Multilingual bottleneck features for subword modeling in zero-resource languages

Enno Hermann[1], *Sharon Goldwater*[1]

[1]ILCC, School of Informatics, University of Edinburgh, UK

ehermann@inf.ed.ac.uk, sgwater@inf.ed.ac.uk

Abstract

How can we effectively develop speech technology for languages where no transcribed data is available? Many existing approaches use no annotated resources at all, yet it makes sense to leverage information from large annotated corpora in other languages, for example in the form of multilingual bottleneck features (BNFs) obtained from a supervised speech recognition system. In this work, we evaluate the benefits of BNFs for subword modeling (feature extraction) in six unseen languages on a word discrimination task. First we establish a strong unsupervised baseline by combining two existing methods: vocal tract length normalisation (VTLN) and the correspondence autoencoder (cAE). We then show that BNFs trained on a single language already beat this baseline; including up to 10 languages results in additional improvements which cannot be matched by just adding more data from a single language. Finally, we show that the cAE can improve further on the BNFs if high-quality same-word pairs are available.

Index Terms: multilingual bottleneck features, subword modeling, unsupervised feature extraction, zero-resource speech technology

1. Introduction

Recent years have seen increasing interest in "zero-resource" speech technology: systems developed for a target language without transcribed data or other hand-curated resources. One challenge for these systems, highlighted by the Zero Resource Speech Challenge (ZRSC) of 2015 [1] and 2017 [2], is to improve subword modeling, i.e., to extract speech features from the target language audio that work well for word discrimination or downstream tasks such as query-by-example.

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.

There is considerable evidence that bottleneck features (BNFs) extracted using a multilingually trained deep neural network (DNN) can improve ASR for target languages with just a few hours of transcribed data [3–7]. However, there has been little work so far exploring supervised multilingual BNFs for target languages with no transcribed data at all. [8, 9] trained *monolingual* BNF extractors and showed that applying them cross-lingually improves word discrimination in a zero-resource setting. [10, 11] 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 [12], the authors 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 correspondence autoencoder [13] 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 [14], the authors 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 are truly zero-resource.

This paper presents a more thorough evaluation of multilingual BNFs, trained on between one and ten languages from the GlobalPhone collection and evaluated on six others. 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.

Since BNF training uses no target language data at all, we also compare to methods that train unsupervised on the target language, either alone or in combination with the multilingual training. We use a correspondence autoencoder (cAE) [13], which 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. In the setting with target language data only, we find that applying vocal tract length normalisation (VTLN) to the input of both the UTD and cAE systems improves the learned features considerably, suggesting that cAE and VTLN abstract over different aspects of the signal. Nevertheless, BNFs trained on just a single other language already outperform the cAE-only training, with multilingual BNFs doing better by a wide margin.

We then tried fine-tuning the multilingual BNFs to the target language by using them as input to the cAE. When trained with UTD word pairs, we found no benefit to this fine-tuning. However, training with manually labeled word pairs did yield benefits, suggesting that this type of supervision can help fine-tune the BNFs if the word pairs are sufficiently high-quality.

2. Experimental setup

2.1. Dataset

We use 16 languages from the GlobalPhone corpus of speech read from news articles [15]. The selected languages and dataset

sizes are shown in Table 1. We consider the 10 languages in the top section with a combined 198.3 hours of speech as high-resource languages, where transcriptions are available to train a supervised automatic speech recognition (ASR) system. We treat the 6 languages in the bottom section as zero-resource languages on which we evaluate the new feature representations.

In addition we use the English Wall Street Journal (WSJ) corpus [16] which is comparable to the GlobalPhone corpus. We either use the entire 81 hours or only a 15 hour subset, so that we can compare the effect of increasing the amount of data for one language with training on data from 4 GlobalPhone languages.

Table 1: *Dataset sizes (hours). About 100 speakers per language with 80% of these in the training set and no speaker overlap.*

Language		Train	Dev	Test
High-resource				
Bulgarian	(BG)	17.1	2.3	2.0
Czech	(CS)	26.8	2.4	2.7
French	(FR)	22.8	2.1	2.0
German	(DE)	14.9	2.0	1.5
Korean	(KO)	16.6	2.2	2.1
Polish	(PL)	19.4	2.8	2.3
Portuguese	(PT)	22.8	1.6	1.8
Russian	(RU)	19.8	2.5	2.4
Thai	(TH)	21.2	1.5	0.4
Vietnamese	(VI)	16.9	1.4	1.5
English81 WSJ	(EN)	81.3	1.1	0.7
English15 WSJ		15.1	-	-
Zero-resource				
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

2.2. Baseline features

For baseline features, we use Kaldi [17] 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 vocal tract length normalisation (VTLN), a simple feature-space speaker adaptation technique that 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, a diagonal-covariance universal background model with 1024 components trained on each language's training data. Warp factors can then be extracted for both the training and for unseen data.

2.3. Bottleneck features

For monolingual training of the high-resource languages, we follow the Kaldi recipes for the GlobalPhone and WSJ corpora and train a subspace Gaussian mixture model (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. We train a time-delay neural network (TDNN) [18] with block softmax [19], i.e. all hidden
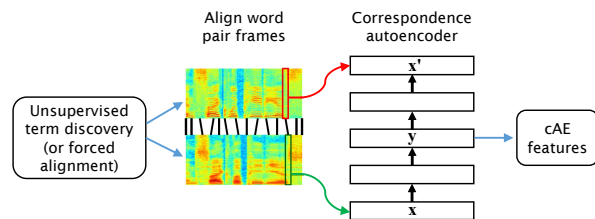

Figure 1: *Correspondence autoencoder training procedure (see section 2.4). Parts of this figure due to Herman Kamper, used with permission.*

layers are shared between languages, but there is a separate output layer for each language and for each training instance only the error at the corresponding language's output layer is used to update the weights. The TDNN has six 625-dimensional hidden layers[1] 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.

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. 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.

2.4. Correspondence autoencoder

In several experiments we further adapt the baseline features or BNFs using a cAE network. The cAE attempts to normalize out non-linguistic factors such as speaker, channel, gender, etc., using top-down information from pairs of similar speech segments. Extracting cAE features requires three steps, as illustrated in Figure 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 DNN. After training, a middle layer $\mathbf{y}$ is used as the learned feature representation.

To obtain the UTD pairs, we used a freely available UTD system[2] [20] 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.

To provide an upper bound on cAE performance, we also report results using *gold standard* same-word pairs for cAE training. As in [12, 13, 21], we force-align the target language data and extract all the same-word pairs that are at least 5 characters and 0.5 seconds long (between 89k and 102k pairs for each language).

---

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

[2]https://github.com/arenjansen/ZRTools

Following [9, 13], we train the cAE model[3] 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.

2.5. Evaluation

We evaluate all speech features on the same-different task [22] 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. By varying the threshold a precision-recall curve can be computed, where 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 as above, from all words that are at least 5 characters and 0.5 seconds long, except that we now also include different-word pairs.

We note that previous work [13, 22] computed 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 explicitly compute the recall only for SWDP pairs to focus the evaluation of features on their speaker invariance.

As a sanity check, we also provide word error rates (WER) for the ASR systems trained on the high-resource languages.

3. Results

3.1. Using target language data only

Our first set of experiments aims to find the best features that can be extracted using target language data only. Previous work has shown that cAE features are better than MFCCs, especially for cross-speaker word discrimination [9], but we know of no direct comparison between cAE features and VTLN, which can also be trained without transcriptions.

Table 2 shows AP results on all target languages for baseline features, cAE features learned using raw features as input (as in previous work), and cAE features learned using VTLN-adapted features as input to either the UTD system, the cAE, or both. We find that cAE features as trained previously are slightly better than MFCC+VTLN, but can be improved considerably by applying VTLN to the input of both UTD and cAE training—indeed,

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
PLP		28.6	39.9	26.9	22.2	25.2	20.4
PLP	+VTLN	46.2	48.2	36.3	37.9	31.4	35.7
PLP+VTLN		40.4	45.7	35.8	25.8	25.9	26.9
PLP+VTLN	+VTLN	**51.5**	**52.9**	**39.6**	**42.9**	**33.4**	**44.4**
Gold pairs		65.3	65.2	55.6	52.9	50.6	60.5
Gold pairs	+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

even using gold pairs as cAE input applying VTLN is beneficial. This suggests that cAE training and VTLN abstract over different aspects of the speech signal, and that both should be used when only target language data is available.

3.2. Multilingual training

Table 3 compares the WER of the monolingual SGMM systems which 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.

Figure 2 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 (from row 4 of Table 2) are a reference for a fully unsupervised system. Secondly, the best cAE features trained with gold standard pairs (from row 6 of 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. However, the biggest performance gain is from adding a second training language—further increases are mostly smaller. The order of languages has only a small effect, although for example adding

Table 3: *Word error rates of monolingual SGMM and 10-lingual TDNN ASR system evaluated on the development sets.*

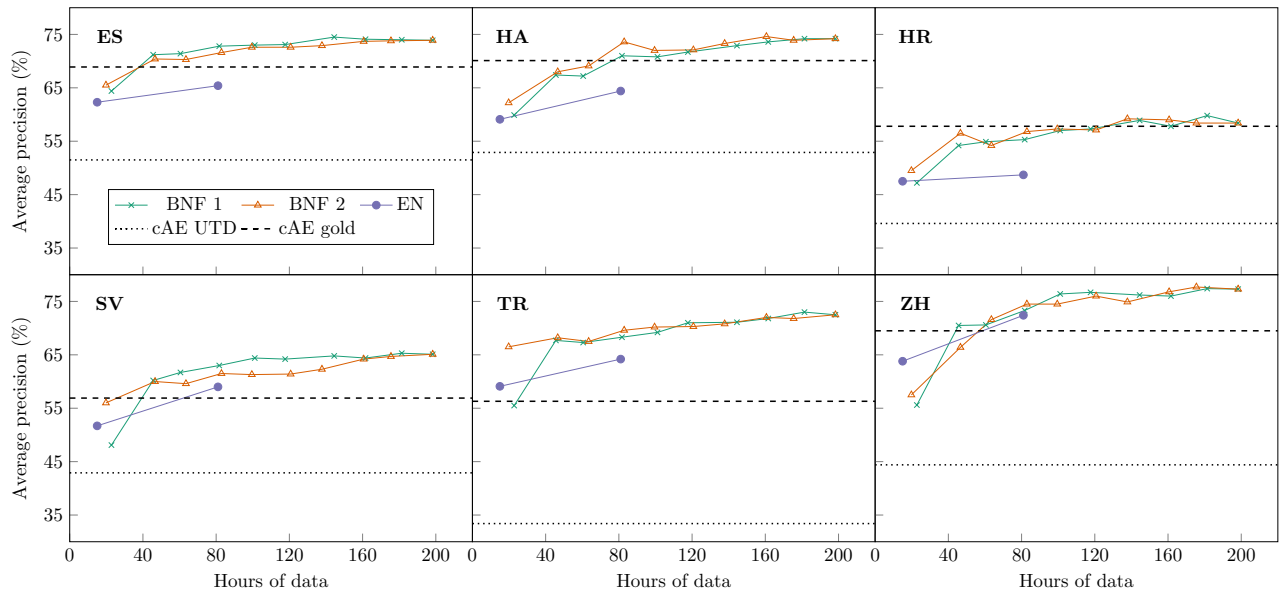| Language	Mono	Multi		Mono	Multi
BG	17.5	16.9	PL	16.5	15.1
CS	17.1	15.7	PT	20.5	19.9
DE	9.6	9.3	RU	27.5	26.9
FR	24.5	24.0	TH	34.3	33.3
KO	20.3	19.3	VI	11.3	11.6

Figure 2: *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.*

Table 4: *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

other Slavic languages is generally associated with an increase in AP on Croatian, suggesting that it may be beneficial to train on languages related to the zero-resource language.

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 hour subset and the full 81 hours of the English WSJ corpus, which corresponds to the amount of data of four GlobalPhone languages. More data does help to some degree, as Figure 2 shows, but except for Mandarin training on just two languages (46 hours) already works better.

3.3. cAE results

Previous work [13] and our baselines in Table 2 show that a fully unsupervised system like a cAE generates features that can discriminate between words much better than standard acoustic features like MFCCs. Is the cAE also able to further improve on

multilingual BNFs which already have a much higher baseline performance?

We trained the cAE with the same sets of 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 4 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. The limiting factor appears to be the quality of the UTD pairs. With gold standard pairs, the cAE features improve in all languages.

4. Conclusions

We evaluated multilingual BNFs trained on up to 10 high-resource languages on a word discrimination task in 6 zero-resource languages. These BNFs outperform both standard acoustic features like MFCCs and cAE features trained in a fully unsupervised way. We showed that training on multiple languages helps the BNFs and that just training on more data in a single language does not work as well. While the cAE is theoretically able to further improve on the BNFs, this does not work in practice if only word pairs discovered by a UTD system are available. In future work we would like to further analyze the complementary nature of VTLN and cAE training and explore the benefits of these multilingual BNFs for down-stream zero-resource applications like speech-to-text translation.

5. Acknowledgements

6. References

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

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

[3] K. Veselý, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, "The Language-independent Bottleneck Features," in *Proc. SLT*, 2012, pp. 336–341.

[4] N. T. Vu, W. Breiter, F. Metze, and T. Schultz, "An investigation on initialization schemes for multilayer perceptron training using multilingual data and their effect on ASR performance," in *Proc. Interspeech*, 2012, pp. 2586–2589.

[5] S. Thomas, S. Ganapathy, and H. Hermansky, "Multilingual MLP features for low-resource LVCSR systems," in *Proc. ICASSP*, 2012, pp. 4269–4272.

[6] J. Cui, B. Kingsbury, B. Ramabhadran, A. Sethy, K. Audhkhasi *et al.*, "Multilingual representations for low resource speech recognition and keyword search," in *Proc. ASRU*, 2015, pp. 259–266.

[7] T. Alumäe, S. Tsakalidis, and R. M. Schwartz, "Improved multilingual training of stacked neural network acoustic models for low resource languages." in *Proc. Interspeech*, 2016, pp. 3883–3887.

[8] Y. Yuan, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Learning neural network representations using cross-lingual bottleneck features with word-pair information," in *Proc. Interspeech*, 2016, pp. 788–792.

[9] D. Renshaw, H. Kamper, A. Jansen, and S. Goldwater, "A Comparison of Neural Network Methods for Unsupervised Representation Learning on the Zero Resource Speech Challenge," in *Proc. Interspeech*, 2015, pp. 3199–3203.

[10] Y. Yuan, C.-C. Leung, L. Xie, H. Chen, B. Ma, and H. Li, "Extracting Bottleneck Features and Word-Like Pairs from Untranscribed Speech for Feature Representation," in *Proc. ASRU*, 2017, pp. 734–739.

[11] H. Chen, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Multilingual Bottle-Neck Feature Learning from Untranscribed Speech," in *Proc. ASRU*, 2017, pp. 727–733.

[12] Y. Yuan, C.-c. Leung, L. Xie, H. Chen, B. Ma, and H. Li, "Pairwise Learning Using Multi-lingual Bottleneck Features for Low-Resource Query-by-Example Spoken Term Detection," in *Proc. ICASSP*, 2017, pp. 5645–5649.

[13] H. Kamper, M. Elsner, A. Jansen, and S. Goldwater, "Unsupervised Neural Network Based Feature Extraction Using Weak Top-Down Constraints," in *Proc. ICASSP*, 2015, pp. 5818–5822.

[14] H. Shibata, T. Kato, T. Shinozaki, and S. Watanabe, "Composite Embedding Systems for Zerospeech 2017 Track 1," in *Proc. ASRU*, 2017, pp. 747–753.

[15] T. Schultz, N. T. Vu, and T. Schlippe, "GlobalPhone: A Multilingual Text & Speech Database in 20 Languages," in *Proc. ICASSP*, 2013, pp. 8126–8130.

[16] D. B. Paul and J. M. Baker, "The Design for the Wall Street Journal-based CSR Corpus," in *Proc. HLT*, 1992, pp. 357–362.

[17] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi Speech Recognition Toolkit," in *Proc. ASRU*, 2011.

[18] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech*, 2015, pp. 3214–3218.

[19] F. Grézl, M. Karafiát, and K. Veselý, "Adaptation of Multilingual Stacked Bottle-neck Neural Network Structure for New Language," in *Proc. ICASSP*, 2014, pp. 7704–7708.

[20] A. Jansen and B. Van Durme, "Efficient spoken term discovery using randomized algorithms," in *Proc. ASRU*, 2011, pp. 401–406.

[21] A. Jansen, S. Thomas, and H. Hermansky, "Weak top-down constraints for unsupervised acoustic model training," in *Proc. ICASSP*, 2013, pp. 8091–8095.

[22] M. A. Carlin, S. Thomas, A. Jansen, and H. Hermansky, "Rapid Evaluation of Speech Representations for Spoken Term Discovery," in *Proc. Interspeech*, 2011, pp. 828–831.