MULTILINGUAL AND CROSSLINGUAL SPEECH RECOGNITION USING PHONOLOGICAL-VECTOR BASED PHONE EMBEDDINGS

Chengrui Zhu, Keyu An, Huahuan Zheng, Zhijian Ou†

Speech Processing and Machine Intelligence (SPMI) Lab, Tsinghua University, China
Beijing National Research Center for Information Science and Technology, China

ABSTRACT

The use of phonological features (PFs) potentially allows language-specific phones to remain linked in training, which is highly desirable for information sharing for multilingual and crosslingual speech recognition methods for low-resourced languages. A drawback suffered by previous methods in using phonological features is that the acoustic-to-PF extraction in a bottom-up way is itself difficult. In this paper, we propose to join phonology driven phone embedding (top-down) and deep neural network (DNN) based acoustic feature extraction (bottom-up) to calculate phone probabilities. The new method is called JoinAP (Joining of Acoustics and Phonology). Remarkably, no inversion from acoustics to phonological features is required for speech recognition. For each phone in the IPA (International Phonetic Alphabet) table, we encode its phonological features to a phonological-vector, and then apply linear or nonlinear transformation of the phonological-vector to obtain the phone embedding. A series of multilingual and crosslingual (both zero-shot and few-shot) speech recognition experiments are conducted on the CommonVoice dataset (German, French, Spanish and Italian) and the AISHLL-1 dataset (Mandarin), and demonstrate the superiority of JoinAP with nonlinear phone embeddings over both JoinAP with linear phone embeddings and the traditional method with flat phone embeddings.

Index Terms— multilingual, crosslingual, speech recognition, phonological feature, phone embedding

1. INTRODUCTION

In recent years, deep neural network (DNN) based automatic speech recognition (ASR) systems have been improved dramatically, which are, however, data-hungry. A well-trained DNN based ASR system for a single language usually requires hundreds to thousands of hours of transcribed speech data. Remarkably, there are more than 7100 languages in the world [1], and most of them are low-resourced languages, for which only limited transcribed speech data are available [2].

To advance ASR for low-resourced languages, multilingual and crosslingual speech recognition methods have long been developed, mainly for acoustic modeling [3, 4, 5, 6, 7, 8, 9] (More discussions in Section 2). Some end-to-end ASR models [10, 11, 12] fold the acoustic model (AM), pronunciation lexicon and language model (LM) into a single neural network, making the models being even more data-hungry and not suitable for low-resourced multilingual speech recognition, which is the main problem we hope to solve in this work[1].

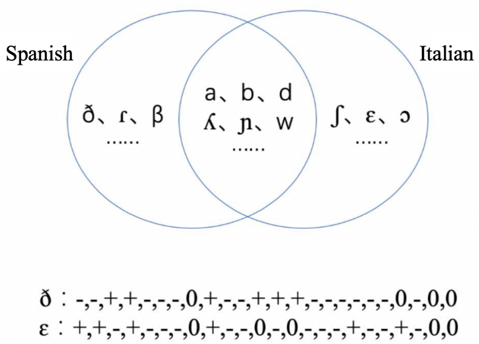

Spanish
Italian
ð、ɾ、β
......
a、b、d
ʎ、ɲ、w
......
ʃ、ε、ɔ
......
ð : -,-,+,+,-,-,-,0,+,-,-,+,+,+,-,-,-,-,-,-,0,-,0,0
ε : +,+,-,+,-,-,-,0,+,-,-,0,-,0,-,-,-,+,-,-,+,-,0,0

Fig. 1. Illustration of the connection between Spanish and Italian. As indicated by the intersection of the two circles, there are some common IPA phones used in both languages, which can be trained with more data from both languages. Notably, there also exist language-specific phones for each language. ð and ε only appears in Spanish and Italian respectively, and thus are not linked in the surface forms. The bottom show the phonological features for ð and ε, which share many common components. Phonological feature components are ordered as listed in Table 1.

Intuitively, the key to successful multilingual and crosslingual recognition is to promote the information sharing in multilingual training and maximize the knowledge transferring from the well trained multilingual model to the model for recognizing the utterances in the new language. To this end, a common practice is that similar sounds across languages are combined into one multilingual phone set. International Phonetic Alphabet (IPA), which classifies sounds based on phonetic knowledge, has been used to create a universal phone set [3, 8, 15]. Often phones are seen as being the "atoms" of speech. But it is now widely accepted in phonology that phones are decomposable into smaller, more fundamental units, sharable across all languages, called phonological distinctive features [16, 17]. Namely, phones can be represented by a set of phonological features, such as voicing, high, low (representing tongue position during vowels), round (for lip rounding), continuant (to distinguish sounds such as vowels and fricatives from stops) and so on, as shown in Table 1. As shown in Fig. 1, the use of phonological features potentially allow language-specific phones to remain linked — to "share statistical strength" in training. This is highly desirable, especially for zero-shot crosslingual speech recognition.

Phonological features (PFs) have been applied in multilingual and crosslingual speech recognition [18, 9]. Previous studies gener-

† Corresponding author. This work is supported by NSFC 61976122. Code is available at https://github.com/thu-spmi/CAT

[1] We suppose that text corpus and pronunciation lexicons or grapheme-to-phoneme (G2P) transducers [13, 14] are available.

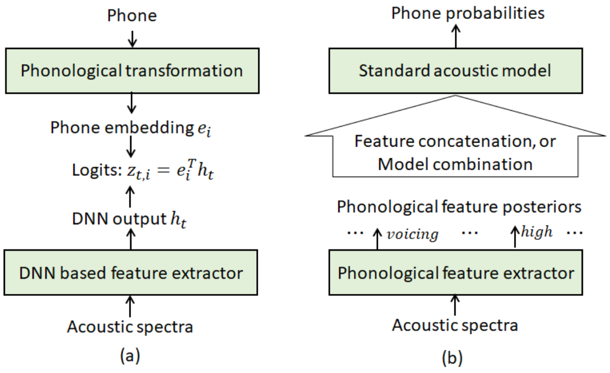

Phone
Phonological transformation
Phone embedding e_i
Logits: $z_{t,i} = e_i^T h_t$
DNN output h_t
DNN based feature extractor
Acoustic spectra
(a)
Phone probabilities
Standard acoustic model
Feature concatenation, or
Model combination
Phonological feature posteriors
$\cdots$ *voicing* $\cdots$ *high* $\cdots$
Phonological feature extractor
Acoustic spectra
(b)

Fig. 2. (a) Phonology driven phone embedding (top-down) and DNN based acoustic feature extraction (bottom-up) are joined to calculate the logits, which define the phone probabilities. (b) Traditional methods in using phonological features are purely bottom-up.

ally take a bottom-up approach and train a phonological feature extractor, often implemented by neural networks. Each training sample consists of a speech frame as the input and the canonical phonological feature components derived from the labeled phone as the target output values. Multiple neural networks are trained, depending on the partition of the phonological features. In the feature concatenation approach, the log posteriors of every phonological class are concatenated together and fed into the high-level acoustic model, which further predicts the phone probabilities [9]. Alternatively, in the model combination approach, the PF probabilities and the phone probabilities from a standard acoustic model are combined to calculate the acoustic score [18].

A drawback suffered by previous methods in using phonological features is that the acoustic-to-PF extraction in a bottom-up way is itself difficult, let alone the training of the phonological feature extractor needs segmented and labeled speech at the phone level. Moreover, previous methods do not provide a principled model to calculate the phone probabilities for unseen phones from the new language towards zero-shot crosslingual recognition. The parameters connecting to the unseen phones in the output layer of the DNN model are initialized either randomly [3] or in an ad-hoc way (taking a weighted average of the parameters of all the seen phones [9]).

In this paper, we propose a new approach to using phonological features for multilingual and crosslingual speech recognition. As illustrated in Fig. 2(a), our approach consists of phonology driven phone embedding (top-down) and DNN based acoustic feature extraction (bottom-up), which are joined to calculate the logits to define the phone probabilities. This is different from the pure bottom-up manner of the traditional methods in using phonological features, as sketched in Fig. 2(b).

Specifically, by using binary encoding of phonological features, we first obtain an encoding vector for each phone in the IPA table, which is referred to as the *phonological-vector*. Then, we apply linear or nonlinear transformation of the phonological-vector to obtain the phone embedding vector for each phone. This step is referred to as the phonological transformation (top-down). Next, we conduct bottom-up calculation on an acoustic DNN, viewed as a cascade of acoustic feature extractors. Finally, the extracted acoustic features and the phone embeddings are joined to calculate the phone (posterior) probabilities, which can be further used to calculate the CTC loss [19] or the CTC-CRF loss [20, 21]. This completes the definition of a multilingual acoustic model, which involves the Joining of Acoustics and Phonology and is thus called the JoinAP method. Remarkably, no inversion from acoustics to phonological features is required for speech recognition. Details about applying the JoinAP model to multilingual and crosslingual speech recognition are given in Section 3.

To evaluate the JoinAP model, a series of multilingual and crosslingual (both zero-shot and few-shot) speech recognition experiments are conducted on the CommonVoice dataset (involving German, French, Spanish and Italian) [22] and the AISHLL-1 dataset (Mandarin) [23]. The main findings are as follows.

- With JoinAP, we can develop a single acoustic model for multilingual speech recognition, which performs better than the traditional multilingual model (namely using flat phone embeddings[2]).
- In zero-shot crosslingual recognition, JoinAP with nonlinear phone embeddings outperforms both JoinAP with linear phone embeddings and the traditional model with flat phone embeddings significantly and consistently.
- In few-shot crosslingual recognition, using JoinAP with nonlinear phone embeddings still yields much better results than using flat phone embeddings; however, the superiority of nonlinear over linear for phone embeddings seems to be weakened, as there are more training data from the target languages.

2. RELATED WORK

In *multilingual speech recognition*, training data for a number of languages, often referred to as seen languages, are merged to train a multilingual AM. Multilingual training is found to outperform monolingual training, which trains the monolingual AMs separately for each seen language [4]. Such advantage is presumably due to the information sharing between seen languages in multilingual training. A common approach is to share the lower layers of the DNNs between languages, while the output layers are language specific [4, 5]. Another widely used approach is to extract the bottleneck features from the bottleneck layer of a multilingual DNN model, which are then used as input features to train the AM for the target language [6, 7].

Crosslingual speech recognition refers to recognizing utterances in a new language, which is unseen in training the multilingual AM. In the zero-shot setting, the multilingual AM is trained and directly used without any transcribed speech from the new, target language [3, 15, 24, 25]. Alternatively, in the few-shot setting, the multilingual AM can be further finetuned or adapted on limited transcribed speech from the new language [8, 9, 26]. Hopefully, knowledge can be transferred, by adapting a well-trained multilingual model to the new language, presumably because the multilingual model should learn some universal phonetic representations and the new language is similar to seen languages, more or less.

Earlier studies in multilingual and crosslingual recognition use context-dependent phone units, which leads to an explosion of units and also needs special care to handle context-dependent modeling across languages [27, 28]. There are recent attempts to use end-to-end ASR models such as CTC with monophones [8, 25] or end-to-end LF-MMI with biphones [29, 28] for multilingual and crosslingual recognition. Remarkably, the end-to-end CTC-CRF model,

[2] In the traditional multilingual model such as based on CTC or CTC-CRF, the weights of the final linear layer before softmax can be regarded as flat, unstructured phone embeddings. In contrast, JoinAP uses phonological-vector based, structured phone embeddings.

which is defined by a CRF (conditional random field) with CTC topology, has been shown to perform significantly better than CTC [20, 21]. Moreover, mono-phone CTC-CRF performs comparably to bi-phone end-to-end LF-MMI [29] and avoids context-dependent modeling with a simpler pipeline, which is particularly attractive for multilingual and crosslingual speech recognition.

When the phonological transformation is linear, our JoinAP model reduces to the model introduced in [25]. But in [25], only zero-shot crosslingual phone recognition is conducted and the model is developed still in a bottom-up way without the idea of joining acoustics (bottom-up) and phonology (top-down).

3. METHOD

This section first explains the definition and construction of phonological vectors. Then we describe the JoinAP method with linear and nonlinear phone embeddings. Finally, we introduce the CTC-CRF based ASR framework to use JoinAP.

3.1. Phonological-vector

Phonological (distinctive) features have been proposed as the basis of spoken language universals, in the sense that while the phones of a language vary, the set of phonological features does not and is the same for all languages. That is, phones can be constructed from a set of phonological features. As shown in Fig. 1, the use of phonological features potentially allow language-specific phones to remain linked, which could benefit the information sharing in multilingual training.

There are different phonological feature sets and phonological systems, among which one of the most popular systems is proposed by Chomsky and Halle in 1968 [16]. Phonological features are categorized into four classes: major class features, manner of articulation features, source features, cavity features. Each feature is marked as '+', '-' or '0'. '+' indicates the presence of that feature, '-' indicates the absence, and '0' means certain phone does not show such feature; for example, it is meaningless for a vowel to possess consonant features, so it will be marked as '0'. In our experiment, we employ PanPhon [30] to obtain the phonological features for IPA symbols. PanPhon uses a total of 24 phonological features. Table 1 gives examples of the feature specifications of some IPA phones, where all 24 features are listed.

Now each phone is described by 24 phonological features, and each feature can take '+', '-' or '0'. Further, we encode each phonological feature by a 2-bit binary vector. Taking the feature "round" as an example, the first bit indicates whether it is "round+" and the second bit indicates "round-". Therefore, if the "round" feature takes '+', the 2-bit vector will be "10"; if the "round" feature is '-', the 2-bit vector will be "01"; if the "round" feature takes '0', the 2-bit vector will be "00". In this way, we can represent the phonological features by a 48-bit vector. Additionally, acoustic training (e.g., based on CTC-CRF) introduces 3 special extra tokens (`<blk>`, `<spn>` and `<nsn>`), so we further add another 3 bits to encode the three special tokens in one-hot. In summary, we obtain a 51-bit encoding vector for each phone in the IPA table, which is referred to as the *phonological-vector*.

3.2. Phone embedding

Based on the phonological-vector representation of phones, we propose to join phonology driven phone embedding (top-down) and DNN based acoustic feature extraction (bottom-up) to calculate the logits, which further define the phone probabilities for ASR. This method, called JoinAP (Joining of Acoustics and Phonology), is different from the traditional bottom-up way to use phonological features for ASR (e.g., by building the phonological feature extactor) (See Fig. 2 for illustration).

Table 1. Phonological features of some IPA phones

Phonological feature	d	ɛ	ð	ə	i	ʥ	k^j
syllabic	-	+	-	+	+	-	-
sonorant	-	+	-	+	+	-	-
consonantal	+	-	+	-	-	+	+
continuant	-	+	+	+	+	-	-
delayed release	-	-	-	-	-	+	-
lateral	-	-	-	-	-	-	-
nasal	-	-	-	-	-	-	-
strident	0	0	0	0	0	0	0
voice	+	+	+	+	+	+	-
spread glottis	-	-	-	-	-	-	-
constricted glottis	-	-	-	-	-	-	-
anterior	+	0	+	0	0	-	-
coronal	+	-	+	-	-	+	-
distributed labial	-	0	+	0	0	+	0
labial	-	-	-	-	-	-	-
high	-	-	-	-	+	+	+
low	-	-	-	-	-	-	-
back	-	-	-	+	-	-	-
round	-	-	-	-	-	-	-
velaric	-	-	-	-	-	-	-
tense	0	-	0	-	+	0	0
long	-	-	-	-	-	-	-
hitone	0	0	0	0	0	0	0
hireg	0	0	0	0	0	0	0

In the traditional multilingual model such as based on CTC or CTC-CRF, one may view the acoustic DNN as a cascade of bottom-up feature extractors. At frame t, the DNN output $h_t \in \mathbb{R}^H$ could be viewed as the projection of speech into some abstract space, pertaining to the spoken phones. Before softmax computation to output phone probabilities, the final linear layer calculates the logits as follows:

$$z_{t,i} = e_i^T h_t \qquad (1)$$

where $e_i \in \mathbb{R}^H$ denotes the weight vector in the final linear layer and could be viewed as a (flat) phone embedding vector for phone i. For simplicity, we omit the bias in describing linear layers throughout the paper.

In JoinAP, we propose to apply linear or nonlinear transformation of the phonological-vector to obtain the phone embedding vector for each phone. This step is referred to as the phonological transformation (top-down) and explained as follows.

The JoinAP-Linear method. Given the phonological-vector $p_i \in \mathbb{R}^{51}$ for phone i, we apply linear transformation of p_i to define the embedding vector for phone i:

$$e_i = Ap_i \in \mathbb{R}^H \qquad (2)$$

where $A \in \mathbb{R}^{H \times 51}$ denotes the transformation matrix. The logits for calculating the phone (posterior) probabilities are still defined as in Eq. (1), which can be transparently used in the CTC-CRF based ASR framework (detailed later).

The JoinAP-Nonlinear method. The nonlinear method is similar to the linear one, except that we apply nonlinear transformation of p_i to define the embedding vector for phone i. In theory, multilayered neural networks could be used for phonological transformation. Here we consider to add one hidden layer as follows:

$$e_i = A_2 \sigma(A_1 p_i) \in \mathbb{R}^H \tag{3}$$

where A_1, A_2 denote the matrices of appropriate sizes, and $\sigma(\cdot)$ denote some nonlinear activation function (e.g. sigmoid). The logits for calculating the phone (posterior) probabilities are still defined as in Eq. (1).

3.3. CTC-CRF based ASR

In this section, we briefly explain the CTC-CRF based framework [20, 21] to use phone embeddings for ASR. Consider discriminative training with the objective to maximize the conditional likelihood [20]:

$$\mathcal{L}(\theta) = -\log p_\theta(\boldsymbol{l}|\boldsymbol{x}) \tag{4}$$

where $\boldsymbol{x} \triangleq (x_1, \cdots, x_T)$ is the speech feature sequence, $\boldsymbol{l} \triangleq (l_1, \cdots, l_L)$ is the phone-label sequence, and θ denotes the model parameters. Note that in speech recognition, $\boldsymbol{x}$ and $\boldsymbol{l}$ are in different lengths and not aligned. To handle this, a hidden state sequence $\boldsymbol{\pi} \triangleq (\pi_1, \cdots, \pi_T)$ and a map $\mathcal{B}(\cdot)$ for mapping $\boldsymbol{\pi}$ to $\boldsymbol{l}$ are introduced. The mapping function $\mathcal{B}$ removes consecutive repetitive labels and blanks in $\boldsymbol{\pi}$ to give $\boldsymbol{l}$. So the posterior of $\boldsymbol{l}$ is defined as:

$$p_\theta(\boldsymbol{l}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\boldsymbol{l})} p_\theta(\boldsymbol{\pi}|\boldsymbol{x}) \tag{5}$$

And the posterior of $\boldsymbol{\pi}$ is further defined by a conditional random field (CRF):

$$p_\theta(\boldsymbol{\pi}|\boldsymbol{x}) = \frac{\exp(\phi_\theta(\boldsymbol{\pi}, \boldsymbol{x}))}{\sum_{\boldsymbol{\pi}'} \exp(\phi_\theta(\boldsymbol{\pi}', \boldsymbol{x}))} \tag{6}$$

where $\phi_\theta(\boldsymbol{\pi}, \boldsymbol{x})$ denotes the potential function of the CRF, defined as:

$$\phi_\theta(\boldsymbol{\pi}, \boldsymbol{x}) = \log p(\boldsymbol{l}) + \sum_{t=1}^{T} \log p_\theta(\pi_t|\boldsymbol{x}) \tag{7}$$

where $\boldsymbol{l} = \mathcal{B}(\boldsymbol{\pi})$, and $p(\boldsymbol{l})$ is realized by an n-gram LM of labels. If $\log p(\boldsymbol{l})$ is omitted in Eq. (7), the potential function becomes self-normalized and CTC-CRF reduces to regular CTC. $p_\theta(\pi_t|\boldsymbol{x})$ represents the phone (posterior) probabilities, which are calculated by softmax from the logits $z_{t,i}$ in Eq. (1) as follows:

$$p_\theta(\pi_t = i|\boldsymbol{x}) = \frac{\exp(z_{t,i})}{\sum_j \exp(z_{t,j})}$$

Remarkably, regular CTC suffers from the conditional independence between the states in $\boldsymbol{\pi}$. In contrast, by incorporating $\log p(\boldsymbol{l})$ into the potential function in CTC-CRF, this drawback is naturally avoided. It has been shown that CTC-CRF outperforms regular CTC consistently on a wide range of benchmarks, and is on par with other state-of-the-art end-to-end models [20, 21, 31]. Moreover, CTC-CRF enjoys data-efficiency in training and works well with monophones [21], which are favorable for low-resourced multilingual and crosslingual speech recognition.

For decoding, we build a weighted finite state transducer (WFST), obtained by composing the CTC topology, pronunciation lexicon and word-level n-gram language model, and use WFST-based decoding.

Table 2. Datasets used in our experiments: the source, the number of IPA phone tokens in every language, the size of train, development and test sets in hours.

Language	Corpora	#Phones	Train	Dev	Test
German	CommonVoice	40	639.4	24.7	25.1
French	CommonVoice	57	465.2	21.9	23.0
Spanish	CommonVoice	30	246.4	24.9	25.6
Italian	CommonVoice	33	89.3	19.7	20.8
Polish	CommonVoice	46	93.2	5.2	6.1
Mandarin	AISHELL-1	96	150.9	18.1	10.0

4. EXPERIMENT

4.1. Experiment dataset and setup

Our experiments are conducted on two datasets, CommonVoice [22] and AISHELL-1 [23]. In our experiment, we use German, French, Spanish and Italian from CommonVoice to train the multilingual models. We carry out zero-shot and few-shot crosslingual experiments on Polish and Mandarin, where Polish comes from CommonVoice and Mandarin comes from AISHELL-1. Detailed data statistics are shown in Table 2.

We employ Phonetisaurus [14], a WFST-based G2P toolkit to generate IPA lexicons for the 6 languages in our experiments. All the monolingual phones were mapped to IPA symbols and we merged the phones from German, French, Spanish and Italian to create the universal phone set for multilingual training.

We use the CTC-CRF based ASR Toolkit - CAT [21], and will release the code in CAT when this work is published. Unless otherwise stated, the acoustic models used in our experiments are all based on CTC-CRF, and word-level N-gram language models are trained on the training transcripts for each language.

In all experiments, 40 dimension filter bank with delta and delta-delta features are extracted as input to the AM, which is 3 blocks of VGG layers followed by a 3-layer BLSTM with 1024 hidden size (namely $H = 2048$). A dropout probability of 50% is applied to the LSTM to prevent overfitting. During training, we use Adam as optimizer, and set initial learning rate as 1e-3. When the performance on development set stops improving, learning rate is adjusted to 1/10 of the previous one until it is less than 1e-5.

4.2. Experiment results

Our experiments are divided into 2 parts, multilingual and (zero-shot and few-shot) crosslingual. The multilingual models trained on the collection of German, French, Spanish and Italian data are tested on these 4 languages for multilingual experiments and on Polish and Mandarin for crosslingual experiments.

4.2.1. Multilingual experiment

Multilingual results are summarized in Table 3. Multilingual acoustic models are trained with 3 methods: the traditional method (namely using flat phone embeddings), JoinAP with linear phone embeddings, and JoinAP with nonlinear phone embeddings (the hidden layer size is 512). For each method, we test on the target language before and after fine-tuning over the data from the target language. Monolingual models on German, French, Spanish and Italian are trained separately for comparisons. The main observations are as follows.

Table 3. Word error rate (WER) results (%) for German, French, Spanish and Italian in the multilingual experiments. Multilingual models are trained with 3 methods: the traditional method using flat phone embeddings ("Flat-Phone"), JoinAP with linear phone embeddings ("JoinAP-Linear"), and JoinAP with nonlinear phone embeddings ("JoinAP-Nonlinear"). The multilingual models can be directly used without finetuning or with finetuning over the training data of the target languages. Monolingual models are trained separately for comparisons.

Language	Flat-Phone monolingual	Flat-Phone w/o finetuning	Flat-Phone finetuning	JoinAP-Linear w/o finetuning	JoinAP-Linear finetuning	JoinAP-Nonlinear w/o finetuning	JoinAP-Nonlinear finetuning
German	13.09	14.36	12.42	13.72	12.45	13.97	12.64
French	18.96	22.73	18.91	22.73	19.54	22.88	19.62
Spanish	15.11	13.93	13.06	13.93	13.19	14.10	13.26
Italian	24.57	25.97	21.77	25.85	21.70	24.06	20.29
Average	17.93	19.25	16.54	19.06	16.72	18.75	16.45

Table 4. About the intersections of the set of phones across languages. For each phone in a language, we count how many languages it appears and define this count to the language-degree of this phone, which may take from 1 to 4. The cell in column j denotes the number of those phones, whose language-degree is $j = 1, 2, 3, 4$.

Language-degree Language	4	3	2	1
German	18	6	8	8
French	18	6	7	26
Spanish	18	4	1	7
Italian	18	5	4	6

Table 5. WER results (%) for Polish in the crosslingual experiments. #Finetune denotes the amount of data used in finetuning (0 means zero-shot).

#Finetune	Flat-Phone	JoinAP-Linear	JoinAP-Nonlinear
0	33.15	35.73	31.80
10 minutes	8.70	7.50	8.10

Table 6. WER results (%) for Mandarin in the crosslingual experiments.

#Finetune	Flat-Phone	JoinAP-Linear	JoinAP-Nonlinear
0	97.10	89.51	88.41
1 hour	25.39	25.21	24.86

Without finetuning, the trained multilingual model can be directly used and works as a single model. In this case, on average, both JoinAP-Nonlinear and JoinAP-Linear perform better than Flat-Phone, and JoinAP-Nonlinear is the strongest. The average relative gain of JoinAP-Nonlinear over Flat-Phone is 3%. But notably, the detailed improvements of JoinAP-Linear over Flat-Phone and of JoinAP-Nonlinear over JoinAP-Linear are in fact language-dependent. For Italian, JoinAP-Nonlinear improves the most over JoinAP-Linear (7%), while for other languages, JoinAP-Linear performs slightly better. JoinAP-Linear performs better or equally well, compared to Flat-Phone. As observed in previous studies [15, 4], the performance differences between different multilingual training methods are affected by several factors, including the phonetic variety in this particular mix of multiple languages, data-scarce/data-rich for the target languages, etc.

After finetuing over the entire data from the target language, we obtain separate models for each target language. Similar to previous studies, the three multilingual training methods all significantly outperform monolingual trained models, on average. JoinAP-Nonlinear reduces the average WER by 8% against the monolingual models. The performance differences between the three multilingual methods themselves become smaller, presumably because the target training data are already rich enough to train models. On average, JoinAP-Nonlinear still performs the strongest.

To analyze, it is shown in Table 4 how the four languages are intersected with each other. We introduce the concept of the language-degree for a phone in multilingual training. Language-degree 4 means that the phone is shared by all the 4 languages, there are 18 such phones. Language-degree 1 means that the phone is language-unique, belonging to only one language. Italian has the smallest number of language-unique phones among the four languages. Many phones in Italian are also shared by other languages. Also note that Italian has the smallest amount of training data, as can be seen from Table 2. These may explain the most significant benefit for Italian from multilingual training. For Italian, finetuned multilingual JoinAP-Nonlinear reduces the WER by 17% again the monolingual Flat-Phone baseline. Also the gain by JoinAP-Nonlinear over JoinAP-Linear in Italian is also the largest (7% without finetuning, 6% after finetuning). On the other hand, French has the largest number of language-unique phones among the four languages, and the training data size is larger. This may explain the small improvement for French from multilingual training.

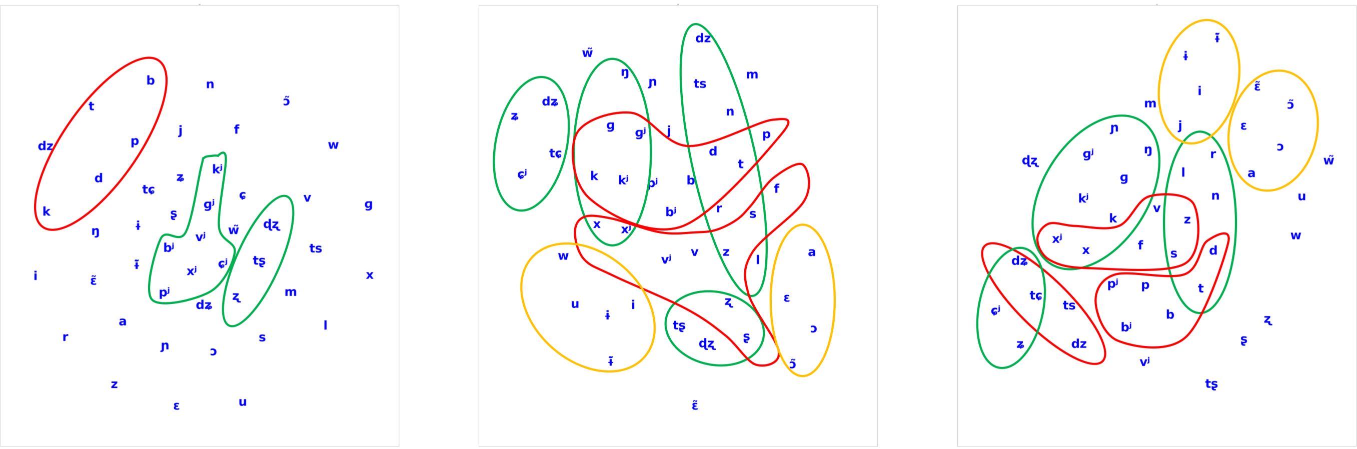
4.2.2. *Crosslingual experiment*

Crosslingual results for Polish and Mandarin are summarized in Table 5 and Table 6 respectively. The two languages are representative in how much the testing language is overlapped with the training languages, or say in the other way, how many unseen phones are in the testing language, as seen in Table 7. Polish represents the much overlapping setting, while Mandarin the less overlapping setting. The results for the two settings are different, as detailed below.

For Flat-Phone, in order to calculate the phone probabilities for unseen phones from the new languages (Polish and Mandarin), the parameters connecting to unseen phones in the output layer of the DNN model are initialized randomly. For JoinAP-Linear and JoinAP-Nonlinear, it is straightforward to calculate the phone embeddings for unseen phones, according to Eq. (2) and Eq. (3), once we obtain the phonological-vectors for those unseen phones.

In the zero-shot setting, for both Polish and Mandarin, JoinAP-Nonlinear outperforms both JoinAP-Linear and Phone-Flat significantly and consistently. The JoinAP-Linear performs worse than Flat-Phone in Polish. This is somewhat unexpected, which may reflect some instability of JoinAP-Linear.

In the few-shot setting, for both Polish and Mandarin, JoinAP-Nonlinear still yields much better results than Phone-Flat; however, the superiority of JoinAP-Nonlinear over JoinAP-Linear seems to be weakened, as there are more training data from the target languages. The JoinAP-Nonlinear performs better than JoinAP-Linear in Mandarin, while not in Polish. 10 minutes of transcribed speech in Polish may be rich enough for JoinAP-Linear to be well adapted, as indicated by the low WER. As can be seen from Table 7, the number of



(a)

(b)

(c)

Fig. 3. Visualization of Polish phone embeddings by t-SNE. (a) Flat phone embeddings, (b) JoinAP-Linear phone embeddings, (c) JoinAP-Nonlinear phone embeddings. They are obtained from the un-finetuned multilingual models in the zero-shot Polish experiment. Red circles indicate the consonants with the same manner of articulation, green circles indicate the consonants with the same place of articulation, and yellow circles indicate similar vowel height.

Table 7. Statistics about Polish and Mandarin, including the number of IPA phone tokens in every language, and the number of unseen phones.

Language	#Phones	#Unseen phones
Polish	46	18
Mandarin	96	79

unseen phones in Polish is far less than in Mandarin. This may also explains the good performance of JoinAP-Linear in Polish. Remarkably, under a large amount of finetuning data for a testing language that is much overlapped with training languages, say, 1 hour for Polish, the performances of different models will tend to saturate and become less differed. So we use 10-min finetuning data for Polish for comparing different models. 10-min for Mandarin will yield results similar to zero-shot, so we use 1-hour for Mandarin few-shot.

To further understand our phonological-vector based phone embeddings, we apply t-SNE [32] to draw the 2048-dimensional phone embeddings on a 2-dimensional map. Fig. 3 shows the maps of the 46 phones in Polish, obtained from the un-finetuned multilingual models. It seems that we can hardly find many sensible groupings for flat phone embeddings from Fig. 3(a). But for JoinAP-Linear and JoinAP-Nonlinear phone embeddings, the maps reflect more notable groupings, where similar phones are found to gather together in the maps. We use red, green and yellow circles to indicate the phones with the same manner of articulation, place of articulation and vowel height respectively. Their detailed IPA features are listed in Table 8. The figures clearly show that the JoinAP based phone embeddings indeed carry phonological information, which could help zero-shot learning. Moreover, it can be seen that the vowels are located in the top-right corner in Fig. 3(c), while the vowels are separated in two corners in Fig. 3(b). And the within-class scattering of (b) seems to be larger than (c). These observations could reflect the superority of JoinAP-nonlinear over JoinAP-Linear.

5. CONCLUSION

In this work, we propose the JoinAP method to join phonology driven phone embedding (top-down) and DNN based acoustic feature extraction (bottom-up). We apply linear or nonlinear transformation of phonological-vectors to obtain phone embeddings, and compare to the traditional method using flat phone embeddings. In the multilingual and crosslingual experiments, JoinAP-Nonlinear generally performs better than JoinAP-Linear and the traditional flat-phone method on average. The improvements are generally the most significant for those target languages such as Italian in our multilingual experiments and Polish in our zero-shot crosslingual experiments, due to their data-scarce and high language-degrees of their phones (i.e., being well shared by other languages), and become weak for those target languages when they become data-rich such as French in our multilingual experiments and Polish in our few-shot crosslingual experiments. In summary, the JoinAP method provides a principled approach to multilingual and crosslingual speech recognition. Some promising directions include exploring DNN based phonological transformation, and pretraining over increasing number of languages.

Table 8. Detailed explanation of Fig. 3. We list the IPA phones in the colored circles from different methods.

Method	Color	Feature	Phones
Flat	Green	Retroflex	ʐ ɖʐ tʂ
		Palatalized	g^j k^j ɕ^j v^j b^j x^j p^j
	Red	Fricative	b t p d k
Linear	Green	Alveolo-palatal	ʑ ɕ ʥ ʨ
		Velar	ŋ g k x g^j k^j x^j
		Alveolar	ʣ ʦ n d t r s z l
		Retroflex	ʂ ʐ ɖʐ tʂ
	Red	Plosive	x v x^j v^j z s f ʐ ʂ
		Fricative	g k p g^j k^j p^j b b^j t d
	Yellow	Close	i u w ɨ ɨ̃
		Open/Open-mid	a ɛ ɔ ɔ̃
Nonlinear	Green	Alveolo-palatal	ʑ ɕ ʥ ʨ
		Velar	ŋ g k x g^j k^j x^j
		Alveolar	n d t r s z l
	Red	Affricate	ʥ ʨ ʣ ʦ
		Plosive	x v x^j z s f
		Fricative	p p^j b b^j t d
	Yellow	Close	i j ɨ ɨ̃
		Open/Open-mid	a ɛ ɛ̃ ɔ ɔ̃

6. REFERENCES

[1] M. Paul Lewis, Ed., *Ethnologue: Languages of the World*, SIL International, Dallas, TX, USA, sixteenth edition, 2009.

[2] Tanja Schultz and Tim Schlippe, "GlobalPhone: Pronunciation dictionaries in 20 languages," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, 2014, pp. 337–341.

[3] T. Schultz and A. Waibel, "Multilingual and crosslingual speech recognition," in *Proc. DARPA Workshop on Broadcast News Transcription and Understanding*, 1998, pp. 259–262.

[4] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *Proc. ICASSP*, 2013, pp. 8619–8623.

[5] Dongpeng Chen and Brian Kan-Wing Mak, "Multitask learning of deep neural networks for low-resource speech recognition," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 23, no. 7, pp. 1172–1183, July 2015.

[6] S. Thomas, S. Ganapathy, and H. Hermansky, "Cross-lingual and multi-stream posterior features for low resource lvcsr systems," in *Proc. INTERSPEECH*, 2010.

[7] Enno Hermann and Sharon Goldwater, "Multilingual bottleneck features for subword modeling in zero-resource languages," in *Proc. INTERSPEECH*, 2018.

[8] Sibo Tong, Philip Garner, and Herve Bourlard, "Multilingual training and cross-lingual adaptation on ctc-based acoustic model," *Speech Communication*, vol. 104, 11 2017.

[9] Sibo Tong, Philip N. Garner, and Hervé Bourlard, "Fast language adaptation using phonological information," in *Proc. INTERSPEECH*, 2018, pp. 2459–2463.

[10] Shinji Watanabe, Takaaki Hori, and John R. Hershey, "Language independent end-to-end architecture for joint language identification and speech recognition," in *Proc. ASRU*, 2017, pp. 265–271.

[11] Shubham Toshniwal, Tara Sainath, Ron Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao, "Multilingual speech recognition with a single end-to-end model," in *Proc. ICASSP*, 2018, pp. 4904–4908.

[12] Bo Li, Yu Zhang, T. Sainath, Yonghui Wu, and William Chan, "Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes," in *Proc. ICASSP*, 2019, pp. 5621–5625.

[13] Mark Hasegawa-Johnson, Leanne Rolston, Camille Goudeseune, Gina-Anne Levow, and Katrin Kirchhoff, "Grapheme-to-phoneme transduction for cross-language asr," in *Statistical Language and Speech Processing(SLSP)*, 2020, pp. 3–19.

[14] Josef R. Novak, Nobuaki Minematsu, and Keikichi Hirose, "WFST-based grapheme-to-phoneme conversion: Open source tools for alignment, model-building and decoding," in *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, 2012, pp. 45–49.

[15] Piotr Zelasko, Laureano Moro-Velázquez, Mark Hasegawa-Johnson, Odette Scharenborg, and Najim Dehak, "That sounds familiar: An analysis of phonetic representations transfer across languages," in *Proc. INTERSPEECH*, 2020.

[16] Noam. Chomsky and Morris Halle, *The sound pattern of English*, Harper Row New York, 1968.

[17] Simon King and Paul Taylor, "Detection of phonological features in continuous speech using neural networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 333–353, 2000.

[18] Sebastian Stüker, Florian Metze, Tanja Schultz, and Alex Waibel, "Integrating multilingual articulatory features into speech recognition," in *Proc. Eurospeech*, 2003, pp. 1033–1036.

[19] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006, p. 369–376.

[20] Hongyu Xiang and Zhijian Ou, "Crf-based single-stage acoustic modeling with ctc topology," in *Proc. ICASSP*, 2019, pp. 5676–5680.

[21] Keyu An, Hongyu Xiang, and Zhijian Ou, "Cat: A ctc-crf based asr toolkit bridging the hybrid and the end-to-end approaches towards data efficiency and low latency," in *Proc. INTERSPEECH*, 2020.

[22] Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber, "Common voice: A massively-multilingual speech corpus," in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020, pp. 4218–4222.

[23] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases & Speech I/O Systems & Assessment*, 2017.

[24] Siyuan Feng, Piotr Żelasko, Laureano Moro-Velázquez, Ali Abavisani, Mark Hasegawa-Johnson, Odette Scharenborg, and Najim Dehak, "How phonotactics affect multilingual and zero-shot asr performance," in *Proc. ICASSP*, 2021, pp. 7238–7242.

[25] Xinjian Li, Siddharth Dalmia, David R. Mortensen, Juncheng Li, Alan W. Black, and Florian Metze, "Towards zero-shot learning for automatic phonemic transcription," in *Proc. AAAI*, 2020, pp. 8261–8268.

[26] Xinjian Li, Siddharth Dalmia, Juncheng Li, Matthew Lee, Patrick Littell, Jiali Yao, Antonios Anastasopoulos, David R Mortensen, Graham Neubig, Alan W Black, et al., "Universal phone recognition with a multilingual allophone system," in *Proc. ICASSP*, 2020.

[27] A. Zgank, B. Imperl, F. T. Johansen, Z. Kacic, and B. Horvat, "Crosslingual speech recognition with multilingual acoustic models based on agglomerative and tree-based triphone clustering.," in *European Conference on Eurospeech Scandinavia*, 2001.

[28] S. Tong, P. N. Garner, and H. Bourlard, "An investigation of multilingual asr using end-to-end lf-mmi," in *Proc. ICASSP*, 2019.

[29] Hossein Hadian, Hossein Sameti, Daniel Povey, and Sanjeev Khudanpur, "End-to-end speech recognition using lattice-free mmi," in *Proc. INTERSPEECH*, 2018, pp. 12–16.

[30] David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori Levin, "PanPhon: A resource for

mapping IPA segments to articulatory feature vectors," in *Proc. the 26th International Conference on Computational Linguistics (COLING)*, 2016.

[31] Huahuan Zheng, Keyu An, and Zhijian Ou, "Efficient neural architecture search for end-to-end speech recognition via straight-through gradients," in *Proc. IEEE SLT*, 2021, pp. 60–67.

[32] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.