Building an ASR System for Mboshi Using A Cross-language Definition of Acoustic Units Approach

Odette Scharenborg[1,2], *Patrick Ebel*[2], *Francesco Ciannella*[3], *Mark Hasegawa-Johnson*[4], *and Najim Dehak*[5]

[1]Multimedia Computing Group, Delft University of Technology, Delft, the Netherlands
[2]Centre for Language Studies, Radboud University, Nijmegen, the Netherlands
[3]Carnegie Mellon University, Pittsburgh, PA, USA
[4]ECE Department & Beckman Institute, University of Illinois, Urbana-Champaign, IL, USA
[5]Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, USA
o.e.scharenborg@tudelft.nl

Abstract

For many languages in the world, not enough (annotated) speech data is available to train an ASR system. Recently, we proposed a cross-language method for training an ASR system using linguistic knowledge and semi-supervised training. Here, we apply this approach to the low-resource language Mboshi. Using an ASR system trained on Dutch, Mboshi acoustic units were first created using cross-language initialization of the phoneme vectors in the output layer. Subsequently, this adapted system was retrained using Mboshi self-labels. Two training methods were investigated: retraining of only the output layer and retraining the full deep neural network (DNN). The resulting Mboshi system was analyzed by investigating per phoneme accuracies, phoneme confusions, and by visualizing the hidden layers of the DNNs prior to and following retraining with the self-labels. Results showed a fairly similar performance for the two training methods but a better phoneme representation for the fully retrained DNN.

Index Terms: Low-resource automatic speech recognition, Cross-language adaptation, Semi-supervised training

1. Introduction

Typically a large amount of annotated speech data is required to build automatic speech recognition (ASR) systems that work reasonably well. However, for many languages in the world, not enough of such (annotated) speech data is available to train an ASR system. Due to differences between languages in phone inventories, and the fact that phones transcribed with the same IPA symbol can be produced slightly differently between different languages [1], using an ASR trained on a different language typically performs quite poorly [2]. Most of the world's languages have however been investigated by field linguists, consequently some information about the language is available. Recently, we proposed a cross-language definition of units method, which uses linguistic knowledge of the low-resource language and a semi-supervised training paradigm to build an ASR system for a low-resource language through the adaptation of a high-resource language ASR system [3].

The crucial part of adapting an ASR system from one language to another language is the creation of phones that are present in the low-resource language but not in the high-resource language. In [3], we proposed the following three-step method to create the missing phone units: First, a DNN-based ASR system is built on a high-resource language, in our case Dutch. Second, the acoustic units for the 'missing' phones are initialized through a linear extrapolation between existing acoustic units in the high-resource ASR system's soft-max layer. Third, the adapted model's output layer is iteratively retrained using self-labelled phone sequences of the low-resource language in order to improve the acoustic phone units.

This approach has three assumptions: 1) the availability of some unlabelled speech data of the low-resource language (in line with the Zero-resource approach, e.g., [4][5][6]); 2) the availability of a 'description' of the phone(me) inventory of the language, e.g., obtained from a field linguist; 3) the availability of enough annotated speech material of a related high-resource language to build an ASR system for that related high-resource language. The proposed approach was tested on the adaptation of a Dutch ASR system to English, which was taken to be a low-resource language. The results showed a significant, though limited, improvement in phone error rate after retraining the acoustic units on the self-labelled utterances.

The work presented in this paper extends the previous work in several important directions. First, the approach is tested on an actual low-resource language without orthography, Mboshi. By doing so, the approach is tested on an unrelated language pair: Dutch and Mboshi, thus investigating the claim in [3] that the proposed method does not rely on having a high-resource *related* language but that the proposed approach should work for any language pair. In [3], it was suggested that part of the reason for the only limited phone recognition improvement was that only the output layer was retrained with the self-labels. Here, we retrain the entire DNN model and compare that with a model in which, like for [3], only the output layer was retrained. Moreover, since the ultimate goal of this work is to build an ASR system for low-resource languages that are able to capture and correctly represent all phonemes of the low-resource language, we provide in-depth performance analyses of the newly created and existing phoneme representations by looking at phoneme confusions and through the visualization of the hidden layers of the DNN.

2. Methodology

2.1. Data

The Spoken Dutch Corpus (Corpus Gesproken Nederlands, CGN, [7]) is a corpus of almost 9M words of Dutch spoken in the Netherlands and in Flanders (Belgium), in 14 different

speech styles. For the experiments reported here, we only used the read speech material from the Netherlands, which amounts to 551,624 words for a total duration of approximately 64 hours of speech. This data was split into a training and test set of 90% and 10%, respectively. Mono-phone forced-alignments and 40-dimensional Filterbank features were created using Kaldi [8].

The Mboshi (Bantu language spoken in Congo-Brazzaville) corpus [9] consists of 5k speech utterances (approximately 4 hours of speech) in Mboshi aligned to French text. The data set also contains linguists' transcriptions in Mboshi in the form of a non-standard graphemic form close to the language phonology [10][11][12]. In our experiments, we used the 2087 training utterances for retraining the *Baseline* model and the 230 validation set utterances for which mono-phone forced alignments were available from [11]. 40-dimensional filterbanks were computed for every 10 ms using Kaldi.

2.2. Cross-language unit definition approach

The adaptation approach consist of three steps (see for more details [3]). First, a baseline DNN is trained on the Dutch CGN. Next, the soft-max layer of the DNN is adapted from the Dutch to the Mboshi phone set (see Section 2.2.2). Third, the adapted soft-max layer is used to decode the Mboshi speech material using a free phone recognition pass. The baseline system is subsequently retrained with these Mboshi self-labels using two methods: In one model only the output layer was retrained, following [3]; in the other model, the full DNN was retrained. In both cases, the models are retrained for 20 epochs, where after each epoch the self-labels are updated (based on the network's own predictions), such that the following epoch uses the previously updated self-labels. The accuracy of the phone recognition systems is evaluated at the frame level by comparing them to the gold standard as created by the forced alignment (see Section 2.1), and is reported as percentage frame classification accuracy.

2.2.1. Baseline model architecture

The baseline model used for the experiments is a feed-forward DNN, implemented using Tensorflow [13]. The baseline DNN consists of 6 fully connected hidden layers, each containing 1024 units trained using logistic sigmoid nonlinearities. The network is trained to optimize a cross entropy loss function via Stochastic Gradient Descent for 20 epochs at a learning rate of 0.1 with batches of size 512 and dropout (0.5). The input to the DNN is a frame of 10 ms duration in a context of its five preceding and succeeding frames. The output layer consists of units with soft-max activation functions and its size depends on the phone set of the language; see Section 2.2.2. The model re-training over the self-labeled training data is similar to that of the training phase, except for a learning rate of 0.01. Accuracy of the *Baseline* model on the CGN test data was 71.74%.

2.2.2. Adaptation of the soft-max layer

The number of different Dutch phones in CGN is 43, while Mboshi has 68 different phones (see for more detail on the Mboshi phone inventory [12]). The output layer of the baseline model trained on Dutch thus needs to be adapted in several ways. Three Mboshi phones, /ʤ, ⁿʤ , ʦ/ are affricates which do not exist in Dutch. Affricates are not presented in the soft-max layer, instead, in both the training and test material, for each affricate acoustic segment, the first half of all frames are assigned the label of the first phoneme of the affricate while the second half of the frames are assigned the label of the second half of the phoneme. /ʤ, ʦ/ can easily be constructed from a sequence of two Dutch phones (i.e., /d/ + /z/ and /t/ + /s/, respectively). ⁿʤ can be seen as a combination of /ⁿd/ + /z/. Sixteen Dutch phones do not exist in Mboshi and these are removed from the soft-max layer. Mboshi has seven vowels (i.e., /i, e, ɛ, a, ɔ, o, u/) which can be long or short and can have four tones. In the current set-up, as Dutch does not have contrastive tones, tone information is removed from the vowels. Since there are no spectral differences between long and short vowels, duration can be dealt with in a post-processing step.

Finally, eight Mboshi phones do not exist in Dutch, these are referred to as 'missing L2 phonemes', and these thus need to be created and added to the soft-max layer of the baseline model. To that end, following [3], for each of the missing L2 phonemes a vector is created in the soft-max layer of the baseline model on the basis of the trained Dutch (L1) phones, and initialized by linearly extrapolating the missing L2 (Mboshi) node in the soft-max layer from existing vectors for the Dutch L1 phones using (adapted from [3]):

$$\vec{V}_{|\varphi|,L2} = \gamma \bullet \vec{V}_{|\varphi|,L1:1} + \alpha\,(\vec{V}_{|\varphi|,L1:2} - \vec{V}_{|\varphi|,L1:3}) \qquad (1)$$

where $\vec{V}_{|\varphi|,L2}$ is the vector of the missing L2 phone φ,L2 that needs to be created, $\vec{V}_{|\varphi|,L1:x}$ are the vectors of the Dutch L1 phones φ,L1:x in the soft-max layer that are used to create the vector for the missing L2 phone φ,L2. Among the three Dutch phones, L1:1 refers to the phone which is used as the starting point from which to extrapolate the missing L2 phone, and L1:2 and L1:3 refer to the L1 phones whose displacement is used as an approximation of the displacement between the Dutch L1 vector and the L2 phone that should be created. γ is a factor which increases the non-linearity of the output function, i.e., it increases the sensitivity to the inputs: if an input is of small value then the output will be less inclined to be active but if the input is of a large value then the output will be active earlier. γ is set to 1.5 on the basis of tuning experiments. α is a factor corresponding to the approximation of the displacement of $\vec{V}_{|\varphi|,L2}$ from $\vec{V}_{|\varphi|,L1:1}$. The total number of Mboshi vectors in the output layer is 31.

Table 1 lists the 8 missing Mboshi phones, and the Dutch L1 phones that are used to create the vectors for the missing Mboshi L2 phonemes. While for /ß/, α is set to 0.5, for the other sounds α is (arbitrarily) set to 0.3 to denote a 'a bit of' prenasalisation and/or postfrication. For /ᵐbᵛ/, .5(m-v) denotes the midway point between the vectors for Dutch /m/ and /v/. The total number of Mboshi vectors in the output layer is 31.

Table 1. *Mapping of the Mboshi (L2) phone not present in the Dutch phoneme inventory.*

L2 phone	Mapping			L2 phone	Mapping		
	L1:1	L1:2	L1:3		L1:1	L1:2	L1:3
ᵐbᵛ	b	.5(m-v)	b	ⁿd	d	n	d
bᵛ	b	v	b	pᶠ	p	f	p
ᵑg	g	ŋ	g	ß	b	v	b
ᵐb	b	m	b	ᵐw	w	m	w

2.3. I-vector representation for visualization

In [14], we presented a method to visualize the different clusters the DNN learned from the information present in the hidden layers using a discrete version of the I-vector representation to

the hidden node activations. The I-vector representation allows us to capture the acoustic variability and model the behavior of the neural responses of the DNNs. The discrete I-vector captures the DNN hidden layer responses, for a given segment of the speech signal, as a shift from the average responses of all phones. This is done by normalizing the hidden node activations, summing over the segment; then projecting the result onto a lower-dimensional basis using non-negative factor analysis (NFA) [16][17], which can be represented as:

$$M=m+Tw$$

where M is the 1024-dimensional segment summary vector, and m is the average across all segments in the corpus. The matrix T models the most important non-negative factors of variability in the DNN's reactions to the set of phone segments. The I-vector w describes the best segment dependent offset within the span of the subspace defined by matrix T. The matrix T is trained using an EM-Like algorithm [17].

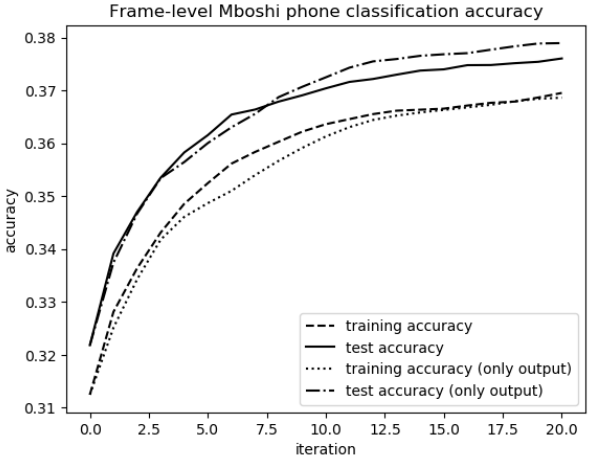

Figure 1. Frame classification accuracies per epoch on the Mboshi training and test sets for the two retrained models.

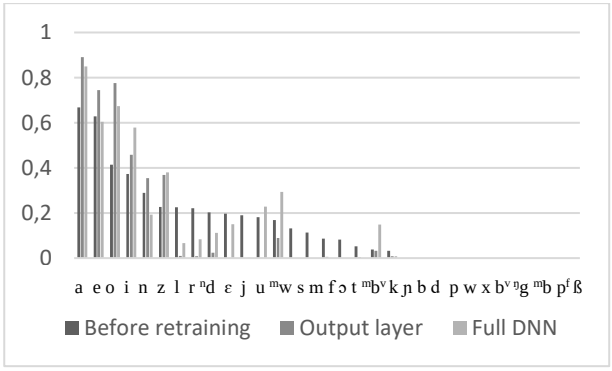

Figure 2. Frame accuracies per phoneme category for the three models.

3. Results

3.1. Frame-level accuracy

The baseline model's frame classification accuracy results, i.e., the Dutch model whose output layer has been adapted to contain only Mboshi phones but which has not yet been retrained on the Mboshi speech self-labels, is 31.27% on the Mboshi training data and 32.18% on the (independent, unseen) Mboshi test data. Figure 1 shows the frame classification accuracy as a variable of the training epoch on the training data (dotted line) and the test data (dashed-dotted line) after only retraining the output layer, and after retraining the full DNN model: training data accuracy (dashed line) and test data accuracy (solid line).As the results clearly show, retraining the baseline model with the self-labeled Mboshi data improves frame-level classification substantially for both the training and test data. The trained phone categories generalize well to unseen test data indicated by the higher performances on the test data. (Note that the models are trained on the self-labels while being evaluated on the ground-truth labels. Performance on the training data thus does not necessarily have to be better than on the test data.) Retraining only the output layer (dashed-dotted line) slightly outperforms the full model retraining (solid) with a small margin of .29% absolute.

Next we investigated which phonemes in particular benefited from the retraining. Ideally, those would not only be the phonemes which already existed in Dutch. Figure 2 shows the frame accuracies per phoneme category before retraining with the self-labels (darkest grey), after retraining the output layer (middle grey), and after retraining the full DNN (lightest grey), ordered from highest to lowest accuracy before retraining. Of the Mboshi phonemes shared with Dutch (i.e., for which a Dutch vector existed in the output layer), for 7 of the 23 phonemes, retraining with the self-labels improved classification accuracy, while for 10 the performance reduced. For 6 phonemes, retraining did not do anything, their performance was 0% before and after retraining. Of the newly created L2 phonemes, 3 phonemes, /ⁿd, ᵐw, ᵐbᵛ/, were already classified correctly for some frames prior to training. Retraining these phonemes improved performance for /ᵐw, ᵐbᵛ/.

For /a, e, o, n/, retraining only the output layer outperformed retraining the full DNN model. For /I, z, u, ᵐw, ᵐbᵛ/ retraining the full model outperformed the output layer model. So, although training on only the output layer improved frame accuracy the most – albeit with only a small margin, retraining the full model did yield improvements for slightly more phonemes (5 vs. 4), and the highest accuracies for 2 of our 3 newly created phonemes. More research is needed to understand the differences between the effect of initialization and retraining on the different phonemes and on how to improve the system for all phonemes.

Looking at the phoneme confusions, we observe that all phonemes, thus including the consonants, are often misclassified as one of the five vowels. In fact, of the top five recognized phonemes for each phoneme, often 2 or 3 are vowels. This is surprising considering the vast differences in articulation between vowels and consonants. Potentially this is a data scarcity issue, however, as the number of frames in the training material with a vowel label is about a magnitude larger than that for consonants, presumably resulting in much better defined vowel categories than consonant categories. Moreover, there is a high confusability with / ᵐbᵛ / and /n/ for all phonemes. These two phones both have a fairly low number of training frames, which makes it surprising they so often show up in the top 5 of recognized phonemes.

One would expect that the new, L2 phonemes with pre-nasalization are also misclassified as nasals, while the phonemes with voiced post-frication may be misclassified as /w/, the closest phoneme to an alveolar voiced fricative, and the L2 phone with unvoiced pre-nasalization as /f/. This however seems not to be the case, which indicates that the newly created vectors are different enough from the original, Dutch vectors on the basis of which these new phonemes were created.

3.2. Visualizations

To further investigate the representations of the phonemes and the effect retraining with the self-labels has on these representations, we visualized the activations of the hidden units of the baseline model and the fully retrained model after retraining with the self-labels. Figures 3-4 show 3D t-sne [18] visualizations of the 500 dimensional I-vectors. These vectors are trained on the activations of the nodes of the last hidden layer to the input sounds for the baseline model and the fully retrained model, respectively [14,16,17] (note that the hidden layers of the model with only output layer retraining are identical to those of the baseline model). The I-vector and t-sne were trained on the activations of all phonemes. Each type of symbol denotes a different phoneme.

Comparing the different phoneme clusters in the two models seems to suggest a slight improvement of the phoneme clusters for the fully retrained model compared to the baseline model. The clusters of the retrained model seem to be organized in a better spherical structure which is the behavior previous noticed in the I-vector representation of the hidden layer. The clusters seem to be denser for the baseline model, while the distances between the clusters seem to be more spread out for the retrained model, compare, e.g., the red crosses (/f/) and black dots (/s/) (right-side of Figures 3 and 4) and the purple stars (/o/) and blue crosses (/ɔ/) left-side of Figure 3, left-bottom of Figure 4). Also the well-performing newly created phoneme vectors seem to be well defined, e.g., /^{m}w/ (green +).

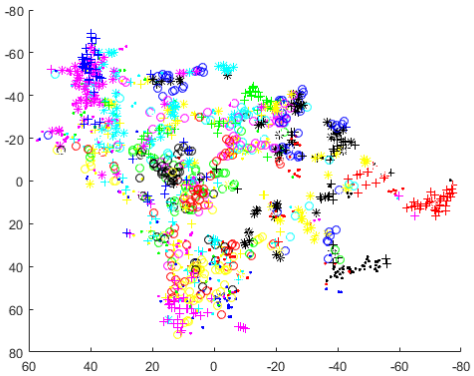

Figure 3. t-sne visualization of the activations of the 6th hidden layer to the Mboshi speech for the baseline model.

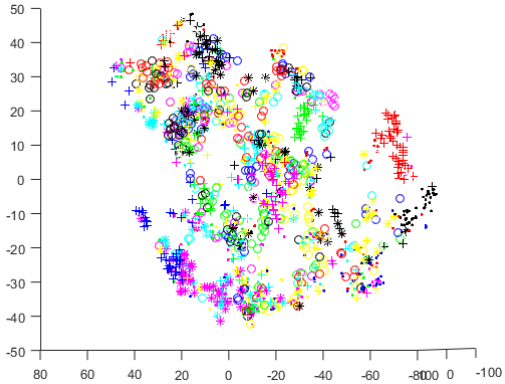

Figure 4. t-sne visualization of the activations of the 6th hidden layer for the fully retrained model.

4. Discussion

We here apply our previously proposed cross-language definition of acoustic units approach [3] to build an ASR system for the low-resource language Mboshi through the adaptation of an ASR system of the high-resource language Dutch. After adaptation of the output layer of the baseline Dutch system to only contain Mboshi phones, and then retraining on the Mboshi self-labels, we obtained an improvement of 6.33% and 6.62% for the fully retrained model and the model with only output layer retraining, respectively. The same method when applied to Dutch to English in [3], showed an increase of .87% absolute. These results cannot be directly compared however as we report frame accuracies while [3] reports phone error rates. Thus, although experiments in cross-language ASR adaptation tend to report that adaptation between related languages is more successful than adaptation among unrelated languages [2], these results show that the proposed approach also works for unrelated language pairs such as Dutch-Mboshi. The visualizations seem to support the conclusion that retraining with self-labels improves the phoneme representations.

The second aim of this paper was to investigate the effect of retraining only the output layer as was done in [3] or retraining the full DNN model with the self-labels on the classification performance and the definition of the acoustic units. The differences between the two training methods were small; with a slightly overall performance for the output only retraining method but a slightly better phoneme representation for the full DNN model retraining method.

The in-depth analyses showed a mixed picture: some phonemes showed improvement after retraining, others a deterioration, while again others are seemingly so poorly defined that they are never correctly classified. The lack of an improvement for phonemes after retraining with the self-labels could be due to data scarcity as many of these phonemes had relatively few training frames, while DNNs are known to be data hungry. Another possible reason is a suboptimal initialization of the feature vectors. Future research will improve the initialization. The phoneme confusion analyses showed that many of the phonemes, including consonants, were confused with vowels. This finding is again likely to be explained by data scarcity: there was a magnitude more training data for the vowels than for the other phonemes. Furthermore, many of the phonemes were confused with /^{m}b^{v}/ and /n/.

Future work will focus on the question why some phoneme vectors benefit from retraining with the self-labels while others do not, and improve the approach accordingly. Moreover, future work will investigate the use of data augmentation methods to increase the importance of the good data [15]. An important step will be the step from frame level classification to phoneme identification, and subsequently word segmentation. Finally, we will continue to improve our visualization technique for the visualization of the hidden and output layers of DNNs as they provide useful information.

To conclude, we successfully applied our cross-language definition of acoustic units approach to Mboshi. Results showed a fairly similar performance for the two training methods but slightly better phoneme representations for the fully retrained DNN. Visualizations were helpful in investigating the representation of the phonemes in the hidden and output layers and will be used in future research to improve the initialization of the phoneme vectors and the retraining of the vectors.

5. Acknowledgements

O.S. was partly supported by a Vidi-grant from NWO (grant number: 276-89-003). The work carried out by P.E. was part of an internship supervised by O.S.

6. References

[1] P.-S. Huang and M. Hasegawa-Johnson, “Cross-dialectal data transferring for Gaussian Mixture Model training in Arabic speech recognition,” in International Conference on Arabic Language Processing (CITALA) pp. 119-122, ISBN 978-9954-9135-0-5, Rabat, Morocco, 2012.

[2] M. Hasegawa-Johnson, P. Jyothi, D. McCloy, M. Mirbagheri, G. di Liberto, A. Das, B. Ekin, C. Liu, V. Manohar, H. Tang, E.C. Lalor, N. Chen, P. Hager, T. Kekona, R. Sloan, A.K.C. Lee, , “ASR for under-resourced languages from probabilistic transcription,” IEEE/ACM Trans. Audio, Speech and Language 25(1):46-59, 2017. doi:10.1109/TASLP.2016.2621659

[3] O. Scharenborg, F. Ciannella, S. Palaskar, A. Black, F. Metze, L. Ondel, M. Hasegawa-Johnson, “Building an ASR system for a low-resource language through the adaptation of a high-resource language ASR system: Preliminary results”, Proceedings of the International Conference on Natural Language, Signal and Speech Processing, Casablanca, Morocco, 2017.

[4] A. Jansen, E. Dupoux, S. Goldwater, M. Johnson, S. Khudanpur, K. Church, N. Feldman, H. Hermansky, F. Metze, R. Rose, “A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition”, Proceedings of ICASSP, 2013.

[5] L. Ondel, L. Burget, J. Cernocky, “Variational Inference for Acoustic Unit Discovery”, Procedia Computer Science, 81, Elsevier Science, http://www.fit.vutbr.cz/research/view_pub.php?id=11224, 2016.

[6] D. Harwath and J. Glass, “Deep multimodal semantic embeddings for speech and images”, IEEE Automatic Speech Recognition and Understanding Workshop, Scottsdale, Arizona, USA, 237-244, 2015.

[7] N.H.J. Oostdijk, W. Goedertier, F. Van Eynde, L. Boves, J.-P. Martens, M. Moortgat, H. Baayen, “Experiences from the Spoken Dutch Corpus project”, Proceedings LREC – Third International Conference on Language Resources and Evaluation, Las Palmas de Gran Canaria, 340-347, 2002.

[8] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, K. Veselý, “The Kaldi speech recognition toolkit”, IEEE Workshop on Automatic Speech Recognition and Understanding, 1-4, 2011.

[9] The dataset will be made available for free by ELRA; its current version is online at: https://github.com/besacier/mboshi-french-parallel-corpus

[10] G. Adda, S. Stüker, M. Adda-Decker, O. Ambouroue, L. Besacier, D. Blachon, H. Bonneau-Maynard, P. Godard, F. Hamlaoui, D. Idiatov, G.-N. Kouarata, L. Lamel, E.-M. Makasso, A. Rialland, M. Van de Velde, F. Yvon, S. Zerbian, “Breaking the unwritten language barrier: The BULB project”, Proceedings SLTU, 5th Workshop on Spoken Language Technologies for Under-resourced languages, 2016.

[11] P. Godard, G. Adda, M. Adda-Decker, J. Benjumea, L. Besacier, J. Cooper-Leavitt, G.-N. Kouarata, L. Lamel, H. Maynard, M. Müller, A. Rialland, S. Stüker, F. Yvon, and M. Zanon Boito, “A very low resource language speech corpus for computational language documentation experiments”, Proceedings of LREC, Miyazaki, Japan, 2018.

[12] J. Cooper-Leavitt, L. Lamel, A. Rialland, M. Adda-Decker, G. Adda, “Developing an Embosi (Bantu C25) speech variant dictionary to model vowel elision and morpheme deletion”, Proceedings of Interspeech, Stockholm, Sweden, 2017.

[13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, C., et al., “TensorFlow: large-scale machine learning on heterogeneous distributed systems”, 2016 https://arxiv.org/abs/1603.04467.

[14] O. Scharenborg, S. Tiesmeyer, M. Hasegawa-Johnson, N. Dehak, “Visualizing phoneme category adaptation in deep neural networks”, Proceedings of Interspeech, Hyderabad, India.

[15] Jaitly, N., Hinton, G.E., “Vocal tract length perturbation (VTLP) improves speech recognition”, Proc. ICML Workshop on Deep Learning for Audio, Speech and Language, 2013.

[16] N. Dehak, “I-vector representation based on GMM and DNN for audio classification”, *Keynote speech at Odyssey 2016 Speaker and Language Workshop*, 2016.

[17] M.H. Bahari, N. Dehak, H. Van hamme, L. Burget, A.M. Ali, & J. Glass, “Non-negative factor analysis of Gaussian mixture model weight adaptation for language and dialect recognition”, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 7, pp. 1117-1129, 2014.

[18] L.J.P. van der Maaten & G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9 (Nov) : 2579-2605, 2008.