DBLSTM BASED MULTILINGUAL ARTICULATORY FEATURE EXTRACTION FOR LANGUAGE DOCUMENTATION

Markus Müller[1], *Sebastian Stüker*[1], *Alex Waibel* [1,2]

[1]Karlsruhe Institute of Technology, Karlsruhe, Germany
[2]Carnegie Mellon University, Pittsburgh PA, USA

ABSTRACT

With more than 7,000 living languages in the world and many of them facing extinction, the need for language documentation is now more pressing than ever. This process is time-consuming, requiring linguists as each language features peculiarities that need to be addressed. While automating the whole process is difficult, we aim at providing methods to support linguists during documentation. One important step in the workflow is the discovery of the phonetic inventory. In the past, we proposed a first approach of first automatically segmenting recordings into phone-line units and second clustering these segments based on acoustic similarity, determined by articulatory features (AFs). We now propose a refined method using Deep Bi-directional LSTMs (DBLSTMs) over DNNs. Additionally, we use Language Feature Vectors (LFVs) which encode language specific peculiarities in a low dimensional representation. In contrast to adding LFVs to the acoustic input features, we modulated the output of the last hidden LSTM layer, forcing groups of LSTM cells to adapt to language related features. We evaluated our approach multilingually, using data from multiple languages. Results show an improvement in recognition accuracy across AF types: While LFVs improved the performance of DNNs, the gain is even bigger when using DBLSTMs.

Index Terms— speech recognition, low-resource, language documentation

1. INTRODUCTION

For certain tasks speech recognition systems for resource-rich languages like English have recently achieved human-like performance [1, 2]. But there exists a long tail of under-resourced languages for which systems of such high recognition accuracy or even systems at all do not yet exist. Special methods are required to handle low-resource scenarios and, while progress has been made in recent years, there still is a large gap in performance between systems for low-resource languages compared to those of resource-rich languages. Building systems requires different types of resources, among them pronunciation dictionaries and audio recordings with time-aligned transcriptions. Such resources do not exist for unknown and potentially unwritten languages. Therefore, when it comes to creating systems for supporting the documentation of a new language, these systems have to work without the traditional resources used for training natural language processing (NLP) systems. One of the first steps in documenting an unwritten language is often to collect audio data in the field. Documentary linguists then, among other tasks, attempt to derive the phonetic inventory and vocabulary of the language. This is a difficult and time-consuming process, because many (unknown) language specific peculiarities need to be considered. While this process is difficult to automate in its entirety [3], we aim at supporting linguists during this process by providing an automatic phonetic transcription process based on a set of automatically derived phone-like units. Based on the automatically detected phonetic inventory and the automatic phonetic transcription, linguists will then have the opportunity to provide feedback. Based on this feedback, the inferred set of units can then be tuned further, e.g., by changing parameters of the transcription and inventory detection systems.

Our transcription and inventory discovery process consists of three steps: 1) the detection of phone boundaries, 2) the automatic recognition of articulatory features (AF) of the detected phone segments and 3) clustering the detected phone segments into a phone inventory based on the detected AF.

In this paper, we focus on two aspects of this chain of steps. We a) show how we can improve AF detection across languages by utilizing language feature vectors and b) examine possibilities to predict the correct number of units to cluster the detected phone segments into in order to obtain a consist and close to the ground truth phone set.

In earlier work we used Gaussian Mixture Models (GMMs)

This work was realized in the framework of the ANR-DFG project BULB (STU 593/2-1 and ANR-14-CE35-002) and also supported by the French Investissements d'Avenir - Labex EFL program (ANR-10-LABX-0083). This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation award number ACI-1053575. Specifically, it used the Blacklight system, which is supported by NSF award number ACI-1041726, at the Pittsburgh Supercomputing Center (PSC). Part of the work reported here was done at the Jelinek Speech and Language Technology Workshop JSALT 2017, in Pittsburgh, and was supported by JHU and CMU via grants from Alphabet, Amazon, Apple, and Microsoft.

for detecting AFs [4] and showed that it is possible to detect AFs across languages using multilingual models[5]. In recent research, we improved monolingual and crosslingual AF extraction by using deep neural networks (DNNs) [6], which improved results over the GMM systems, and DBLSTMs [7] which yielded yet better results. In this work, we propose and improved DBLSTM based AF extraction, by enhancing the feature inputs into the DBLSTMs with Language Feature Vectors (LFVs) [8] to train neural networks that are better able to detect articulatory features across languages.

When it comes to clustering of the detected phone units into a coherent phone set, one of the challenges is to, explicitly or implicitly, determine the correct amount of units in the inventory. One way to do this, is to cluster sets of different sizes and to estimate the quality of the resulting clusterings in an unsupervised manner. In this paper we thus examine methods of measuring the quality of clustering results and whether from these measures the correct phone inventory size can be derived.

To evaluate our setup, we pretended English to be an unknown language. This allows us to evaluate against the ground truth in order to assess the quality of the articulatory feature detection and unit discovery.

The rest of this paper is organized as follows: In the next Section, we provide an overview of related work in the field. In Sections 3 and 4, we describe our proposed approach, followed by the experimental setup in Section 5. The results are presented in Section 6. This paper concludes with Section 7, where we also provide an outlook to future work.

2. RELATED WORK

2.1. Articulatory Features

The use of AFs has been proposed for different tasks in the past. For example, AFs were used to improve the robustness of speech recognition systems [9, 10]. AFs describe the targets of the articulators in the human vocal tract for the phones of a language. Thus, a phone can be seen as a shorthand notation for a specific bundle of AFs. Therefore AFs can be seen as the atomic units describing the speech sounds produced by the vocal tract.

Each language features a certain phone inventory, called a phone set, while the total number of phones that can be produced by humans is naturally limited by the anatomy of the human vocal tract. Phone sets from different languages often overlap to a certain degree. But as AFs are the building blocks of phones, they are more language universal in nature and their overlap among languages is generally larger than that of phones. Hence, they are better suited for crosslingual acoustic modeling [5].

2.2. Feature Augmentation for Multilingual AF Detection

Feature augmentation is a common technique to provide additional input features to neural networks in order for the network to be able to better compensate for certain modalities. One of the most widespread methods for providing features that help networks to compensate for speaker variability is the use of i-Vectors [11] in addition to acoustic input features for building large vocabulary continuous speech recognition (LVCSR) systems. The addition of i-Vectors enables neural networks to adapt to different speaker characteristics. It is also possible to train a speaker adaptive neural network [12].

For multilingual automatic speech recognition, we proposed a similar approach, but instead of adapting to multiple speakers, we proposed a language code encoding language properties in a multilingual scenario, demonstrating that language properties could be encoded using a low dimensional feature vector extracted in the notion of bottleneck features [8].

In addition to augmenting input features, another adaptation method has been proposed in [13]. The authors proposed to combine several networks into a larger network using a code to modulate the outputs of each individual network. We use a similar approach by modulating the output of the last LSTM layer in our setup.

2.3. Language Documentation

Documenting an unknown language poses several difficulties, among them the discovery of the language's phone inventory. Creating a phonetic transcription manually is a time consuming process that requires trained specialists. Next is the question of detecting allophones among the found phones in order to establish a phoneme inventory. While there are approaches to automating these tasks [3], it has been shown that the entire process is difficult, if not impossible, to automate. But there are methods that are able to discover distinct acoustic units in unknown languages. This was demonstrated by HMM based approaches like [14], more recent approaches using neural networks [15], but also GMM based methods [16] demonstrated as part of the Zero Resource Challenge [17]. We also demonstrated a first approach towards deriving a set of phone-like units [6], but we assumed the number of acoustic units to be known.

3. LANGUAGE ADAPTIVE DBLSTM BASED AF EXTRACTION

In the past, we reported on AF classification methods based on DNNs [18] and proposed a first approach using DBLSTMs [7]. In total, we used 7 different types of AFs, as shown in Table 1. These AFs can be divided into two classes: There are 3 types of AFs for consonants with the prefix '*c*' and 4 types of AFs for vowels with the prefix '*v*'. Similar to previous approaches, we trained individual networks for each AF

to prevent co-adaption to certain AF combinations that may occur in one language but not in another. As each AF applies to either consonants or vowels, we added to each AF an additional class indicating "does not apply". In order to obtain training data for AFs, we used a LVCSR system to force align transcriptions to recordings at phone level. This system used 3 sub-phones per phone: begin, middle and end. We trained the AF recognition networks only on sub-phones of type middle. We assume that for these sub-phones the articulators will reach their targets to the most extent during their continuous movement over the course of the speech production process[18, 4].

Table 1. Overview of AF types used

Type	# Classes	Description
cplace	8	Place of articulation
ctype	6	Type of articulation
cvox	2	Voiced
vfront	3	Tongue x position
vheight	3	Tongue y position
vlng	4	Type of vowel
vrnd	2	Lips rounded

In this work, we improve our DBLSTM based approach for AF detection from [7] by incorporating language feature vectors. Previous experiments have shown improvements for DNNs when appending LFVs to the input features. But appending additional features to the input might not be ideal for recurrent neural networks, especially if the features encode language specific peculiarities, which could be considered higher order features. A more suitable way of adding these features is to incorporate them deep into the recurrent network architecture. Our approach is based on the Meta-PI network[13], but differs in that we did not train mixture weights after training the network, but derived a language representation beforehand instead and used it to modulate the outputs of the last hidden BLSTM layer. The architecture is shown in Figure 1.

The LFVs have a dimensionality of 42, hence we choose the number of cells in each LSTM layer of the network to be a multiple of 42. This way, we could divide the cells into 42 groups of equal size. The outputs of each group was multiplied (modulated) with one dimension of the LFVs. To extract these features, we used the same setup as in [8]: A feed-forward DNN is trained to detect languages. The second last layer of this DNN is a bottle-neck layer. After training, we discarded the layers after this layer and used the output activations as LFVs. The setup for LFV extraction consisted of two networks. The first network was used to extract BNFs from acoustic input features. It was trained using phone states as targets and a combination of lMel and tonal features as input. Using these BNFs, we trained the network for language identification, which was then used to extract LFVs.

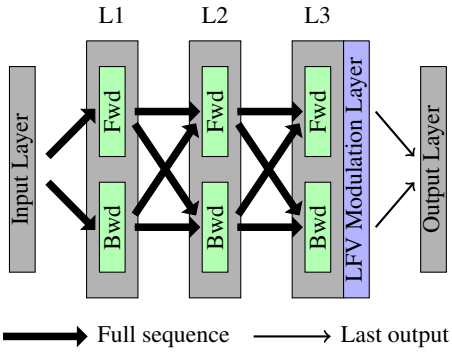

Fig. 1. DBLSTM network architecture. The full sequence gets propagated through LSTM layers. Only the final output is retained after the last LSTM layer, being modulated and forwarded to the output layer.

In preliminary experiments, we evaluated modulating the output of different hidden layers with LFVs, but given the resulting frame error rates, we chose to modulate the outputs of the last DBLSTM layer. This can be considered a regularization technique, similar to dropout. But instead of randomly omitting the output of entire neurons, we force groups of neurons to learn features depending on language properties by weighting their output accordingly.

We chose the hyper parameters of the network based on a setup for speech recognition [19]. The network featured 3 hidden bi-directional layers with 504 LSTM cells for each direction. Based on preliminary experiments [7], we chose a sequence covering a context of $+/-$ 15 frames. This results in a sequence length of 31, 15 frames of symmetric context on each side centered around one middle frame. The full sequence output for both directions gets concatenated and then forwarded to the next layer. After the final LSTM layer, we only retain the final sequence output, modulate it with LFVs and forward it to the output layer. The output layer is a feed forward layer which maps the output of the final DBLSTM layer to AF targets.

To further reduce the error rate, we applied a method similar to newbob scheduling: We restarted the network training with a decreased learning rate after an epoch if the recognition accuracy decreased. The training was restarted with the learning rate being decreased by a factor of 0.5.

4. PHONETIC UNIT DISCOVERY

Our goal is to discover phone-like units for unknown, potentially unwritten languages. Our proposed approach consists of three steps: First, we detect phone boundaries in the audio recordings of the new language, i.e. we segment the audio into phone-like segments without assigning labels to the segments. To do this we have demonstrated a method based

on DBLSTMs [20], achieving state-of-the-art performance on TIMIT.

In the second step, we classify AFs for each of these segments, using the method presented here. Ideally, we would have resorted to extracting AFs only on frames marked as middle frames by the speech recognizer, but since we do not have such an alignment, we approximated this by taking the inner third of each segment. While not being ideal, this should exclude co-articulation effects sufficiently well.

In the third step, the segments are clustered based on the AF detected for them in order to derive a set of phonetic units, using k-Means clustering [6]. This method requires the number of classes (phones) to be known beforehand, which is normally not the case in our scenario.

4.1. Evaluating the Cluster Count

To estimate the quality of the clustering, we used two methods. For determining the clustering performance in a supervised way, we computed the adjusted mutual information (AMI) score [21]. To evaluate the performance unsupervised, the Mel Cepstral Distortion (MCD) [22] was used. While it is primarily used to assess the quality of text-to-speech (TTS) systems, it is also suited to evaluate the discovered units: First, a TTS system is built based on the discovered units, next the distortion of the generated speech is computed [23]. The resulting MCD score serves as a measure of how distorted the generated speech is. Higher values correspond to high distortions and to a lower performance.

5. EXPERIMENTAL SETUP

5.1. Data Preparation

We based our experiments on the multilingual Euronews corpus [24], which consists of TV broadcast news. It features recordings from 10 languages, with 70h of data per language. For training and evaluation of AFs, we used data from English, French, German and Turkish. In addition to data from Euronews, we also evaluated our results on Mbosi, a language from the Bantu family. The data we used was collected as part of the BULB project [25]. The data was recorded in the field, and then later re-spoken by a native speaker in a controlled environment. To generate training data for the AF extractors, we trained speech recognition systems for English, French, German and Turkish and used these systems to force align the transcripts to the recordings at a phone level. This selection was based on the availability of pronunciations generated by MaryTTS [26], whose language definition files we also used to establish a mapping between phones and AFs. The systems were trained using the Janus Recognition Toolkit (JRTk) [27] which features the IBIS single-pass decoder [28]. The phone alignments were mapped to AFs in order to generate training data. For the clustering of individual segments into phone-like units and the evaluation of these clusterings, we used Scikit-learn [29].

5.2. Neural Network Training

As contrastive experiments, we trained multilingual bottleneck features. The network we used for these experiments was trained using data from French, German, Italian, Russian and Turkish. We explicitly excluded English, that we considered a faux low-resource language in this paper. It was trained multilingually with shared hidden layers, language specific output layers and context-dependent sub-phone states as targets. It featured 5 hidden layers with 1,600 neurons each, except for the second last layer which was the bottleneck layer with 42 neurons. We used lMel and tonal features (FFV [30] and pitch [31]), extracted using a 32ms window with a 10ms frame-shift.

Based on BNFs extracted via this network, we trained the network for LFV extraction, which was trained on data from all available languages within Euronews (except English). As input features, we stacked bottle-neck features using a context of 33 frames, but only using every 3rd frame. We used this increased context because the language information is long-term in nature, and benefits from a larger context window compared to systems trained for speech recognition. The second last layer was a bottle-neck layer, and the network was trained with the language identity as targets, encoded using a one-hot encoding. While the 5 hidden layers of the network featured 1,600 neurons, the bottle-neck had a size of 42. The setup was similar to [8].

Training the DBLSTM networks for AF extraction, we used lMel and tonal features. We used Adam to compute the weight updates with a mini-batch size of 256 and a sequence length of 31, which corresponds to a context of $+/-$ 15 frames around a central frame. To train our networks, we used a framework based on Lasagne [32] and Theano [33].

6. RESULTS

We divided the results section into two parts: First, we evaluated our proposed AF extraction method. Second, we used the extracted features to cluster phone-like segments into phone-like units. We evaluated different features, as well as clustering methods. We evaluated the best configuration on the Embosi data, demonstrating the application of our proposed approach in a real-world scenario.

6.1. Articulatory Feature Extraction

Tables 2 and 3 show the results for detecting AFs using multiple approaches. We added numbers from recent publications based on DNNs and DBLSTMs for reference. Adding LFVs improves the classification performance for both DNN and DBLSTM based approaches. Combining DBLSTMs with

LFVs results in the lowest FERs, throughout all AF types. This indicates that AFs are, although being more universal in comparison to a phone set, to a certain degree biased towards different languages.

Table 2. Classification error of AFs for consonants, being trained on German, French and Turkish using 70h per language. The results show the FER on the validation set.

Network Type	cplace	ctype	cvox
DNN	8.4	8.2	7.8
DNN + LFV	7.0	6.7	6.3
DBLSTM	5.7	6.4	7.1
DBLSTM + LFV	**5.0**	**5.3**	**5.0**

Table 3. Classification error of AFs for vowels, being trained on German, French and Turkish using 70h per language. The results show the FER on the validation set.

Network Type	vfront	vheight	vlng	vrnd
DNN	7.2	7.9	7.3	6.1
DNN + LFV	5.8	6.6	5.7	5.0
DBLSTM	6.1	6.0	6.9	5.7
DBLSTM + LFV	**4.8**	**5.2**	**4.6**	**4.0**

6.2. Phonetic unit discovery

As next step, we evaluated the use of different features for clustering phone-like units. Based on both AFs and DBNFs, we clustered segments using k-Means. Using k-Means clustering, the number of classes has to be determined prior to the clustering process. We compared the clustering performance with a supervised analysis on English, using the AMI score and by varying the number of classes.

As shown in Figure 2, the scores using DBNFs showed a plateau over a wide range of class counts, while the scores of the AF based clusterings show a peak at 33 classes. Although the peak does not represent the actual number of phones present in English (38), the score of the peak is close to the score of the actual number of classes, as shown in Table 4.

Table 4. AMI Score for clusterings using either DBNFs or AFs.

Feature Type	33 classes	38 classes
DBNFs	0.489	0.481
AFs	0.397	0.394

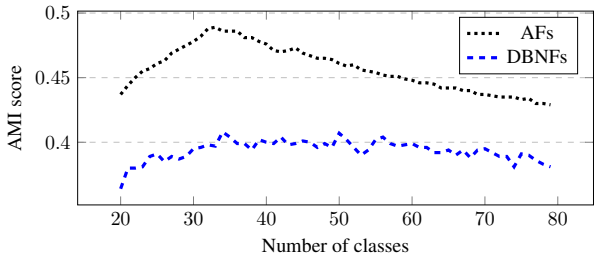

Fig. 2. Comparison of adjusted mutual information (AMI) scores using different class counts for k-Means clustering, with features based on DBNFs and AFs

6.3. Evaluation on Embosi

Based on extracted AFs, we evaluated our pipeline of segmentation and clustering on Embosi data. Based on the detected phones, we built a TTS system and computed the MCD score. As baseline, we used manually created phone labels. In comparison to that, we combined both the automatic segmentation and clustering ("Segment + Cluster"). As shown in Table 5, the MCD score rises from 5.25 to 5.78. While this indicates an increased distortion, the clustered units could be used to synthesize Mbosi speech.

Table 5. MCD Scores for different conditions

System	MCD Score
Baseline	5.25
Segment + Cluster	5.78

7. CONCLUSION

We have presented approaches for both AF extraction, as well as phonetic unit discovery. Using DBLSTMs in combination with LFVs, we could lower the FER of AF extractors. Regarding the clustering of phone-like segments into a set of phonetic units, we compared using DBNFs and AFs and showed that AFs are better suited for this task. Based on this approximation, a set of phone-like units could be derived. Because of language specific peculiarities, deriving the phone set is a task requiring linguists with expert knowledge. Hence, we did not aim at deriving the exact phone set, but instead to provide a first approximation. In addition we evaluated our pipeline using data from Embosi. Future work includes the evaluation of additional clustering methods and unsupervised metrics with the goal of deriving phonetic units more accurately in order to ease language documentation.

8. REFERENCES

[1] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig, "The microsoft 2016 conversational speech recognition system," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5255–5259.

[2] George Saon, Gakuto Kurata, Tom Sercu, Kartik Audhkhasi, Samuel Thomas, Dimitrios Dimitriadis, Xiaodong Cui, Bhuvana Ramabhadran, Michael Picheny, Lynn-Li Lim, et al., "English Conversational Telephone Speech Recognition by Humans and Machines," *arXiv preprint arXiv:1703.02136*, 2017.

[3] Timothy Kempton and Roger K Moore, "Discovering the Phoneme Inventory of an Unwritten Language: A Machine-Assisted Approach," *Speech Communication*, vol. 56, pp. 152–166, 2014.

[4] Florian Metze and Alex Waibel, "A Flexible Stream Architecture for ASR Using Articulatory Features," in *INTERSPEECH*, 2002.

[5] S. Stüker, T. Schultz, F. Metze, and A. Waibel, "Multilingual Articulatory Features," in *ICASSP*. 2003, vol. 1, pp. 144–147, IEEE.

[6] Markus Müller, Jörg Franke, Sebastian Stüker, and Alex Waibel, "Towards Phoneme Inventory Discovery for Documentation of Unwritten Languages," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017.

[7] Markus Müller, Jörg Franke, Sebastian Stüker, and Alex Waibel, "Improving Phoneme Set Discovery for Documenting Unwritten Languages," *Elektronische Sprachsignalverarbeitung (ESSV) 2017*, 2017.

[8] Markus Müller, Sebastian Stüker, and Alex Waibel, "Language Adaptive DNNs for Improved Low Resource Speech Recognition," in *Interspeech*, 2016.

[9] Florian Metze, *Articulatory Features for Conversational Speech Recognition*, Ph.D. thesis, Karlsruhe, Univ., Diss., 2005, 2005.

[10] Karen Livescu1, Özgür Çetin, Mark Hasegawa-Johnson, Simon King, Chris Bartels, Nash Borges, Arthur Kantor, Partha Lal, Lisa Yung, Ari Bezman, Stephen Dawson-Haggerty, Bronwyn Woods, Joe Frankel, Mathew Magimai-Doss, and Kate Saenko, "Articulatory Feature-Based Methods for Acoustic and Audio-Visual Speech Recognition: Summary from the 2006 JHU Summer Workshop," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007.* April 15–20 2007, IEEE.

[11] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny, "Speaker Adaptation of Neural Network Acoustic Models Using i-Vectors," in *ASRU*. IEEE, 2013, pp. 55–59.

[12] Yajie Miao, Hao Zhang, and Florian Metze, "Towards Speaker Adaptive Training of Deep Neural Network Acoustic Models," 2014.

[13] John B Hampshire and Alex Waibel, "The Meta-Pi Network: Building Distributed Knowledge Representations for Robust Multisource Pattern Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 751–769, 1992.

[14] Balakrishnan Varadarajan, Sanjeev Khudanpur, and Emmanuel Dupoux, "Unsupervised Learning of Acoustic Sub-Word Units," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*. Association for Computational Linguistics, 2008, pp. 165–168.

[15] Daniel Renshaw, Herman Kamper, Aren Jansen, and Sharon Goldwater, "A Comparison of Neural Network Methods for Unsupervised Representation Learning on the Zero Resource Speech Challenge," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[16] Michael Heck, Sakriani Sakti, and Satoshi Nakamura, "Unsupervised Linear Discriminant Analysis for Supporting DPGMM Clustering in the Zero Resource Scenario," *Procedia Computer Science*, vol. 81, pp. 73–79, 2016.

[17] Maarten Versteegh, Roland Thiolliere, Thomas Schatz, Xuan Nga Cao, Xavier Anguera, Aren Jansen, and Emmanuel Dupoux, "The Zero Resource Speech Challenge 2015," in *Proceedings of Interspeech*, 2015.

[18] Markus Müller, Sebastian Stüker, and Alex Waibel, "Towards Improving Low-Resource Speech Recognition Using Articulatory and Language Features," in *Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT)*, Seattle, U.S.A., 2016.

[19] Albert Zeyer, Ralf Schlüter, and Hermann Ney, "Towards Online-Recognition with Deep Bidirectional LSTM Acoustic Models," in *Proceedings of the Interspeech*, San Francisco, CA, USA, 2016.

[20] Jörg Franke, Markus Müller, Sebastian Stüker, and Alex Waibel, "Phoneme Boundary Detection using Deep Bidirectional LSTMs," in *Speech Communication; 12. ITG Symposium; Proceedings of*. VDE, 2016.

[21] Nguyen Xuan Vinh, Julien Epps, and James Bailey, "Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance," *Journal of Machine Learning Research*, vol. 11, no. Oct, pp. 2837–2854, 2010.

[22] Mikiko Mashimo, Tomoki Toda, Kiyohiro Shikano, and Nick Campbell, "Evaluation of cross-language voice conversion based on gmm and straight," 2001.

[23] Pallavi Baljekar, Sunayana Sitaram, Prasanna Kumar Muthukumar, and Alan W Black, "Using articulatory features and inferred phonological segments in zero resource speech processing," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[24] Roberto Gretter, "Euronews: A Multilingual Benchmark for ASR and LID," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[25] Sebastian Stüker et al., "Innovative Technologies for Under-Resourced Language Documentation: The BULB Project," in *CCURL 2016*, 2016.

[26] Marc Schröder and Jürgen Trouvain, "The German text-to-speech synthesis system MARY: A tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, no. 4, pp. 365–377, 2003.

[27] Monika Woszczyna et al., "JANUS 93: Towards Spontaneous Speech Translation," in *International Conference on Acoustics, Speech, and Signal Processing 1994*, Adelaide, Australia, 1994.

[28] Hagen Soltau, Florian Metze, Christian Fugen, and Alex Waibel, "A One-Pass Decoder Based on Polymorphic Linguistic Context Assignment," in *Automatic Speech Recognition and Understanding, 2001. ASRU'01. IEEE Workshop on*. IEEE, 2001, pp. 214–217.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[30] Kornel Laskowski, Mattias Heldner, and Jens Edlund, "The Fundamental Frequency Variation Spectrum," in *Proceedings of the 21st Swedish Phonetics Conference (Fonetik 2008)*, Gothenburg, Sweden, June 2008, pp. 29–32.

[31] Kjell Schubert, "Grundfrequenzverfolgung und deren Anwendung in der Spracherkennung," M.S. thesis, Universität Karlsruhe (TH), Germany, 1999, In German.

[32] Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, et al., "Lasagne: First release.," Aug. 2015.

[33] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016.