Computer Speech & Language 65 (2021) 101098

Contents lists available at ScienceDirect

Computer Speech & Language

journal homepage: www.elsevier.com/locate/csl

Multilingual and unsupervised subword modeling for zero-resource languages

Enno Hermann[a,1]*, Herman Kamper[b], Sharon Goldwater[a]

[a] School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, UK
[b] Department of E&E Engineering, Stellenbosch University, Stellenbosch 7600, South Africa

ARTICLE INFO

Article History:
Received 29 April 2019
Revised 3 February 2020
Accepted 19 March 2020
Available online 17 April 2020

Keywords:
Multilingual bottleneck features
Subword modeling
Unsupervised feature extraction
Zero-resource speech technology

ABSTRACT

Subword modeling for zero-resource languages aims to learn low-level representations of speech audio without using transcriptions or other resources from the target language (such as text corpora or pronunciation dictionaries). A good representation should capture phonetic content and abstract away from other types of variability, such as speaker differences and channel noise. Previous work in this area has primarily focused unsupervised learning from target language data only, and has been evaluated only intrinsically. Here we directly compare multiple methods, including some that use only target language speech data and some that use transcribed speech from other (non-target) languages, and we evaluate using two intrinsic measures as well as on a downstream unsupervised word segmentation and clustering task. We find that combining two existing target-language-only methods yields better features than either method alone. Nevertheless, even better results are obtained by extracting target language bottleneck features using a model trained on other languages. Cross-lingual training using just one other language is enough to provide this benefit, but multilingual training helps even more. In addition to these results, which hold across both intrinsic measures and the extrinsic task, we discuss the qualitative differences between the different types of learned features.

1. Introduction

Recent years have seen increasing interest in speech technology for "zero-resource" languages, where systems must be developed for a target language without using transcribed data or other hand-curated resources from that language. Such systems could potentially be applied to tasks such as endangered language documentation or query-by-example search for languages without a written form. One challenge for these systems, highlighted by the Zero Resource Speech Challenge (ZRSC) shared tasks of 2015 (Versteegh et al., 2015) and 2017 (Dunbar et al., 2017), is to improve subword modeling, i.e., to extract or learn speech features from the target language audio. Good features should be more effective at discriminating between linguistic units, e.g. words or subwords, while abstracting away from factors such as speaker identity and channel noise.

The ZRSCs were motivated largely by questions in artificial intelligence and human perceptual learning, and focused on approaches where no transcribed data from *any* language is used. Yet from an engineering perspective it also makes sense to explore how training data from higher-resource languages can be used to improve speech features in a zero-resource language.

This paper explores several methods for improving subword modeling in zero-resource languages, either with or without the use of labeled data from other languages. Although the individual methods are not new, our work provides a much more thorough

*Corresponding author.
E-mail addresses: enno.hermann@idiap.ch (E. Hermann), kamperh@sun.ac.za (H. Kamper), sgwater@inf.ed.ac.uk (S. Goldwater).
[1] Present address: Idiap Research Institute, Rue Marconi 19, 1920 Martigny, Switzerland.

https://doi.org/10.1016/j.csl.2020.101098
0885-2308/

empirical evaluation of these methods compared to the existing literature. We experiment with each method both alone and in combinations not tried before, and provide results across a range of target languages, evaluation measures, and tasks.

We start by evaluating two methods for feature extraction that are trained using (untranscribed) target language data only: traditional vocal tract length normalization (VTLN) and the correspondence autoencoder (cAE) proposed more recently by Kamper et al. (2015). The cAE learns to abstract away from signal noise and variability by training on pairs of speech segments extracted using an unsupervised term discovery (UTD) system—i.e., pairs that are likely to be instances of the same word or phrase. We confirm previous work showing that cAE features outperform Mel-frequency cepstral coefficients (MFCCs) on a word discriminability task, although we also show that this benefit is not consistently better than that of simply applying VTLN. More interestingly, however, we find that applying VTLN to the input of the cAE system improves the learned features considerably, leading to better performance than either method alone. These improvements indicate that cAE and VTLN abstract over different aspects of the signal, and suggest that VTLN might also be a useful preprocessing step in other recent neural-network-based unsupervised feature-learning methods.

Next, we explore how multilingual annotated data can be used to improve feature extraction for a zero-resource target language. We train multilingual bottleneck features (BNFs) on between one and ten languages from the GlobalPhone collection and evaluate on six other languages (simulating different zero-resource targets). We show that training on more languages consistently improves performance on word discrimination, and that the improvement is not simply due to more training data: an equivalent amount of data from one language fails to give the same benefit. In fact, we observe the largest gain in performance when adding the second training language, which is already better than adding three times as much data from the same language. Moreover, when compared to our best results from training unsupervised on target language data only, we find that BNFs trained on just a single other language already outperform the target-language-only training, with multilingual BNFs doing better by a wide margin.

Although multilingual training outperforms unsupervised target-language training, it could still be possible to improve on the multilingual BNFs by using them as inputs for further target-language training. To test this hypothesis, we passed the multilingual BNFs as input to the cAE. When trained with UTD word pairs, we found no benefit to this method. However, training with manually labeled word pairs did yield benefits, suggesting that this type of supervision can help improve on the BNFs if the word pairs are sufficiently high-quality.

The results above were presented as part of an earlier conference version of this paper (Hermann and Goldwater, 2018). Here, we expand upon that work in several ways. First, we include new results on the corpora and evaluation measures used in the ZRSC, to allow more direct comparisons with other work. In doing so, we also provide the first set of results on identical systems evaluated using both the same-different and ABX evaluation measures. This permits the two measures themselves to be better compared. Finally, we provide both a qualitative analysis of the differences between the different features we extract, and a quantitative evaluation on the downstream target-language task of unsupervised full-coverage speech segmentation and clustering using the system of Kamper et al. (2017). This is the first time that multilingual features are used in such a system, which performs a complete segmentation of input speech into hypothesized words. As in our intrinsic evaluations, we find that the multilingual BNFs consistently outperform the best unsupervised cAE features, which in turn outperform or do similarly to MFCCs.

2. Unsupervised training, target language only

We start by investigating how unlabeled data from the target language alone can be used for unsupervised subword modeling and how speaker normalisation with VTLN can improve such systems. Below we first review related work and provide a brief introduction to the cAE and VTLN methods. We then describe our experiments directly comparing these methods, both alone and in combination.

2.1. Background and motivation

Various approaches have been applied to the problem of unsupervised subword modeling. Some methods work in a strictly bottom-up fashion, for example by extracting posteriorgrams from a (finite or infinite) Gaussian mixture model trained on the unlabeled data (Zhang and Glass, 2009; Huijbregts et al., 2011; Chen et al., 2015), or by using neural networks to learn representations using autoencoding (Zeiler et al., 2013; Badino et al., 2014; 2015) or other loss functions (Synnaeve and Dupoux, 2016). Other methods incorporate weak top-down supervision by first extracting pairs of similar word- or phrase-like units using unsupervised term detection, and using these to constrain the representation learning. Examples include the correspondence autoencoder (cAE) (Kamper et al., 2015) and ABNet (Synnaeve et al., 2014). Both aim to learn representations that make similar pairs even more similar; the ABNet additionally tries to make different pairs more different.

In this work we use the cAE in our experiments on unsupervised representation learning, since it performed well in the 2015 ZRSC, achieved some of the best-reported results on the same-different task (which we also consider), and has readily available code. As noted above, the cAE attempts to normalize out non-linguistic factors such as speaker, channel, gender, etc., by using top-down information from pairs of similar speech segments. Extracting cAE features requires three steps, as illustrated in Fig. 1. First, an unsupervised term discovery (UTD) system is applied to the target language to extract pairs of speech segments that are likely to be instances of the same word or phrase. Each pair is then aligned at the frame level using dynamic time warping (DTW), and pairs of aligned frames are presented as the input $\mathbf{x}$ and target output $\mathbf{x}'$ of a deep neural network (DNN). After training, a middle layer $\mathbf{y}$ is used as the learned feature representation.

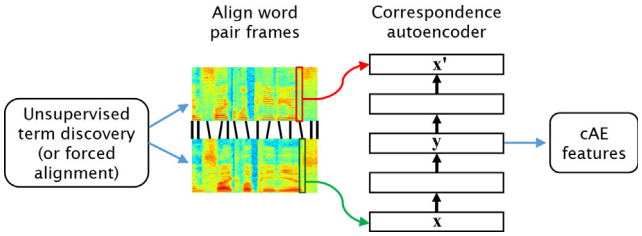


Fig. 1. Correspondence autoencoder training procedure (see Section 2.1).

The cAE and other unsupervised methods described above implicitly aim to abstract away from speaker variability, and indeed they succeed to some extent in doing so (Kamper et al., 2017). Nevertheless, they provide less explicit speaker adaptation than standard methods used in supervised automatic speech recognition (ASR), such as feature-space maximum likelihood linear regression (fMLLR) (Gales, 1998), learning hidden unit contributions (LHUC) (Swietojanski et al., 2016) or i-vectors (Saon et al., 2013). Explicit speaker adaptation seems to have attracted little attention until recently (Zeghidour et al., 2016; Heck et al., 2016; Tsuchiya et al., 2018) in the zero-resource community, perhaps because most of the standard methods assume transcribed data is available.

Nevertheless, recent work suggests that at least some of these methods may be applied effectively even in an unsupervised setting. In particular, Heck et al. (2017, 2018b) won the ZRSC 2017 using a typical ASR pipeline with speaker adaptive fMLLR and other feature transforms. They adapted these methods to the unsupervised setting by first obtaining phone-like units with the Dirichlet Process Gaussian mixture model (DPGMM), an unsupervised clustering technique, and then using the cluster assignments as unsupervised phone labels during ASR training.

In this work we instead consider a very simple feature-space adaptation method, vocal tract length normalization (VTLN), which normalizes a speaker's speech by warping the frequency-axis of the spectra. VTLN models are trained using maximum likelihood estimation under a given acoustic model—here, an unsupervised GMM. Warp factors can then be extracted for both the training data and for unseen data.

Although VTLN has recently been used by a few zero-resource speech systems (Chen et al., 2015; Heck et al., 2017; 2018b), its impact in these systems is unclear because there is no comparison to a baseline without VTLN. Chen et al. (2017) did precisely such a comparison and showed that applying VTLN to the input of their unsupervised feature learning method improved its results in a phoneme discrimination task, especially in the cross-speaker case. However, we don't know whether other feature learning methods are similarly benefited by VTLN, nor even how VTLN on its own performs in comparison to more recent methods. Thus, our first set of experiments is designed to answer these questions by evaluating the benefits of using VTLN and cAE learning, both on their own and in combination.

2.2. Experimental setup

We use the GlobalPhone corpus of speech read from news articles (Schultz et al., 2013). We chose 6 languages from different language families as **zero-resource** languages on which we evaluate the new feature representations. That means our models do not have any access to the transcriptions of the training data, although transcriptions still need to be available to run the evaluation. The selected languages and dataset sizes are shown in Table 1. Each GlobalPhone language has recordings from around 100 speakers, with 80% of these in the training sets and no speaker overlap between training, development, and test sets.

For baseline features, we use Kaldi (Povey et al., 2011) to extract MFCCs+Δ+$\Delta\Delta$ and PLPs+Δ+$\Delta\Delta$ with a window size of 25 ms and a shift of 10 ms, and we apply per-speaker cepstral mean normalization. We also evaluated MFCCs and PLPs with VTLN. The acoustic model used to extract the warp factors was a diagonal-covariance GMm with 1024 components. A single GMM was trained unsupervised on each language's training data.

Table 1
Zero-resource languages, dataset sizes in hours.

Language		Train	Dev	Test
GlobalPhone				
Croatian	(HR)	12.1	2.0	1.8
Hausa	(HA)	6.6	1.0	1.1
Mandarin	(ZH)	26.6	2.0	2.4
Spanish	(ES)	17.6	2.1	1.7
Swedish	(SV)	17.4	2.1	2.2
Turkish	(TR)	13.3	2.0	1.9
ZRSC 2015				
Buckeye English	(EN-B)			5
Xitsonga	(TS)			2.5

To train the cAE, we obtained UTD pairs using a freely available UTD system[1] (Jansen and Van Durme, 2011) and extracted 36k word pairs for each target language. Published results with this system use PLP features as input, and indeed our preliminary experiments confirmed that MFCCs did not work as well. We therefore report results using only PLP or PLP+VTLN features as input to UTD. Following Renshaw et al. (2015) and Kamper et al. (2015), we train the cAE model[2] by first pre-training an autoencoder with eight 100-dimensional layers and a final layer of size 39 layer-wise on the entire training data for 5 epochs with a learning rate of 2.5×10^{-4}. We then fine-tune the network with same-word pairs as weak supervision for 60 epochs with a learning rate of 2.5×10^{-5}. Frame pairs are presented to the cAE using either MFCC, MFCC+VTLN, or BNF representation, depending on the experiment (preliminary experiments indicated that PLPs performed worse than MFCCs, so MFCCs are used as the stronger baseline). Features are extracted from the final hidden layer of the cAE as shown in Fig. 1.

To provide an upper bound on cAE performance, we also report results using *gold standard* same-word pairs for cAE training. As in Kamper et al. (2015), Jansen et al. (2013), Yuan et al. (2017b), we force-align the target language data and extract all the same-word pairs that are at least 5 characters and 0.5 s long (between 89k and 102k pairs for each language).

2.3. Evaluation

All experiments in this section are evaluated using the same-different task (Carlin et al., 2011), which tests whether a given speech representation can correctly classify two speech segments as having the same word type or not. For each word pair in a pre-defined set S the DTW cost between the acoustic feature vectors under a given representation is computed. Two segments are then considered a match if the cost is below a threshold. Precision and recall at a given threshold τ are defined as

$$P(\tau) = \frac{M_{\text{SW}}(\tau)}{M_{\text{all}}(\tau)}, \quad R(\tau) = \frac{M_{\text{SWDP}}(\tau)}{|S_{\text{SWDP}}|}$$

where M is the number of same-word (SW), same-word different-speaker (SWDP) or all discovered matches at that threshold and $|S_{\text{SWDP}}|$ is the number of actual SWDP pairs in S. We can compute a precision-recall curve by varying τ. The final evaluation metric is the average precision (AP) or the area under that curve. We generate evaluation sets of word pairs for the GlobalPhone development and test sets from all words that are at least 5 characters and 0.5 s long, except that we now also include different-word pairs.

Previous work (Carlin et al., 2011; Kamper et al., 2015) calculated recall with all SW pairs for easier computation because their test sets included a negligible number of same-word same-speaker (SWSP) pairs. In our case the smaller number of speakers in the GlobalPhone corpora results in up to 60% of SW pairs being from the same speaker. We therefore always explicitly compute the recall only for SWDP pairs to focus the evaluation of features on their speaker invariance.

2.4. Results and discussion

Table 2 shows AP results on all target languages for cAE features learned using raw features as input (as in previous work) and for cAE features learned using VTLN-adapted features as input to either the UTD system, the cAE, or both. Baselines are raw MFCCs, or MFCCs with VTLN. MFCCs with VTLN have not previously been compared to more recent unsupervised subword modeling methods, but as our results show, they are a much stronger baseline than MFCCs alone. Indeed, they are nearly as good as cAE features (as trained in previous work). However, we obtain much better results by applying VTLN to both the cAE and UTD input features (MFCCs and PLPs, respectively). Individually these changes each result in substantial improvements that are consistent across all 6 languages, and applying VTLN at both stages helps further. Indeed, applying VTLN is beneficial even when using gold pairs as cAE input, although to a lesser degree.

So, although previous studies have indicated that cAE training and VTLN are helpful individually, our experiments provide further evidence and quantification of those results. In addition, we have shown that combining the two methods leads to further improvements, suggesting that cAE training and VTLN abstract over different aspects of the speech signal and should be used together. The large gains we found with VTLN, and the fact that it was part of the winning system in the 2017 ZRSC, suggest that it is also likely to help in combination with other unsupervised subword modeling methods.

3. Supervision from high-resource languages

Next we investigate how labeled data from high-resource languages can be used to obtain improved features on a target zero-resource language for which no labeled data is available. Furthermore, are there benefits in deriving this weak supervision from multiple languages?

3.1. Background and motivation

There is considerable evidence that BNFs extracted using a multilingually trained DNN can improve ASR for target languages with just a few hours of transcribed data (Veselý et al., 2012; Vu et al., 2012; Thomas et al., 2012; Cui et al., 2015; Alumäe et al., 2016).

[1] https://github.com/arenjansen/ZRTools.
[2] https://github.com/kamperh/speech_correspondence.

Table 2
Average precision scores on the same-different task (dev sets), showing the effects of applying VTLN to the input features for the UTD and/or cAE systems. cAE input is either MFCC or MFCC+VTLN. Topline results (rows 5–6) train cAE on gold standard pairs, rather than UTD output. Baseline results (final rows) directly evaluate acoustic features without UTD/cAE training. Best unsupervised result in bold.

UTD input	cAE input	ES	HA	HR	SV	TR	ZH
cAE systems:							
PLP	MFCC	28.6	39.9	26.9	22.2	25.2	20.4
PLP	MFCC+VTLN	46.2	48.2	36.3	37.9	31.4	35.7
PLP+VTLN	MFCC	40.4	45.7	35.8	25.8	25.9	26.9
PLP+VTLN	MFCC+VTLN	**51.5**	**52.9**	**39.6**	**42.9**	**33.4**	**44.4**
Gold pairs	MFCC	65.3	65.2	55.6	52.9	50.6	60.5
Gold pairs	MFCC+VTLN	68.9	70.1	57.8	56.9	56.3	69.5
Baseline: MFCC		18.3	19.6	17.6	12.3	16.8	18.3
Baseline: MFCC+VTLN		27.4	28.4	23.2	20.4	21.3	27.7

However, there has been little work so far exploring supervised multilingual BNFs for target languages with no transcribed data at all. Yuan et al. (2016) and Renshaw et al. (2015) trained *monolingual* BNF extractors and showed that applying them cross-lingually improves word discrimination in a zero-resource setting. Yuan et al. (2017a) and Chen et al. (2017) trained a multilingual DNN to extract BNFs for a zero-resource task, but the DNN itself was trained on untranscribed speech: an unsupervised clustering method was applied to each language to obtain phone-like units, and the DNN was trained on these unsupervised phone labels.

We know of only two previous studies of supervised multilingual BNFs for zero-resource speech tasks. In the first, Yuan et al. (2017b) trained BNFs on either Mandarin, Spanish or both, and used the trained DNNs to extract features from English (simulating a zero-resource language). On a query-by-example task, they showed that BNFs always performed better than MFCCs, and that bilingual BNFs performed as well or better than monolingual ones. Further improvements were achieved by applying weak supervision in the target language using a cAE trained on English word pairs. However, the authors did not experiment with more than two training languages, and only evaluated on English.

In the second study, Shibata et al. (2017) built multilingual systems using either seven or ten high-resource languages, and evaluated on the three "development" and two "surprise" languages of the ZRSC 2017. However, they included transcribed training data from four out of the five evaluation languages, so only one language's results (Wolof) were truly zero-resource.

Our experiments therefore aim to evaluate on a wider range of target languages, and to explore the effects of both the *amount* of labeled data, and the *number of languages* from which it is obtained.

3.2. Experimental setup

We picked another 10 languages (different from the target languages described in Section 2.2) with a combined 198.3 h of speech from the GlobalPhone corpus. We consider these as **high-resource** languages, for which transcriptions are available to train a supervised ASR system. The languages and dataset sizes are listed in Table 3. We also use the English Wall Street Journal (WSJ) corpus (Paul and Baker, 1992) which is comparable to the GlobalPhone corpus. It contains a total of 81 h of speech, which we either use in its entirety or from which we use a 15 h subset; this allows us to compare the effect of increasing the amount of data for one language with training on similar amounts of data but from different languages.

Table 3
High-resource languages, dataset sizes in hours.

Language		Train
Bulgarian	(BG)	17.1
Czech	(CS)	26.8
French	(FR)	22.8
German	(DE)	14.9
Korean	(KO)	16.6
Polish	(PL)	19.4
Portuguese	(PT)	22.8
Russian	(RU)	19.8
Thai	(TH)	21.2
Vietnamese	(VI)	16.9
English81 WSJ	(EN)	81.3
English15 WSJ		15.1

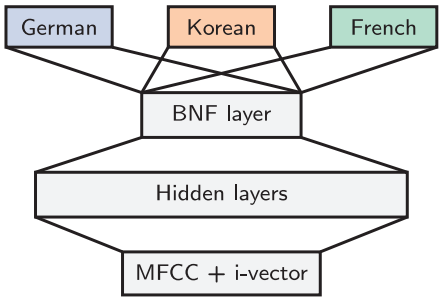


Fig. 2. Multilingual acoustic model training architecture. All layers are shared between languages except for the language-specific output layers at the top.

Supervised models trained on these high-resource languages are evaluated on the same set of zero-resource languages as in Section 2. Transcriptions of the latter are still never used during training.

For initial monolingual training of ASR systems for the high-resource languages, we follow the Kaldi recipes for the GlobalPhone and WSJ corpora and train a subspace GMM (SGMM) system for each language to get initial context-dependent state alignments; these states serve as targets for DNN training.

For multilingual training, we closely follow the existing Kaldi recipe for the Babel corpus (Trmal et al., 2017). We train a time-delay neural network (TDNN) (Waibel et al., 1989; Lang et al., 1990; Peddinti et al., 2015) with block softmax (Grézl et al., 2014), i.e. all hidden layers are shared between languages, but there is a separate output layer for each language. For each training instance only the error at the corresponding language's output layer is used to update the weights. This architecture is illustrated in Fig. 2. The TDNN has six 625-dimensional hidden layers[3] followed by a 39-dimensional bottleneck layer with ReLU activations and batch normalization. Each language then has its own 625-dimensional affine and a softmax layer. The inputs to the network are 40-dimensional MFCCs with all cepstral coefficients to which we append i-vectors for speaker adaptation. The network is trained with stochastic gradient descent for 2 epochs with an initial learning rate of 10^{-3} and a final learning rate of 10^{-4}.

In preliminary experiments we trained a separate i-vector extractor for each different sized subset of training languages. However, results were similar to training on the pooled set of all 10 high-resource languages, so for expedience we used the 100-dimensional i-vectors from this pooled training for all reported experiments. The i-vectors for the zero-resource languages are obtained from the same extractor. This allows us to also apply speaker adaptation in the zero-resource scenario and to draw a fair comparison with our best cAE results that made use of speaker normalisation in the form of VTLN. The results will only show the effect of increasing the number of training languages because the acoustic models are always trained with i-vectors. Including i-vectors yielded a small performance gain over not doing so; we also tried applying VTLN to the MFCCs for TDNN training, but found no additional benefit.

3.3. Results and discussion

As a sanity check we include word error rates (WER) for the acoustic models trained on the high-resource languages. Table 4 compares the WER of the monolingual SGMM systems that provide the targets for TDNN training to the WER of the final model trained on all 10 high-resource languages. The multilingual model shows small but consistent improvements for all languages except Vietnamese. Ultimately though, we are not so much interested in the performance on typical ASR tasks, but in whether BNFs from this model also generalize to zero-resource applications on unseen languages.

Fig. 3 shows AP on the same-different task of multilingual BNFs trained from scratch on an increasing number of languages in two randomly chosen orders. We provide two baselines for comparison, drawn from our results in Table 2. Firstly, our best cAE features trained with UTD pairs (row 4, Table 2) are a reference for a fully unsupervised system. Secondly, the best cAE features trained with gold standard pairs (row 6, Table 2) give an upper bound on the cAE performance.

In all 6 languages, even BNFs from a monolingual TDNN already considerably outperform the cAE trained with UTD pairs. Adding another language usually leads to an increase in AP, with the BNFs trained on 8−10 high-resource languages performing the best, also always beating the gold cAE. The biggest performance gain is obtained from adding a second training language—further increases are mostly smaller. The order of languages has only a small effect, although for example adding other Slavic languages is generally associated with an increase in AP on Croatian. This suggests that it may be beneficial to train on languages related to the zero-resource language if possible, but further experiments need to be conducted to quantify this effect.

To determine whether these gains come from the diversity of training languages or just the larger amount of training data, we trained models on the 15 h subset and the full 81 h of the English WSJ corpus, which corresponds to the amount of data of four

[3] The splicing indexes are -1,0,1 -1,0,1 -1,0,1 -3,0,3 -3,0,3 -6,-3,0 0.

Table 4
Word error rates of monolingual SGMM and 10-lingual TDNN ASR system evaluated on the development sets.

Language	Mono	Multi	Language	Mono	Multi
BG	17.5	16.9	PL	16.5	15.1
CS	17.1	15.7	PT	20.5	19.9
DE	9.6	9.3	RU	27.5	26.9
FR	24.5	24.0	TH	34.3	33.3
KO	20.3	19.3	VI	11.3	11.6

GlobalPhone languages. More data does help to some degree, as Fig. 3 shows. But, except for Mandarin, training on just two languages (46 h) already works better.

4. Evaluation using ZRSC data and measures

Do the results from the previous experiments generalise to other corpora and how do they compare to other works? So far we used data from GlobalPhone, which provides corpora collected and formatted similarly for a wide range of languages. However, GlobalPhone is not freely available and no previous zero-resource studies have used these corpora, so in this section we also provide results on the Zero Resource Speech Challenge (ZRSC) 2015 (Versteegh et al., 2015) data sets, which have been widely used in other work. The target languages that we treat as zero-resource are English (from the Buckeye corpus (Pitt et al., 2007)) and Xitsonga (NCHLT corpus (De Vries et al., 2014)). Table 1 includes the statistics of the subsets of these corpora that were used in the ZRSC 2015 and in this work. These corpora are not split into train/dev/test; since training is unsupervised, the system is simply trained directly on the unlabeled test set (which could also be done in deployment). Importantly, no hyperparameter tuning is done on the Buckeye or Xitsonga data, so these results still provide a useful test of generalization. Notably, the Buckeye English corpus contains conversational speech and is therefore different in style from the rest of our data.

For training the cAE on the Buckeye English and Xitsonga corpora, we use the same sets of UTD pairs as Renshaw et al. (2015), which were discovered from frequency-domain linear prediction (FDLP) features. We evaluate using both the same-different measures from above, as well as the ABX phone discriminability task (Schatz et al., 2013) used in the ZRSC and other recent work (Versteegh et al., 2015; Dunbar et al., 2017). The ABX task evaluates phoneme discriminability using minimal pairs: sequences of three phonemes where the central phoneme differs between the two sequences *A* and *B* in the pair, such as b ih n and b eh n. Feature representations are then evaluated on how well they can identify a third triplet *X* as having the same phoneme sequence as either *A* or *B*. See Versteegh et al. (2015) and Dunbar et al. (2017) for details on how the scores are computed and averaged over speakers and phonemes to obtain the final ABX error rate. One usually distinguishes between the *within-speaker*

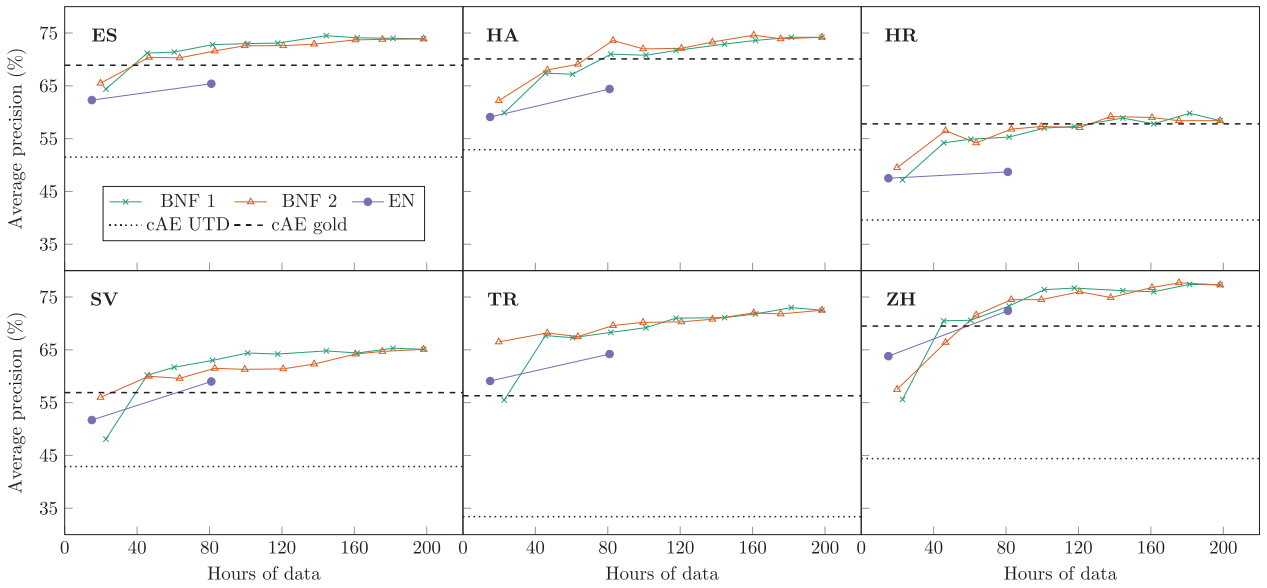


Fig. 3. Same-different task evaluation on the development sets for BNFs trained on different amounts of data. We compare training on up to 10 different languages with additional data in one language (English). For multilingual training, languages were added in two different orders: FR-PT-DE-TH-PL-KO-CS-BG-RU-VI (BNFs 1) and RU-CZ-VI-PL-KO-TH-BG-PT-DE-FR (BNFs 2). Each datapoint shows the result of adding an additional language. As baselines we include the best unsupervised cAE and the cAE trained on gold standard pairs from rows 4 and 6 of Table 2.

error rate where all three triplets belong to the same speaker, and the *cross-speaker* error rate where *A* and *B* are from the same and *X* from a different speaker.

The ABX evaluation includes all such minimal pair phoneme triplets of the evaluation corpus. These pairs therefore rarely correspond to full words, making it a somewhat abstract task whose results may be difficult to interpret when summarizing it as a single final metric. ABX can however be very suitable for more fine-grained analysis of speech phenomena by including only specific phonetic contrasts in the evaluation (Schatz et al., 2018). In contrast, the same-different task always compares whole words and directly evaluates how good feature representations are at telling whether two utterances are the same word or not. Thus it has an immediate link to applications like spoken term detection and it allows easier error analysis. It is also faster to prepare the same-different evaluation set and run the evaluation. We wish to verify that the ABX and same-different measures correlate well, to better compare studies that use only one of them and to allow choosing the task that is more appropriate for the situation at hand.

Table 5 shows results on the Xitsonga and Buckeye English corpora. Here we compare ABX error rates computed with the ZRSC 2015 (Versteegh et al., 2015) evaluation scripts with AP on the same-different task. To the best of our knowledge, this is the first time such a comparison has been made. The results on both tasks correlate well, especially when looking at the ABX cross-speaker error rate because the same-different evaluation as described in Section 2.3 also focuses on cross-speaker pairs. As might be expected VTLN only improves cross-speaker, but not within-speaker ABX error rates.

For comparison we also include ABX results of the official ZRSC 2015 topline (Versteegh et al., 2015), which are posteriorgrams obtained from a supervised speech recognition system, the current state-of-the-art system[4] (Heck et al., 2018b) which even outperforms the topline for English, and the system of Riad et al. (2018) which is the most recent form of the ABNet (Synnaeve et al., 2014), an architecture that is similar to our cAE.

These systems score better than all of our features, but are not directly comparable for several reasons. Firstly, it is unclear how these systems were optimized, since there was no separate development set in ZRSC 2015. Secondly, our features are all 39-dimensional to be directly comparable with MFCCs, whereas the other two systems have higher dimensionality—indeed, the dimensionality of the winning DPGMM system from the ZRSC 2017 was even greater, with more than 1000 dimensions (Heck et al., 2017)—and we don't know whether the competing systems would work as well with fewer dimensions. Such higher dimensional features may be useful in some circumstances, but require more memory and processing power to use, which could be undesirable or even prohibitive for some down-stream applications (such as the unsupervised segmentation and clustering system used in Section 6). Heck et al. (2018a) propose a more complex sampling approach to reduce the potentially very high dimensionality of the features obtained with the DPGMMs in their previous work. However, the resulting dimensionality is still around 90−120 and cannot be controlled precisely, which might be required for down-stream applications.

This complexity of evaluating zero-resource subword modeling systems was addressed in the ZRSC 2019, where the bitrate of the features was added as another evaluation metric alongside the ABX performance (Dunbar et al., 2019). This means that systems are now compared on two dimensions and researchers may choose to trade off between the two, while ultimately the goal is to find a representation that performs well on both measures, like phonemes.

The BNFs are in any case competitive with the higher dimensional features, and have the advantage that they can be built using standard Kaldi scripts and do not require any training on the target language, so can easily be deployed to new languages. The competitive result of Riad et al. (2018) also shows that in general a system trained on word pairs discovered from a UTD system can perform very well.

5. Can we improve the multilingual BNFs?

So far we have shown that multilingual BNFs that are completely agnostic to the target language work better than any of the features trained using only the target language data. However, in principle it could be possible to improve performance further by passing the BNFs as inputs to models that train on the target language data in an unsupervised fashion. We explored this possibility by simply training a cAE using BNFs as input rather than MFCCs. That is, we trained the cAE with the same word pairs as before, but replaced VTLN-adapted MFCCs with the 10-lingual BNFs as input features, without any other changes in the training procedure. Table 6 (penultimate row) shows that the cAE trained with UTD pairs is able to slightly improve on the BNFs in some cases, but this is not consistent across all languages and for Croatian the cAE features are much worse. On the other hand, when trained using gold standard pairs (final row), the resulting cAE features *are* consistently better than the input BNFs. This indicates that BNFs can in principle be improved by further unsupervised target-language training, but the top-down supervision needs to be of higher quality than the current UTD system provides.

This observation leads to a further question: could we improve the UTD pairs themselves by using our improved features (either BNFs or cAE features) as input to the UTD system? If the output is a better set of UTD pairs than the original set, these could potentially be used to further improve the features, and perhaps the process could be iterated as illustrated in Fig. 4. As far as we know, no previously published work has combined unsupervised subword modeling with a UTD system.[5]

Unfortunately, after considerable effort to make this work we found that the ZRTools UTD system seems to be too finely tuned towards features that resemble PLPs to get good results from our new features. Examining the similarity plots for several pairs of

[4] The ZRSC website maintains a list of results: https://zerospeech.com/2015/results.html.

[5] While some other work, such as Lee et al. (2015) and Walter et al. (2013), has focused on joint phonological and lexical discovery, these do not perform representation learning on the low-level features.

Table 5
Comparison of AP on the same-different task (higher is better) and ABX cross-/within-speaker error rates (lower is better) for the Buckeye English and Xitsonga corpora.

Features	Dimensions	English		Xitsonga	
		ABX	Same-diff	ABX	Same-diff
Unsupervised					
MFCC	39	28.4 / 15.5	19.14	33.4 / 20.9	10.46
MFCC+VTLN	39	26.5 / 15.4	24.19	31.9 / 21.4	13.33
cAE	39	24.0 / 14.5	31.97	23.8 / 14.8	22.79
cAE+VTLN	39	22.9 / 14.3	37.85	22.6 / 14.5	47.41
Weak multilingual supervision					
BNF	39	18.0 / 12.4	60.19	17.0 / 12.3	63.44
ZRSC Topline (Versteegh et al., 2015)	49	16.0 / 12.1	-	4.5 / 3.5	-
Heck et al. (2018b)	139–156	14.9 / 10.0	-	11.7 / 8.1	-
Riad et al. (2018)	100	17.2 / 10.4	-	15.2 / 9.4	-

Table 6
AP on the same-different task when training cAE on the 10-lingual BNFs from above (cAE-BNF) with UTD and gold standard word pairs (test set results). Baselines are MFCC+VTLN and the cAE models from rows 4 and 6 of Table 2 that use MFCC+VTLN as input features. Best result without target language supervision in bold.

Features	ES	HA	HR	SV	TR	ZH
MFCC+VTLN	44.1	22.3	25.0	34.3	17.9	33.4
cAE UTD	72.1	41.6	41.6	53.2	29.3	52.8
cAE gold	85.1	66.3	58.9	67.1	47.9	70.8
10-lingual BNFs	**85.3**	**71.0**	**56.8**	72.0	**65.3**	77.5
cAE-BNF UTD	85.0	67.4	40.3	**74.3**	64.6	**78.8**
cAE-BNF gold	89.2	79.0	60.8	79.9	69.5	81.6

utterances helps explain the issue, and also reveals interesting qualitative differences between our learned features and the PLPs, as shown in Figs. 6 and 5. Darker areas in these plots indicate higher acoustic similarity, so diagonal line segments point to similar sequences, as in Fig. 6 where both utterances contain the words *estados unidos*. The ZRTools UTD toolkit identifies these diagonal lines with fast computer vision techniques (Jansen and Van Durme, 2011) and then runs a segmental-DTW algorithm only in the candidate regions for efficient discovery of matches.

PLPs are designed to contain fine-grained acoustic information about the speech signal and can therefore vary a lot throughout the duration of a phoneme. Accordingly, the diagonal lines in Fig. 6(a) are very thin and there is a lot of spurious noise that does not necessarily correspond to phonetically similar units. This pattern is similar for VTLN-adapted PLPs in (b), but with less noise.

On the other hand, cAE features and BNFs are trained to ignore such local variation within phonemes. This results in significantly different appearance of frame-wise cosine similarity plots of two utterances. The trained features remain more constant throughout the duration of a phoneme, resulting in wider diagonal lines in the similarity plots. Especially cAE features are very good at learning

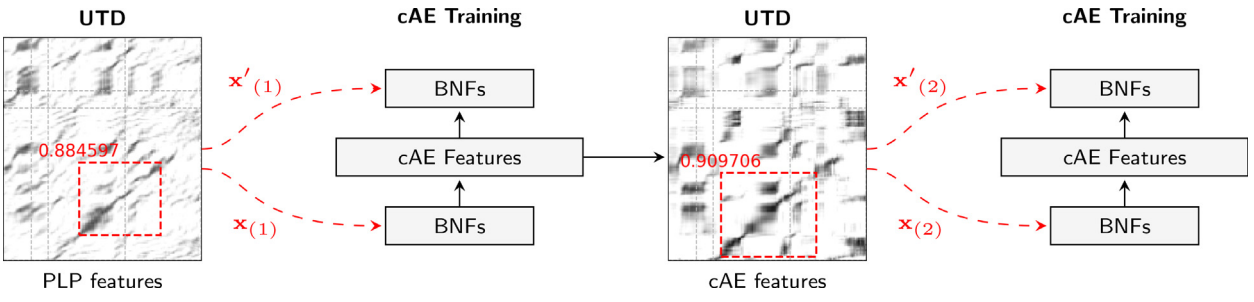
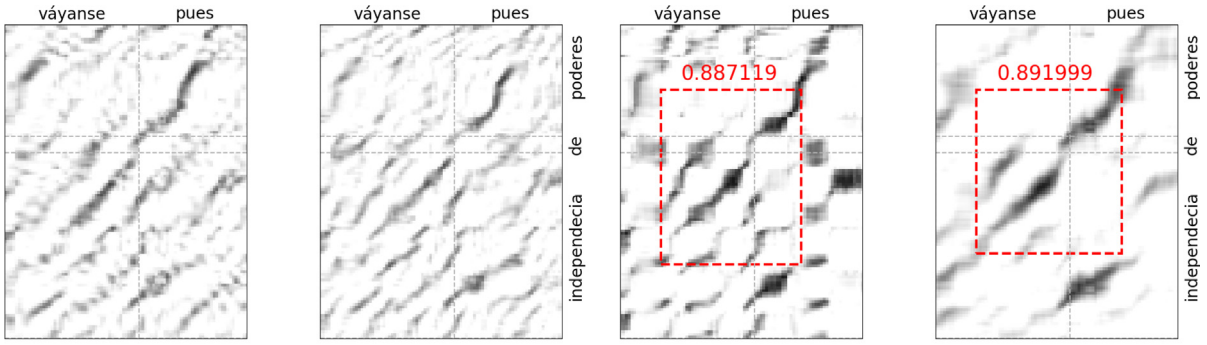


Fig. 4. Cycling cAE and UTD systems: Iteration 1) UTD is run on PLP features to obtain word pairs. Pairs (**x, x'**) are then represented using multilingual BNFs and used to train the cAE. Iteration 2) Features from the last hidden cAE layer are passed to the UTD system, which discovers new word pairs that can be used in the next iteration of cAE training.



(a) PLP (b) PLP-VTLN (c) cAE UTD (d) BNF

Fig. 5. Frame-wise cosine similarity matrices for two Spanish utterances from different speakers, comparing different feature representations. Dark regions correspond to high cosine similarity and values below 0.4 are clipped. Red rectangles mark matches discovered by the UTD system and include their DTW similarity scores. The discovered matches are incorrect—although phonetically similar—and found only for cAE features and BNFs.

phoneme-level information, indicated by the large rectangular blocks in Fig. 6(c) where phonemes of the two utterances match or are very similar. We also found the boundaries of these blocks to align well with actual phoneme boundaries provided by forced alignment. This is despite the cAE not having any information about phoneme identities or boundaries during training.

Parameters in the segmental DTW algorithm of ZRTools, which searches for exact matches within the identified diagonal line segments where matches are likely to occur, include the maximum deviation of the path from the diagonal, and similarity budgets and thresholds that determine how far the path should extend at each end (Jansen and Van Durme, 2011). While ZRTools still finds the diagonal lines in cAE features and BNFs, the DTW algorithm then finds too many exact matches because the lines are much wider and similarity values overall higher than for PLPs. For example Fig. 5 shows a typical example of phonetically similar, but incorrect matches that are only discovered in cAE features and BNFs.

While this behaviour can be compensated for to some degree by changing ZRTools' parameters, we could not identify metrics that can reliably predict whether a given set of discovered UTD pairs will result in better cAE performance. Thus tuning these parameters is very time-consuming because it requires running both UTD and cAE training for each step. Although it might be possible to eventually identify a set of DTW parameters that can work with features that are relatively stable within phones, it could be more productive to consider different approaches for these types of features.

6. Downstream application: segmentation and clustering

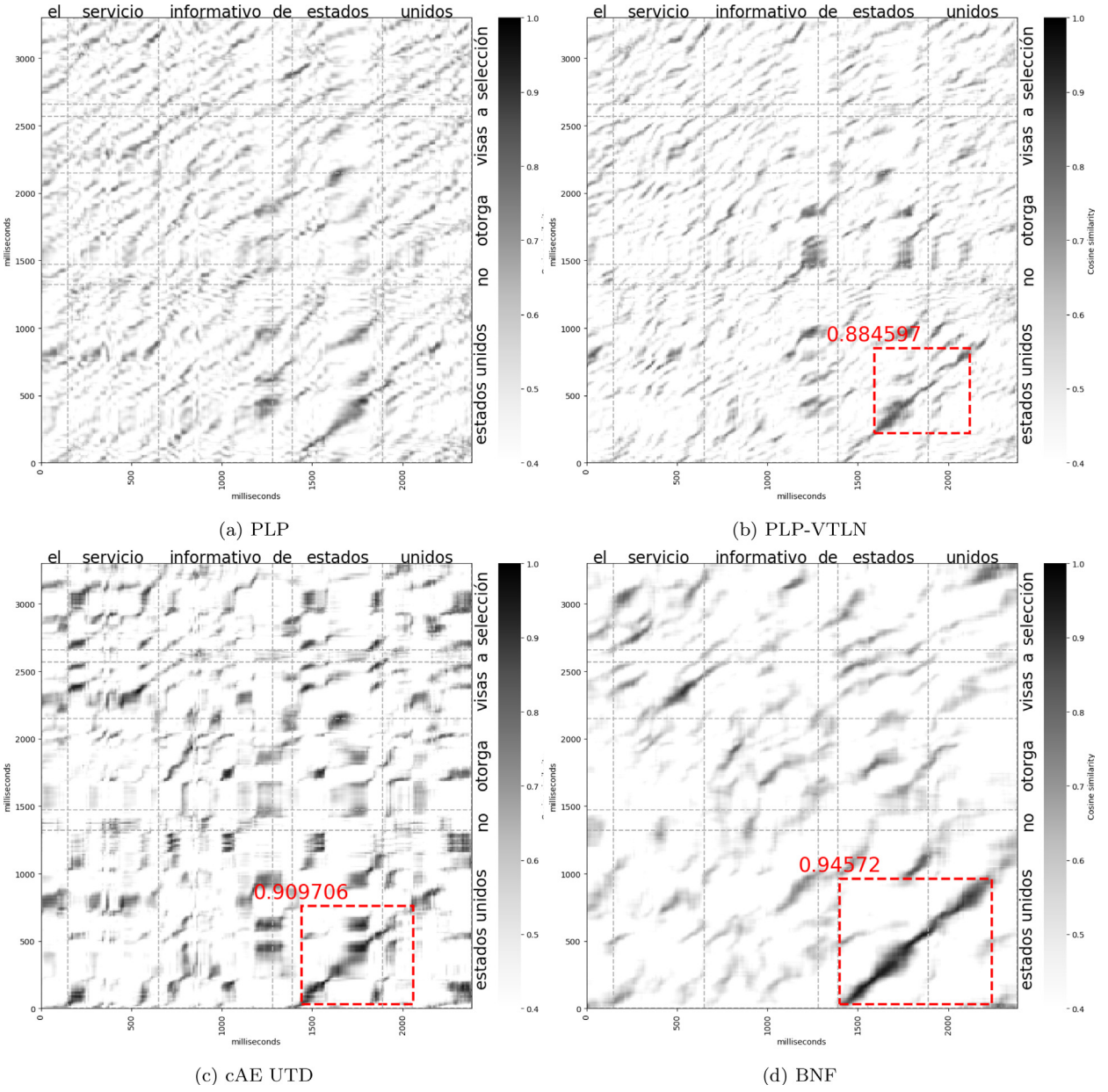
Our experiment with the UTD system was disappointing, suggesting that although cAE features and BNFs improve intrinsic discriminability measures, they may not work with some downstream zero-resource tools. However, ZRTools is a single example. To further investigate the downstream effects of the learned features, we now consider the task of full-coverage speech segmentation and clustering. The aim here is to tokenize the entire speech input into hypothesized categories, potentially corresponding to words, and to do so without any form of supervision—essentially a form of unsupervised speech recognition. Such systems could prove useful from a speech technology perspective in low-resource settings, and could be useful in studying how human infants acquire language from unlabeled speech input.

Here we specifically investigate whether our BNFs improve the Bayesian embedded segmental Gaussian mixture model (BES-GMM), first proposed by Kamper et al. (2016). This approach relies on a mapping where potential word segments (of arbitrary length) are embedded in a fixed-dimensional acoustic vector space. The model, implemented as a Gibbs sampler, builds a whole-word acoustic model in this acoustic embedding space, while jointly performing segmentation. Several acoustic word embedding methods have been considered, but here we use the very simple approach also used by Kamper et al. (2017): any segment is uniformly downsampled so that it is represented by the same fixed number of frame-level features, which are then flattened to obtain the fixed-dimensional embedding (Levin et al., 2013).

6.1. Experimental setup and evaluation

We retrained the cAE and BNF models to return 13-dimensional features with all other parameters unchanged to be consistent with the experiments of Kamper et al. (2017) and for computational reasons. We also did not tune any hyperparameters of the BES-GMM for our new input features. Nonetheless, our baseline cAE results do not exactly correspond to the ones of Kamper et al. (2017) because for example the MFCC input features have been extracted with a different toolkit and we used a slightly different training procedure.

We use several metrics to compare the resulting segmented word tokens to ground truth forced alignments of the data. By mapping every discovered word token to the ground truth word with which it overlaps most, average **cluster purity** can be calculated as the total proportion of correctly mapped tokens in all clusters. More than one cluster may be mapped to the same ground truth



(a) PLP (b) PLP-VTLN

(c) cAE UTD (d) BNF

Fig. 6. Frame-wise cosine similarity matrices for two Spanish utterances from different speakers, comparing different feature representations. Dark regions correspond to high cosine similarity and values below 0.4 are clipped. Red rectangles mark matches discovered by the UTD system and include their DTW similarity scores. In this case the match is not found with PLPs as input features.

word type. In a similar way, we can calculate **unsupervised word error rate (WER)**, which uses the same cluster-to-word mapping but also takes insertions and deletions into account. Here we consider two ways to perform the cluster mapping: many-to-one, where more than one cluster can be assigned the same word label (as in purity), or one-to-one, where at most one cluster is mapped to a ground truth word type (accomplished in a greedy fashion). We also compute the **gender and speaker purity** of the clusters, where we want to see clusters that are as diverse as possible on these measures, i.e., low purity. To explicitly evaluate how accurate the model performs segmentation, we compare the proposed word boundary positions to those from forced alignments of the data (falling within a single true phoneme from the boundary). We calculate boundary precision and recall, and report the resulting **word boundary F-scores**. We also calculate **word token F-score**, which requires that both boundaries from a ground truth word token be correctly predicted.

Table 7
Segmentation and clustering results (lower scores are better, except for token and boundary F-score, and cluster purity).

Features	WER		F-score		Purity		
	one-to-one ↓	many-to-one ↓	Token ↑	Boundary ↑	Cluster ↑	Gender ↓	Speaker ↓
English							
MFCC	93.7	82.0	29.0	42.4	29.9	87.6	55.9
cAE	93.7	82.4	28.9	42.3	29.3	83.1	49.9
cAE+VTLN	93.6	82.1	29.0	42.3	29.9	75.8	44.8
BNF	**92.0**	**77.9**	**29.4**	**42.9**	**36.6**	**67.6**	**35.5**
Xitsonga							
MFCC	102.4	89.8	19.4	43.6	24.5	87.1	43.0
cAE	101.8	89.7	19.5	43.2	24.5	82.5	37.6
cAE+VTLN	100.7	84.7	20.1	44.5	31.0	74.7	32.7
BNF	**96.4**	**76.9**	**20.6**	**44.6**	**38.8**	**65.6**	**27.5**

6.2. Results

Table 7 compares MFCCs, cAE features (with and without VTLN) and BNFs as input to the system of Kamper et al. (2017). It shows that both VTLN and BNFs help on all metrics, with improvements ranging from small to more substantial and BNFs clearly giving the most benefit. The effects of VTLN are mostly confined to reducing both gender and speaker purity of the identified clusters (which is desirable) while maintaining the performance on other metrics.[6] This means that the learned representations have become more invariant to variation in speaker and gender, which is exactly what VTLN aims to do. However, this appears to be insufficient to also help other metrics, aligning with the experiments by Kamper et al. (2017) that indicate that improvements on the other metrics are hard to obtain.

On the other hand, BNFs result in better performance across all metrics. While some of these improvements are small, they are very consistent across all metrics. In particular, we observe a much higher cluster purity and lower word error rates, which both indicate that more tokens are correctly identified. Gender and speaker purity have decreased further, which means that the BNFs are even more agnostic to gender and speaker variations than the cAE features with VTLN. This shows that the BNFs are also useful for down-stream tasks in zero-resource settings. It especially demonstrates that such BNFs which are trained on high-resource languages without seeing any target language speech at all are a strong alternative to fully unsupervised features for practical scenarios or could in turn be used to improve unsupervised systems trained on the target language speech data.

7. Conclusions

In this work we investigated different representations obtained using data from the target language alone (i.e., fully unsupervised) and from multilingual supervised systems trained on labeled data from non-target languages. We found that the correspondence autoencoder (cAE), a recent neural approach to unsupervised subword modeling, learns complementary information to the more traditional approach of VTLN. This suggests that VTLN should also be considered by other researchers using neural approaches. On the other hand, our best results were achieved using multilingual bottleneck features (BNFs). Although these results do not completely match the state-of-the-art features learned from target language data only (Heck et al., 2017; 2018b), they still perform well and have the advantage of only requiring a single model from which features can be immediately extracted for new target languages. Our BNFs showed robust performance across the 8 languages we evaluated without language-specific parameter-tuning. In addition, it is easy to control the dimensionality of the BNFs, unlike in the nonparametric models of Heck et al. (2017, 2018b), and this allowed us to use them in the downstream task of word segmentation and clustering. We observed consistent improvements from BNFs across all metrics in this downstream task, and other work demonstrates that these features are also useful for downstream keyword spotting in settings with very small amounts of labeled data (Menon et al., 2018). We also showed that it is theoretically possible to further improve BNFs with language-specific unsupervised training, and we hope to explore models that can do this more reliably than the cAE in the future.

Finally, our qualitative analysis showed that both cAE features and BNFs tend to vary much less over time than traditional PLPs, supporting the idea that they are better at capturing phonetic information rather than small variations in the acoustics. Although this property helps explain the better performance on intrinsic measures and the segmentation task, it harms performance for unsupervised term discovery, where the system seems heavily tuned towards PLPs. Therefore, our work also points to the need for term discovery systems that are more robust to different types of input features.

[6] Perfectly balanced clusters would have a speaker purity of 8.3% for English and 4.2% for Xitsonga, and a gender purity of 50% for both corpora.

Acknowledgements

This research was funded in part by a James S. McDonnell Foundation Scholar Award.

References

Alumäe, T., Tsakalidis, S., Schwartz, R.M., 2016. Improved multilingual training of stacked neural network acoustic models for low resource languages. In: Proc. Interspeech, pp. 3883–3887.
Badino, L., Canevari, C., Fadiga, L., Metta, G., 2014. An auto-encoder based approach to unsupervised learning of subword units. In: Proc. ICASSP, pp. 7634–7638.
Badino, L., Mereta, A., Rosasco, L., 2015. Discovering discrete subword units with binarized autoencoders and hidden-Markov-model encoders. In: Proc. Interspeech.
Carlin, M.A., Thomas, S., Jansen, A., Hermansky, H., 2011. Rapid evaluation of speech representations for spoken term discovery. In: Proc. Interspeech, pp. 828–831.
Chen, H., Leung, C.-C., Xie, L., Ma, B., Li, H., 2015. Parallel inference of dirichlet process gaussian mixture models for unsupervised acoustic modeling: a feasibility study. In: Proc. Interspeech, pp. 3189–3193.
Chen, H., Leung, C.-C., Xie, L., Ma, B., Li, H., 2017. Multilingual bottle-neck feature learning from untranscribed speech. In: Proc. ASRU, pp. 727–733.
Cui, J., Kingsbury, B., Ramabhadran, B., Sethy, A., Audhkhasi, K., et al., 2015. Multilingual representations for low resource speech recognition and keyword search. In: Proc. ASRU, pp. 259–266.
De Vries, N.J., Davel, M.H., Badenhorst, J., Basson, W.D., De Wet, F., Barnard, E., De Waal, A., 2014. A smartphone-based ASR data collection tool for under-resourced languages. Speech Commun. 56, 119–131.
Dunbar, E., Algayres, R., Karadayi, J., Bernard, M., Benjumea, J., Cao, X.-N., Miskic, L., Dugrain, C., Ondel, L., Black, A.W., Besacier, L., Sakti, S., Dupoux, E., 2019. The zero resource speech challenge 2019: TTS without T. In: Proc. Interspeech, pp. 1088–1092.
Dunbar, E., Cao, X.N., Benjumea, J., Karadayi, J., Bernard, M., Besacier, L., Anguera, X., Dupoux, E., 2017. The zero resource speech challenge 2017. In: Proc. ASRU, pp. 323–330.
Gales, M.J., 1998. Maximum likelihood linear transformations for HMM-based speech recognition. Comput. Speech Lang. 12 (2), 75–98.
Grézl, F., Karafiát, M., Veselý, K., 2014. Adaptation of multilingual stacked bottle-neck neural network structure for new language. In: Proc. ICASSP, pp. 7704–7708.
Heck, M., Sakti, S., Nakamura, S., 2016. Supervised learning of acoustic models in a zero resource setting to improve DPGMM clustering. In: Proc. Interspeech, pp. 1310–1314.
Heck, M., Sakti, S., Nakamura, S., 2017. Feature optimized DPGMM clustering for unsupervised subword modeling: a contribution to zerospeech 2017. In: Proc. ASRU, pp. 740–746.
Heck, M., Sakti, S., Nakamura, S., 2018. Dirichlet process mixture of mixtures model for unsupervised subword modeling. 14 (8), 1–16.
Heck, M., Sakti, S., Nakamura, S., 2018. Learning supervised feature transformations on zero resources for improved acoustic unit discovery. IEICE Trans. Inf. Syst. 101 (1), 205–214. https://doi.org/10.1587/transinf.2017EDP7175.
Hermann, E., Goldwater, S., 2018. Multilingual bottleneck features for subword modeling in zero-resource languages. In: Proc. Interspeech, pp. 2668–2672.
Huijbregts, M., McLaren, M., Van Leeuwen, D., 2011. Unsupervised acoustic sub-word unit detection for query-by-example spoken term detection. In: Proc. ICASSP, pp. 4436–4439.
Jansen, A., Thomas, S., Hermansky, H., 2013. Weak top-down constraints for unsupervised acoustic model training. In: Proc. ICASSP, pp. 8091–8095.
Jansen, A., Van Durme, B., 2011. Efficient spoken term discovery using randomized algorithms. In: Proc. ASRU, pp. 401–406. https://doi.org/10.1109/ASRU.2011.6163965.
Kamper, H., Elsner, M., Jansen, A., Goldwater, S., 2015. Unsupervised neural network based feature extraction using weak top-down constraints. In: Proc. ICASSP, pp. 5818–5822.
Kamper, H., Jansen, A., Goldwater, S., 2016. Unsupervised word segmentation and lexicon discovery using acoustic word embeddings. IEEE/ACM Trans. Audio Speech Lang. Process. 24 (4), 669–679.
Kamper, H., Jansen, A., Goldwater, S., 2017. A segmental framework for fully-unsupervised large-vocabulary speech recognition. Comput. Speech Lang. 46, 154–174.
Lang, K.J., Waibel, A.H., Hinton, G.E., 1990. A time-Delay neural network architecture for isolated word recognition. Neural Netw. 3 (1), 23–43.
Lee, C.-y., O'Donnell, T.J., Glass, J., 2015. Unsupervised lexicon discovery from acoustic input. Trans. Assoc. Comput. Linguist. 3, 389–403.
Levin, K., Henry, K., Jansen, A., Livescu, K., 2013. Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings. In: Proc. ASRU, pp. 410–415.
Menon, R., Kamper, H., Yilmaz, E., Quinn, J., Niesler, T., 2018. ASR-free CNN-DTW keyword spotting using multilingual bottleneck features for almost zero-resource languages. In: Proc. SLTU, pp. 20–24.
Paul, D.B., Baker, J.M., 1992. The design for the wall street journal-based CSR corpus. In: Proc. HLT, pp. 357–362.
Peddinti, V., Povey, D., Khudanpur, S., 2015. A time delay neural network architecture for efficient modeling of long temporal contexts. In: Proc. Interspeech, pp. 3214–3218.
Pitt, M.A., Dilley, L., Johnson, K., Kiesling, S., Raymond, W., Hume, E., Fosler-Lussier, E., 2007. Buckeye corpus of conversational speech (second release). Department of Psychology, Ohio State University.
Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlíček, P., Qian, Y., Schwarz, P., Silovský, J., Stemmer, G., Veselý, K., 2011. The Kaldi speech recognition toolkit. In: Proc. ASRU.
Renshaw, D., Kamper, H., Jansen, A., Goldwater, S., 2015. A comparison of neural network methods for unsupervised representation learning on the zero resource speech challenge. In: Proc. Interspeech, pp. 3199–3203.
Riad, R., Dancette, C., Karadayi, J., Zeghidour, N., Schatz, T., Dupoux, E., 2018. Sampling strategies in siamese networks for unsupervised speech representation learning. In: Proc. Interspeech, pp. 2658–2662.
Saon, G., Soltau, H., Nahamoo, D., Picheny, M., 2013. Speaker adaptation of neural network acoustic models using i-vectors. In: Proc. ASRU, pp. 55–59.
Schatz, T., Bach, F., Dupoux, E., 2018. Evaluating automatic speech recognition systems as quantitative models of cross-lingual phonetic category perception. J. Acoust. Soc. Am. 143 (5), 372–378.
Schatz, T., Peddinti, V., Bach, F., Jansen, A., Dupoux, E., Schatz, T., Peddinti, V., Bach, F., Jansen, A., Hermansky, H., Schatz, T., Peddinti, V., Bach, F., 2013. Evaluating speech features with the minimal-pair ABX Task : analysis of the classical MFC/PLP pipeline. In: Proc. Interspeech, pp. 1781–1785.
Schultz, T., Vu, N.T., Schlippe, T., 2013. GlobalPhone: a multilingual text & speech database in 20 languages. In: Proc. ICASSP, pp. 8126–8130.
Shibata, H., Kato, T., Shinozaki, T., Watanabe, S., 2017. Composite embedding systems for zerospeech 2017 Track 1. In: Proc. ASRU, pp. 747–753.
Swietojanski, P., Li, J., Renals, S., 2016. Learning hidden unit contributions for unsupervised acoustic model adaptation. IEEE/ACM Trans. Audio Speech Lang. Process. 24 (8), 1450–1463.
Synnaeve, G., Dupoux, E., 2016. A temporal coherence loss function for learning unsupervised acoustic embeddings. Procedia Comput. Sci. 81, 95–100.
Synnaeve, G., Schatz, T., Dupoux, E., 2014. Phonetics embedding learning with side information. In: Proc. SLT, pp. 106–111.
Thomas, S., Ganapathy, S., Hermansky, H., 2012. Multilingual MLP features for low-resource LVCSR Systems. In: Proc. ICASSP, pp. 4269–4272.
Trmal, J., Wiesner, M., Peddinti, V., Zhang, X., Ghahremani, P., Wang, Y., Manohar, V., Xu, H., Povey, D., Khudanpur, S., 2017. The Kaldi OpenKWS system: improving low resource keyword search. In: Proc. Interspeech, pp. 3597–3601.
Tsuchiya, T., Tawara, N., Ogawa, T., Kobayashi, T., 2018. Speaker invariant feature extraction for zero-resource languages with adversarial learning. In: Proc. ICASSP, pp. 2381–2385.

Versteegh, M., Thiolliere, R., Schatz, T., Cao, X.N., Anguera, X., Jansen, A., Dupoux, E., 2015. The zero resource speech challenge 2015. In: Proc. Interspeech, pp. 3169–3173.
Veselý, K., Karafiát, M., Grézl, F., Janda, M., Egorova, E., 2012. The language-independent bottleneck features. In: Proc. SLT, pp. 336–341. https://doi.org/10.1109/SLT.2012.6424246.
Vu, N.T., Breiter, W., Metze, F., Schultz, T., 2012. An investigation on initialization schemes for multilayer perceptron training using multilingual data and their effect on ASR performance. In: Proc. Interspeech, pp. 2586–2589.
Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., Lang, K.J., 1989. Phoneme recognition using time-delay neural networks. IEEE Trans. Acoust. 37 (3), 328–339.
Walter, O., Korthals, T., Haeb-Umbach, R., Raj, B., 2013. A hierarchical system for word discovery exploiting DTW-based initialization. 2013 IEEE Workshop on Automatic Speech Recognition and Understanding. IEEE, pp. 386–391.
Yuan, Y., Leung, C.-C., Xie, L., Chen, H., Ma, B., Li, H., 2017. Extracting bottleneck features and word-like pairs from untranscribed speech for feature representation. In: Proc. ASRU, pp. 734–739.
Yuan, Y., Leung, C.-c., Xie, L., Chen, H., Ma, B., Li, H., 2017. Pairwise learning using multi-lingual bottleneck features for low-resource query-by-example spoken term detection. In: Proc. ICASSP, pp. 5645–5649.
Yuan, Y., Leung, C.-C., Xie, L., Ma, B., Li, H., 2016. Learning neural network representations using cross-lingual bottleneck features with word-pair information. In: Proc. Interspeech, pp. 788–792.
Zeghidour, N., Synnaeve, G., Usunier, N., Dupoux, E., 2016. Joint learning of speaker and phonetic similarities with siamese networks. In: Proc. Interspeech 2016, pp. 1295–1299.
Zeiler, M.D., Ranzato, M., Monga, R., Mao, M., Yang, K., Le, Q.V., Nguyen, P., Senior, A., Vanhoucke, V., Dean, J., et al., 2013. On rectified linear units for speech processing. In: Proc. ICASSP, pp. 3517–3521.
Zhang, Y., Glass, J.R., 2009. Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams. In: Proc. ASRU, pp. 398–403.