54	4.83	16.44	15.49	-	3.37	7.01	
VE	19.80	15.32	12.67	27.24	17.89	3.55	-	12.53	
GL	15.14	12.38	14.17	23.59	19.87	9.38	14.02	-	
avg.				17.84					

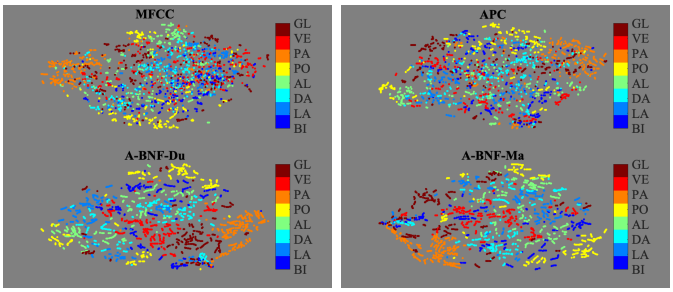

Fig. 9: T-SNE visualization of the frame-level MFCC, APC, A-BNF-Du and A-BNF-Ma representations. Each color denotes a different PoA attribute.

information in within-speaker conditions. The comparison on the average PoA pairwise ABX AF error rates for A-BNF-Du and A-BNF-Ma (right-most column and bottom row in Table IX) shows little (less than absolute 0.25%) difference in both the within-speaker and across-speaker conditions. Moreover, Fig. 8 shows no clear advantage between A-BNF-Du and A-BNF-Ma on PoA attribute-level ABX AF error rates. This suggests that place of articulation information is less language-dependent than phoneme information, as the phoneme-level results showed consistently better results for A-BNF-Du over A-BNF-Ma.

Fig. 8 shows that palatal has the lowest attribute-level ABX AF error rate. This is due to low pairwise ABX AF error rates between palatal and any other PoA attribute as shown in Table IX. The fact that the A-BNF-Du and A-BNF-Ma are better able to capture palatal information than information of the other PoA attributes is likely due to the fact that palatal only concerns a single phoneme (/Y/) with a clear acoustic pattern. This is unlike the other PoA attribute that only consists of a single phoneme: glottal (/HH/). The attribute-level ABX AF error rate of glottal is much higher than that of palatal, and is similar to velar. This is likely due to the very low energy

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

of /HH/ which makes it hard to distinguish the acoustics of /HH/ from silence and other phonemes (such as the first parts of stops).

Alveolar has the highest attribute-level ABX AF error rate for A-BNF-Du and A-BNF-Ma, which is due to high pairwise ABX AF error rates between alveolar and any other PoA attribute (see Table IX; except for the alveolar-palatal pair). Bilabial and postalveolar also have high attribute-level ABX AF error rates for A-BNF-Du and A-BNF-Ma: higher than other attributes except alveolar. This is likely due to the PoA attributes of alveolar, bilabial and postalveolar containing phonemes with at least three different manners of articulation (see Table I), leading to highly diverse acoustics within each of the PoA attributes.

Fig. 9 shows the t-SNE visualizations of the speech frames when using the MFCC, APC, A-BNF-Du, and A-BNF-Ma feature representations and labeling the frames with their PoA attributes. Each color represents a different PoA attribute, and each sample point stands for a speech frame. This figure clearly demonstrates that the PoA attributes form more explicit attribute-specific patterns when using the A-BNF-Du and A-BNF-Ma feature representations than with the MFCC and APC representations (compare the top panels with the bottom panels). The cluster of palatals is coherent in the A-BNF-Du and A-BNF-Ma representations. The clusters of glottals and velars are coherent in A-BNF-Du, and less coherent in A-BNF-Ma. There are no coherent clusters of the other five PoA attributes shown in A-BNF-Du and A-BNF-Ma. This is consistent with Fig. 8 which shows palatals, glottals and velars having lower attribute-level ABX AF error rates than the other five attributes in A-BNF-Du and A-BNF-Ma. Bilabials, labiodentals, and dentals show overlap in both A-BNF-Du and A-BNF-Ma.

Comparing the results of the MoA and PoA analyses shows that the A-BNF-Du and A-BNF-Ma representations are better able to capture the underlying MoA and PoA information than the MFCC and APC representations. At the same time MoA information is better captured than PoA information by the A-BNF-Du and A-BNF-Ma representations (compare average pairwise ABX AF error rates in Tables VIII and IX). This is in line with results found in [74] which showed that a naive feed-forward DNN trained for the vowel-consonant classification task captures manner of articulation information better than place of articulation information (without being explicitly trained to do so).

E. AF-level analysis results: Vowel height and backness

Our final analyses focus on the effectiveness of our approach in capturing vowel height and backness information. Attribute-level ABX AF error rates (%) for vowel height are illustrated in Fig. 10, and those for vowel backness are illustrated in Fig. 11. Pairwise ABX AF error rates (%) of A-BNF-Du and A-BNF-Ma are listed in Tables X and XI, respectively.

Fig. 10 and Fig. 11 show that, for both vowel height and backness, the attribute in the middle, i.e., *mid* in height and *central* in backness, performs consistently worse than the other attributes in both the within- and the across-speaker conditions.

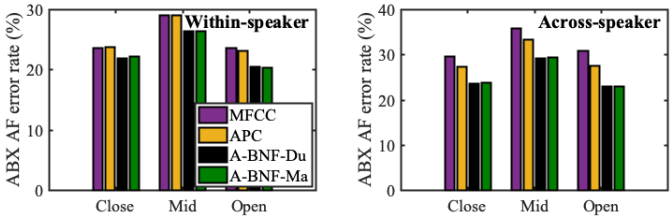

Fig. 10: Vowel height attribute-level ABX AF error rate (%) of MFCC, APC, A-BNF-Du and A-BNF-Ma.

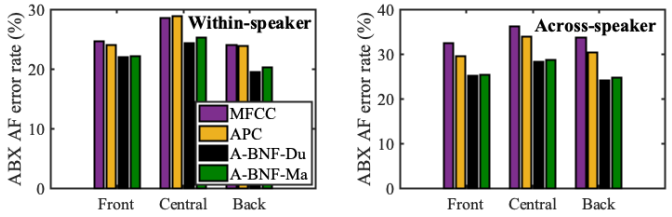

Fig. 11: Vowel backness attribute-level ABX AF error rate (%) of MFCC, APC, A-BNF-Du and A-BNF-Ma.

TABLE X: Vowel height pairwise ABX AF error rates (%) of A-BNF-Du and A-BNF-Ma. **Pink** numbers denote across-speaker error rates, **black** numbers denote within-speaker error rates. "Cl, Mi, Op" stand for "Close, Mid, Open".

(a) A-BNF-Du

	Cl	Mi	Op	avg.
Cl	-	27.75	15.99	22.93
Mi	30.00	-	25.06	
Op	17.38	28.47	-	
avg.	25.40			

(b) A-BNF-Ma

	Cl	Mi	Op	avg.
Cl	-	28.28	16.08	23.02
Mi	30.29	-	24.71	
Op	17.30	28.53	-	
avg.	25.40			

TABLE XI: Vowel backness pairwise ABX AF error rates (%) of A-BNF-Du and A-BNF-Ma. **Pink** numbers denote across-speaker error rates, **black** numbers denote within-speaker error rates. "Fr, Ce, Ba" stand for "Front, Central, Back".

(a) A-BNF-Du

	Fr	Ce	Ba	avg.
Fr	-	26.78	17.20	21.98
Ce	29.28	-	21.96	
Ba	21.12	27.35	-	
avg.	25.90			

(b) A-BNF-Ma

	Fr	Ce	Ba	avg.
Fr	-	27.15	17.34	22.64
Ce	29.52	-	23.44	
Ba	21.47	27.94	-	
avg.	26.25			

This worse performance is due to the large confusion of *mid* with *close* and *open* (see Table X), and of *central* with *front* and *back* (see Table XI).

Fig. 10 and Fig. 11 also show that, similar to what was observed for MoA and PoA for consonants, A-BNF-Du and A-BNF-Ma outperform MFCC and APC features in capturing the attributes of vowel height and backness. Comparing these results to the attribute-level ABX AF error rates for MoA (Fig. 6) and PoA (Fig. 8) shows that the monophthong vowel-related[10] AF attributes achieved higher error rates than the consonant-related ones (i.e., MoA and PoA). In fact, most attribute-level ABX AF error rates for vowel height and backness obtained from A-BNF-Du and A-BNF-Ma fall within

[10]Diphthongs are excluded in the vowel height and backness analyses, see Section IV-C.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

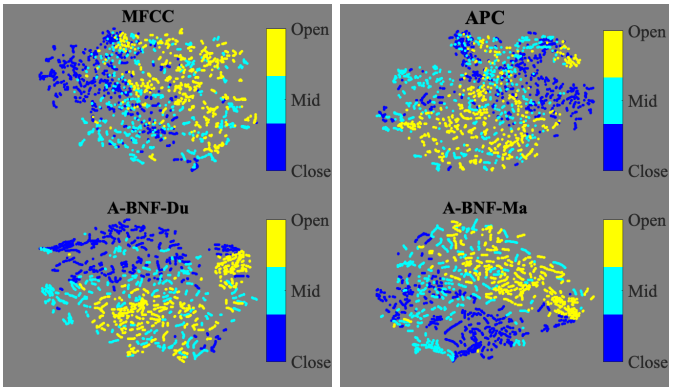

Fig. 12: T-SNE visualization of the frame-level MFCC, APC, A-BNF-Du and A-BNF-Ma representations. Each color denotes a different vowel height attribute.

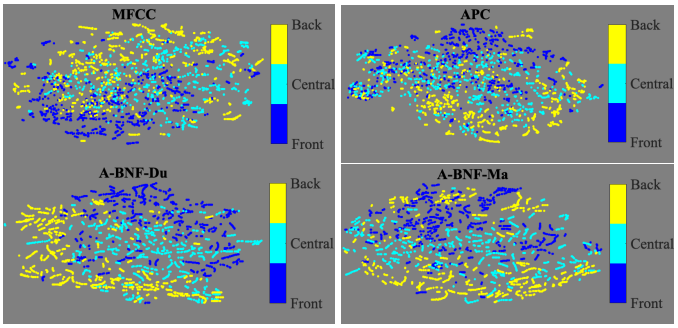

Fig. 13: T-SNE visualization of the frame-level MFCC, APC, A-BNF-Du and A-BNF-Ma representations. Each color denotes a different vowel backness attribute.

the $20\% - 30\%$ range, whereas most of those of MoA and PoA fall within the $10\% - 20\%$ range. This observation is in accordance with findings in the phoneme-level analysis, where monophthongs are found to benefit less than consonants by our proposed approach (see Section VII-B).

Fig. 12 and Fig. 13 show the t-SNE visualizations of the frames when using the MFCC, APC, A-BNF-Du, and A-BNF-Ma feature representations and labeling them with vowel height and vowel backness respectively. Each color represents a different attribute in vowel height or backness, and each sample point stands for a speech frame. Both figures show that the A-BNF-Du and A-BNF-Ma representations are better than MFCC and APC in capturing information that distinguishes vowel height and backness attributes, as they form more explicit attribute-specific patterns than with MFCC and APC. Fig. 12 shows that for A-BNF-Du and A-BNF-Ma, while there are noticeable overlaps between the *mid-close* pair and between the *mid-open* pair, little overlap is observed between the *close-open* pair. It indicates that the information to distinguish the *close* and *open* attributes in vowel height are well learned by the proposed approach. This is in contrast to vowel backness (see Fig. 13) where there is extensive overlap between *front*, *central* and *back* for A-BNF-Du and A-BNF-Ma.

Interestingly, the improvement of A-BNF-Du and A-BNF-Ma over MFCC in capturing vowel height and backness information seems not to be due to front-end APC pretraining. Comparing APC with MFCC in Fig. 10 and Fig. 11 shows that front-end APC pretraining has very limited or even a negative effect on capturing vowel height and backness information, especially in the within-speaker evaluation condition. This is contrary to what was observed for consonant information, where APC pretraining was found to be effective in capturing MoA and PoA (see Fig. 6 and Fig. 8). These results are in line with those observed in the phoneme analyses in Fig. 3, where the efficacy of APC pretraining is more prominent on consonants than on monophthongs.

VIII. Conclusion

The present study addresses unsupervised subword modeling. A two-stage learning framework that consists of an APC front-end and a cross-lingual DNN-BNF back-end was proposed to tackle this problem. To evaluate the proposed approach, in addition to the widely adopted ABX subword discriminability metric, a comprehensive and systematic analysis was carried out at the phoneme-level and the articulatory feature (AF)-level to investigate the type of information that is (and is not) captured by the newly created feature representations. In order to do so, new metrics that focus on phoneme-level ABX subword discriminability and attribute-level ABX AF discriminability have been proposed.

Experiments were conducted using two databases: Libri-light and ZeroSpeech 2017. Using the overall ABX subword discriminability metric, the experimental results show that our approach is competitive or even superior to the state-of-the-art [62]. Front-end APC pretraining brings performance improvement to the entire learning framework compared to a system with only the DNN-BNF back-end. Performance further increased when the amount of training material was increased from 600 hours to $6,000$ hours. The proposed system's best performance, achieved by using $6,000$ hours of untranscribed training data without any linguistic knowledge of the target language, is very close to that of a supervised system trained on $1,000$-hour transcribed data of the target language. Moreover, the proposed back-end performs better than a cross-lingual AM based BNF method in exploiting cross-lingual knowledge transfer.

Subsequent in-depth analyses investigated what information was captured by the newly created feature representations and this was compared to the information captured by baseline MFCC features and front-end APC features. The phoneme-level analysis showed that compared to MFCC, our two-stage approach achieves is better able to capture diphthong information than monophthong vowel information, and this is true in both the front-end and the back-end of our approach. For consonants, the improvement in capturing phoneme information from MFCC to our approach varies greatly to different consonants. Our results showed a positive correlation between the effectiveness of the back-end in capturing a target phoneme's information and the quality of cross-lingual phone labels assigned to that target phoneme.

The AF-level analyses showed that the proposed approach is better than MFCC and front-end APC features in capturing

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

manner and place of articulation information and vowel height and backness information. In the analysis of MoA, stop and fricative information are less well captured than affricate, approximant, and nasal information. The analysis of PoA showed that palatal is the best captured attribute, which is partially explained by the palatal AF attribute only consisting of a single phoneme /Y/, while most other PoA attributes consist of multiple phonemes with multiple manners of articulation. The analyses indicate MoA is better captured by the proposed approach than PoA which is in line with previous research [74], and both MoA and PoA information are better captured than vowel height and backness information. Comparing the outcomes of the analyses at the AF and phoneme level suggests that AF information is less language-dependent than phoneme information, which is in line with the linguistic principles underlying articulatory features and phonemes.

In conclusion, both the front-end and back-end of the proposed approach are effective in capturing information that distinguishes individual phonemes. It demonstrates the importance of both the front-end and the back-end of our approach in the task of unsupervised subword modeling. Regarding AF information, the front-end is effective in capturing MoA and PoA information, but is less well able to capture vowel height and backness information. In contrast, the back-end is effective in capturing all the MoA, PoA, vowel height and backness information. The phoneme-level and the AF-level analyses both indicate monophthong vowel information is much more difficult to capture than consonant information. This suggests that a possible direction to improve unsupervised subword modeling is investigating methods that improve the effectiveness of capturing monophthong vowel information.

REFERENCES

[1] P. K. Austin and J. Sallabank, *The Cambridge handbook of endangered languages*. Cambridge University Press, 2011.

[2] O. Scharenborg, L. Ondel, S. Palaskar, P. Arthur, F. Ciannella, M. Du, E. Larsen, D. Merkx, R. Riad, L. Wang, E. Dupoux, L. Besacier, A. W. Black, M. Hasegawa-Johnson, F. Metze, G. Neubig, S. Stüker, P. Godard, and M. Müller, "Speech technology for unwritten languages," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 964–975, 2020.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[4] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 30–42, 2011.

[5] C.-y. Lee and J. Glass, "A nonparametric bayesian approach to acoustic model discovery," in *Proc. ACL*, 2012, pp. 40–49.

[6] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Acoustic segment modeling with spectral clustering methods," *IEEE/ACM Trans. ASLP*, vol. 23, no. 2, pp. 264–277, 2015.

[7] H. Chen, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Parallel inference of Dirichlet process Gaussian mixture models for unsupervised acoustic modeling: A feasibility study," in *Proc. INTERSPEECH*, 2015, pp. 3189–3193.

[8] L. Ondel, L. Burget, J. Cernocký, and S. Kesiraju, "Bayesian phonotactic language model for acoustic unit discovery," in *Proc. ICASSP*, 2017, pp. 5750–5754.

[9] A. Tjandra, B. Sisman, M. Zhang, S. Sakti, H. Li, and S. Nakamura, "VQVAE unsupervised unit discovery and multi-scale code2spec inverter for Zerospeech Challenge 2019," in *Proc. INTERSPEECH*, 2019, pp. 1118–1122.

[10] S. Feng, T. Lee, and Z. Peng, "Combining adversarial training and disentangled speech representation for robust zero-resource subword modeling," in *Proc. INTERSPEECH*, 2019, pp. 1093–1097.

[11] M. Versteegh, R. Thiollière, T. Schatz, X.-N. Cao, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015." in *Proc. INTERSPEECH*, 2015, pp. 3169–3173.

[12] C.-y. Lee, T. J. O'Donnell, and J. R. Glass, "Unsupervised lexicon discovery from acoustic input," *TACL*, vol. 3, pp. 389–403, 2015.

[13] L. Ondel, P. Godard, L. Besacier, E. Larsen, M. Hasegawa-Johnson, O. Scharenborg, E. Dupoux, L. Burget, F. Yvon, and S. Khudanpur, "Bayesian models for unit discovery on a very low resource language," in *Proc. ICASSP*, 2018, pp. 5939–5943.

[14] M. Heck, S. Sakti, and S. Nakamura, "Feature optimized DPGMM clustering for unsupervised subword modeling: A contribution to zerospeech 2017," in *Proc. ASRU*, 2017, pp. 740–746.

[15] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Advances in NIPS*, 2017, pp. 6306–6315.

[16] S. Feng and T. Lee, "Exploiting cross-lingual speaker and phonetic diversity for unsupervised subword modeling," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 27, no. 12, pp. 2000–2011, 2019.

[17] E. Dunbar, X.-N. Cao, J. Benjumea, J. Karadayi, M. Bernard, L. Besacier, X. Anguera, and E. Dupoux, "The zero resource speech challenge 2017," in *Proc. ASRU*, 2017, pp. 323–330.

[18] H. Chen, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Unsupervised bottleneck features for low-resource query-by-example spoken term detection," in *INTERSPEECH*, 2016, pp. 923–927.

[19] S. Feng, T. Lee, and H. Wang, "Exploiting language-mismatched phoneme recognizers for unsupervised acoustic modeling," in *Proc. ISCSLP*, 2016, pp. 1–5.

[20] L.-s. Lee, J. Glass, H.-y. Lee, and C.-a. Chan, "Spoken content retrieval—beyond cascading speech recognition with text retrieval," *IEEE/ACM Trans. ASLP*, vol. 23, no. 9, pp. 1389–1420, 2015.

[21] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen *et al.*, "Libri-light: A benchmark for ASR with limited or no supervision," in *Proc. ICASSP*, 2020, pp. 7669–7673.

[22] T. K. Ansari, R. Kumar, S. Singh, and S. Ganapathy, "Deep learning methods for unsupervised acoustic modeling - LEAP submission to zerospeech challenge 2017," in *Proc. ASRU*, 2017, pp. 754–761.

[23] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018.

[24] W.-N. Hsu, Y. Zhang, and J. R. Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," in *Advances in NIPS*, 2017, pp. 1876–1887.

[25] P.-J. Last, H. A. Engelbrecht, and H. Kamper, "Unsupervised feature learning for speech using correspondence and siamese networks," *IEEE Signal Processing Letters*, vol. 27, pp. 421–425, 2020.

[26] N. Zeghidour, G. Synnaeve, N. Usunier, and E. Dupoux, "Joint learning of speaker and phonetic similarities with siamese networks," in *Proc. INTERSPEECH*, 2016, pp. 1295–1299.

[27] H. Shibata, T. Kato, T. Shinozaki, and S. Watanabe, "Composite embedding systems for zerospeech2017 track 1," in *Proc. ASRU*, 2017, pp. 747–753.

[28] S. Feng and T. Lee, "Exploiting speaker and phonetic diversity of mismatched language resources for unsupervised subword modeling," in *Proc. INTERSPEECH*, 2018, pp. 2673–2677.

[29] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, "An unsupervised autoregressive model for speech representation learning," in *Proc. INTERSPEECH*, 2019, pp. 146–150.

[30] S. Feng and O. Scharenborg, "Unsupervised Subword Modeling Using Autoregressive Pretraining and Cross-Lingual Phone-Aware Modeling," in *Proc. Interspeech 2020*, 2020, pp. 2732–2736.

[31] J. Chang and J. W. Fisher III, "Parallel sampling of DP mixture models using sub-cluster splits," in *Advances in NIPS*, 2013, pp. 620–628.

[32] Y. Higuchi, N. Tawara, T. Kobayashi, and T. Ogawa, "Speaker adversarial training of DPGMM-based feature extractor for zero-resource languages," in *Proc. INTERSPEECH*, 2019, pp. 266–270.

[33] H. Chen, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Multilingual bottleneck feature learning from untranscribed speech," in *Proc. ASRU*, 2017, pp. 727–733.

[34] Y. Yuan, C.-C. Leung, L. Xie, H. Chen, B. Ma, and H. Li, "Extracting bottleneck features and word-like pairs from untranscribed speech for feature representations," in *Proc. ASRU*, 2017, pp. 734–739.

[35] B. Wu, S. Sakti, J. Zhang, and S. Nakamura, "Optimizing DPGMM clustering in zero-resource setting based on functional load," in *Proc. SLTU*, 2018, pp. 1–5.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3076914, IEEE Open Journal of Signal Processing

[36] T. K. Ansari, R. Kumar, S. Singh, S. Ganapathy, and S. Devi, "Unsupervised HMM posteriograms for language independent acoustic modeling in zero resource conditions," in *Proc. ASRU*, 2017, pp. 762–768.

[37] C. Manenti, T. Pellegrini, and J. Pinquier, "Unsupervised speech unit discovery using k-means and neural networks," in *Proc. SLSP*, 2017, pp. 169–180.

[38] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," in *Proc. INTERSPEECH*, 2019, pp. 161–165.

[39] L. Badino, C. Canevari, L. Fadiga, and G. Metta, "An auto-encoder based approach to unsupervised learning of subword units," in *Proc. ICASSP*. IEEE, 2014, pp. 7634–7638.

[40] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," in *Proc. ICCV*, 2017, pp. 2051–2060.

[41] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using wavenet autoencoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2041–2053, 2019.

[42] B. van Niekerk, L. Nortje, and H. Kamper, "Vector-Quantized Neural Networks for Acoustic Unit Discovery in the ZeroSpeech 2020 Challenge," in *Proc. Interspeech 2020*, 2020, pp. 4836–4840. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1693

[43] P. L. Tobing, T. Hayashi, Y.-C. Wu, K. Kobayashi, and T. Toda, "Cyclic Spectral Modeling for Unsupervised Unit Discovery into Voice Conversion with Excitation and Waveform Modeling," in *Proc. INTERSPEECH*, 2020, pp. 4861–4865.

[44] A. Tjandra, S. Sakti, and S. Nakamura, "Transformer VQ-VAE for Unsupervised Unit Discovery and Speech Synthesis: ZeroSpeech 2020 Challenge," in *Proc. INTERSPEECH*, 2020, pp. 4851–4855.

[45] E. Dunbar, J. Karadayi, M. Bernard, X.-N. Cao, R. Algayres, L. Ondel, L. Besacier, S. Sakti, and E. Dupoux, "The Zero Resource Speech Challenge 2020: Discovering Discrete Subword and Word Units," in *Proc. INTERSPEECH*, 2020, pp. 4831–4835.

[46] W.-N. Hsu and J. R. Glass, "Extracting domain invariant features by unsupervised learning for robust automatic speech recognition," in *Proc. ICASSP*, 2018, pp. 5614–5618.

[47] S. Feng and T. Lee, "Improving unsupervised subword modeling via disentangled speech representation learning and transformation," in *Proc. INTERSPEECH*, 2019, pp. 281–285.

[48] M. Rivière, A. Joulin, P. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," in *Proc. ICASSP*, 2020, pp. 7414–7418.

[49] L. van Staden and H. Kamper, "A comparison of self-supervised speech representations as input features for unsupervised acoustic word embeddings," in *Proc. STL*, 2021, pp. 927–934.

[50] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.

[51] Beijing DataTang Technology Co., Ltd, "Aidatatang 200zh, a free Chinese Mandarin speech corpus."

[52] E. Hermann, H. Kamper, and S. Goldwater, "Multilingual and unsupervised subword modeling for zero-resource languages," *Computer Speech & Language*, vol. 65, p. 101098, 2021.

[53] S. Feng and T. Lee, "On the linguistic relevance of speech units learned by unsupervised acoustic modeling," in *Proc. INTERSPEECH*, 2017, pp. 2068–2072.

[54] D. Harwath and J. Glass, "Towards visually grounded sub-word speech unit discovery," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3017–3021.

[55] Y. Matusevych, T. Schatz, H. Kamper, N. H. Feldman, and S. Goldwater, "Evaluating computational models of infant phonetic learning across languages," *arXiv preprint arXiv:2008.02888*, 2020.

[56] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." in *Proc. INTERSPEECH*, 2014, pp. 338–342.

[57] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. ICML*, 2014, pp. 1764–1772.

[58] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016, pp. 4960–4964.

[59] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Proc. INTERSPEECH*, 2016, pp. 2751–2755.

[60] E. Dunbar, R. Algayres, J. Karadayi, M. Bernard, J. Benjumea, X.-N. Cao, L. Miskic, C. Dugrain, L. Ondel, A. W. Black, L. Besacier, S. Sakti, and E. Dupoux, "The zero resource speech challenge 2019: TTS without T," in *Proc. INTERSPEECH*, 2019, pp. 1088–1092.

[61] T. Tsuchiya, N. Tawara, T. Ogawa, and T. Kobayashi, "Speaker invariant feature extraction for zero-resource languages with adversarial learning," in *Proc. ICASSP*, 2018, pp. 2381–2385.

[62] E. Kharitonov, M. Rivière, G. Synnaeve, L. Wolf, P.-E. Mazaré, M. Douze, and E. Dupoux, "Data augmenting contrastive learning of speech representations in the time domain," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 215–222.

[63] W. Munson and M. B. Gardner, "Standardizing auditory tests," *The Journal of the Acoustical Society of America*, vol. 22, no. 5, pp. 675–675, 1950.

[64] Wikipedia contributors, "Arpabet — Wikipedia, the free encyclopedia," 2020, [Online; accessed 7-July-2020]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=ARPABET&oldid=965012012

[65] N. Oostdijk, "The spoken Dutch corpus. overview and first evaluation." in *LREC*. Athens, Greece, 2000, pp. 887–894.

[66] L. van der Werff, "kaldi_egs_CGN." [Online]. Available: https://github.com/laurensw75/kaldi_egs_CGN

[67] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv*, vol. abs/1412.6980, 2014.

[68] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[69] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proc. ICSLP*, 2002, pp. 901–904.

[70] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition." in *Proc. INTERSPEECH*, 2015, pp. 3586–3589.

[71] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *Proc. ICASSP*, 2014, pp. 2494–2498.

[72] [Online]. Available: http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[73] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[74] O. Scharenborg, N. van der Gouw, M. A. Larson, and E. Marchiori, "The representation of speech in deep neural networks," in *Proc. MMM*, vol. 11296. Springer, 2019, pp. 194–205.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/