2016 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 13–16, 2016, SALERNO, ITALY

SPEAKER IDENTIFICATION AND CLUSTERING USING CONVOLUTIONAL NEURAL NETWORKS

Yanick Lukic, Carlo Vogt, Oliver Dürr, Thilo Stadelmann

Zurich University of Applied Sciences, Winterthur, Switzerland

ABSTRACT

Deep learning, especially in the form of convolutional neural networks (CNNs), has triggered substantial improvements in computer vision and related fields in recent years. This progress is attributed to the shift from designing features and subsequent individual sub-systems towards learning features and recognition systems end to end from nearly unprocessed data. For speaker clustering, however, it is still common to use handcrafted processing chains such as MFCC features and GMM-based models. In this paper, we use simple spectrograms as input to a CNN and study the optimal design of those networks for speaker identification and clustering. Furthermore, we elaborate on the question how to transfer a network, trained for speaker identification, to speaker clustering. We demonstrate our approach on the well known TIMIT dataset, achieving results comparable with the state of the art– without the need for handcrafted features.

Index Terms— Speaker Identification, Speaker Clustering, Convolutional Neural Network

1. INTRODUCTION

Automatic speaker recognition is an important key technology on the way to semantic multimedia understanding by machines. It comes in several flavors: For example, *speaker identification* refers to the task of inferring the speaker's identity of a new utterance, given a set of known voice models. *Speaker clustering* describes the task of telling who spoke when for a sequence of utterances, without prior knowledge of neither the number nor identities of speakers [1]. The clustering task is substantially more complex and hence studies show that this increased complexity leads to error rates an order of magnitude higher than for respective identification tasks even on very clean and plentiful data [2][3]. This paper is concerned with the advancement of pure speaker recognition capabilities in order to close this apparent gap, and therefore considers an experimental setup apart from additionally complicating application-specific effects (like e.g. channel mismatch, un-pure segmentation, background noise) to focus on the single question: *How to capture the essence of a voice reliably and robustly?*

Due to the multiscale nature of speech [4], this fundamental speaker recognition task per se poses hard challenges on pattern recognition systems: Speech segments not only convey the identity of a speaker, but also content (phonemes, forming words and sentences), emotion, origin (cultural, regional), health and age status (voices vary with the physiological condition of the vocal tract) as well as possibly background noise (channel characteristics, background sounds, interfering speech). The respective layers of information are convoluted into the single-dimensional time domain signal.

Traditionally, the speaker identification task has been approached using Gaussian Mixture Models (GMMs) on Mel Frequency Cepstrum Coefficient feature vectors (MFCCs) [5]. More recently, this framework has been extended using joint factor analysis [6] and intermediate vectors (i-vectors) [7] to form compact, fixed-length and maximally speaker-specific representations of an utterance. Despite being the state-of-the-art approach and well-working industry standard, this approach in principle has major shortcomings: Using MFCC feature vectors, the all-purpose answer for all audio analysis tasks [8], no specific voice-related characteristics of the speech signal despite the gross spectral envelope of short frames are exploited. Specifically, no speaker-discriminating features are sought, and some (as e.g. pitch information) are even knowingly neglected.

Speaker clustering (also called *speaker diarization* if segmentation into speaker-specific segments and clustering of these segments into speaker-specific groups is approached simultaneously) usually builds upon the same methods used for speaker identification [9]. Recent approaches rely on enriched input data: The very good results of [10] for rich transcription of e.g. meetings, lectures or TV programs are based on multiple distant microphone (multi-stream) processing techniques in order to cope with challenges like overlapping speech; other works incorporate additional modalities like accompanying video to extend the technology's application to scenario[s] much more difficult than the ones used so far [11]. These efforts have made speaker identification and clustering an application-ready technology in several domains of practical relevance. They have however done so by carefully engineering the respective systems to cope with certain challenges of the environment, e.g. the behavior of multiple

978-1-5090-0746-2/16/$31.00

speakers and interfering sound sources, *besides* improving the core voice recognition capabilities.

In this paper, we propose a novel solution to improve the speaker recognition pipeline. We apply Convolutional Neural Networks (CNNs) [12] on spectrograms in order to be able to learn speaker-specific features from a rich acoustic source representation. We evaluate our method on a well-known and acoustically easy database and experimental setup in order to show improvement in pure voice recognition capability apart from application-dependent nuisances. Initial results are on par with state of the art benchmark results, using a completely different and much less fine-tuned approach than conventional systems. The rest of this paper is organized as follows: Section 2 describes our approach and the relevant literature in detail. We start with the task of speaker identification and later move on to speaker clustering. Section 3 then reports on our experimental setup and results of a series of evaluations. Section 4 contains the conclusions and areas for future work.

2. A CNN APPROACH TO VOICE RECOGNITION

2.1. Related work

Convolutional neural networks are a variant of deep learning approaches that remove the necessity of handcrafted feature extraction on inputs with local correlation structure such as images or the spatio-temporal dependencies of spectrograms. They have become a standard method for many pattern recognition tasks at least since [13], including speech processing [14]. For example, Abdel-Hamid et al. show how the time-dependency of successive frames of speech relates to the convolutional filters in CNNs [15]. In [16], Lee et al. demonstrate (using a different type of Deep Learning model) how spectrograms can be used as input to diverse audio recognition tasks including speaker identification. McLaren et al. recently built a speaker recognition system that used senone probabilities computed by a CNN for speaker recognition, and compared it favorably with an i-vector approach [17].

Chen et al. trained a siamese deep network to directly compare two voice segments discriminatively [18]. Instead of using the feature learning capabilities of a CNN, they rely on MFCC features. Yella et al. also use the idea of speaker discriminative training and derive speaker-specific features from the hidden layer of a neural network of just 3 layers, which they subsequently feed into a GMM/HMM system for speaker clustering [19]. This idea of *speaker embeddings* is further explored by Rouvier, Bousquet and Favre with a non-convolutional deep network using a 61440 dimensional super-vector obtained from a Gaussian Mixture Universal Background Model (GMM-UBM) as input [20]. In contrast, we propose to learn the features directly from spectrograms by using CNNs trained for speaker identification and then use one of the post-convolutional layers as feature representation.

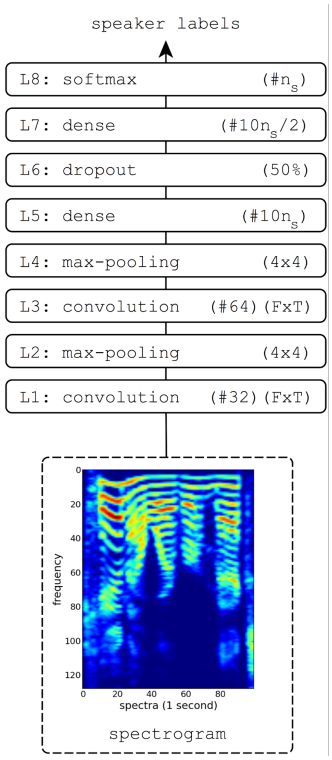

Fig. 1. The basic architecture we used in all our experiments. $F \times T$ in the convolution layers correspond to the frequency$\times$time filter applied.

2.2. Speaker identification by CNNs

CNNs extend the well-known idea from image processing of filters as weighted sums of pixels, by making the filter coefficients learnable. They consist of several such convolutional layers which apply a learned set of filters to subsequent small local parts of the input (e.g. a 3×3 area, which is then replicated over the whole input space). Each convolutional layer is followed by a max-pooling layer, that generates a lower resolution version of the convolutional layer's activations by taking the maximum filter activation out of e.g. a 2×2 window. This ensures some degree of shift and distortion invariance. Finally, fully connected layers combine all outputs of the last max-pooling layer to do the classification [14].

By using CNNs, we circumvent the necessity of strongly preprocessing the raw audio data and thereby losing possibly valuable information. In contrast, our approach enables the network to choose the necessary features specifically for the identification task from a wide range of available possibilities. Given labeled utterances, we are able to train the network from end to end (starting from spectrograms as in [16]).

Our architecture (see Fig. 1) is based on [21]: The network has 32 and 64 filters, respectively, for its two convolutional layers of size $F \times T$ for the frequency (F) and time (T) axis. We evaluate the optimal filter sizes F and T in section

3.1. Each convolutional layer is followed by a max-pooling layer with pooling size 4×4 and stride 2×2 (during evaluations, we also tried pooling and stride in only on direction). The network is completed by two dense layers with $10 \cdot n_s$ and $5 \cdot n_s$ neurons respectively, where n_s is the number of speakers to be identified. We use rectified linear units as activation functions in all layers [22] and apply softmax to the output. To prevent overfitting, we include a dropout layer between the dense layers with a dropout rate of 0.5. The basic architecture remains always the same, we only increase its complexity as we increase the number of speakers to be identified.

The input to the network is constructed as follows: First, we compute a mel-spectrogram with 128 elements in frequency direction for each sentence of the data set using the python library `librosa` [23]. We have 16kHz sampling rate, 1024 samples FFT window length and 160 samples as hop length. We then perform dynamic range compression of the spectrograms by applying the elementwise function $f(x) = \log(1 + C \cdot x)$ as in [21], with $C = 10'000$. Second, we extract one second long snippets of non-overlapping pieces from the spectrograms and use these images of 128×100 pixels the as basic input to the CNN. The network is trained using minibatch gradient descent (batch size 128) with Nesterov momentum, using cross-entropy as the loss function. To achieve a high diversity in the composition of the minibatches, a random sentence in the training set is selected, and from this a random snippet of one second is used. A minibatch consists of 128 such randomly chosen snippets. For building and training the CNN, we use the `Lasagne` library [24].

2.3. Speaker clustering by identification networks

For the speaker clustering task, we use a two-step approach: First, we train a standard speaker identification CNN as described above, but with a number of target speakers considerably larger than the expected maximal number of speaker clusters (e.g., 500 target speakers, if later on < 100 speaker clusters are expected). We then evaluate the clustering performance using either the activations of a specific hidden layer (L5 or L7 according to Fig. 1) or the final softmax layer L8 of this trained speaker identification network as speaker specific features for the clustering task. We build these representations by creating spectrograms for each utterance to be clustered, feeding all respective non-overlapping, subsequent, one second long snippets to the identification network and receiving the activations of either a hidden or the softmax layer.

The "why" of using any post-convolution layer's activation follows the same reasoning as with other embedding methods like e.g. word embeddings in text analysis [25]: The hidden layers gather relevant features to solve a related task (here: speaker identification), and thus their activations serve as higher-level representations of the subject (here: voice). To use the softmax layers activation instead can be motivated as follows: If a pre-trained network encounters an unknown speaker, the produced feature vector should show a probability distribution over multiple speakers (as in cohort modeling [26]), and different snippets from the same speaker should show a similar distribution.

As a second step, we use the following higher-level features as input to a standard clustering algorithm to arrive at a final partitioning and number of speakers of the utterances to cluster: We average individual snippets representations to construct a feature vector per utterance from the possibly multiple snippets of this utterance, and then cluster these utterance-vectors (see section 3.3 for details).

3. EXPERIMENTAL SETUP AND RESULTS

Our experiments replicate the exact setup of [2] on the TIMIT dataset [27] to allow for comparison. The dataset contains studio quality recordings of 630 speakers (192 female, 438 male), sampled at 16kHz, covering the eight major dialects of American English. Each speaker reads ten phonetically rich sentences, from which we use six for training, two for validation, and two for testing. We train CNNs to perform speaker identification on a subset of 100 speakers as well as on the full dataset.

We carry out three experiments: First, we examine the optimal convolutional filter dimension. Second, we evaluate the speaker identification performance of our network on the whole TIMIT test set using the optimal convolutional filters. Third, we evaluate the performance on the clustering task using the pre-trained speaker identification network.

3.1. Frequency- vs. time convolution

In this experiment, we evaluate the optimal dimension for our convolutional filters and the associated pooling and stride of the max-pooling layers (L2 and L4). The first convolutional filter does convolutions of $F \times T$ pixels in the frequency and time domain, respectively, and thus is active for patterns of maximal extension $F \times T$. Filters along the different directions (time only, frequency only, and both) have been tested, together with varying sizes among these directions. To be able to quickly test different filters we used a network for identifying only 10 speakers. Filters with pooling and stride in both directions achieve a good result on the identification task. When using filters only in time or frequency direction with corresponding single-directional pooling and stride, comparable results are reached, but the networks get much bigger. Astonishingly, a filter in frequency direction with pooling and stride in both directions also accomplishes good results. A filter in time direction with pooling and stride in both directions on the other hand performs clearly worse.

A 2D filter with pooling and stride in both directions is suitable to capture features of both dimensions, which makes it intuitively more capable and more easily interpretable.

Additionally, it results in considerably smaller networks (because of the 2D dimensionality reduction during pooling). Thus, we chose a 4×4 filter with pooling 4×4 and stride 2×2 for our other experiments.

3.2. Speaker identification performance

In this experiment we show the results for the identification task using all speakers from the TIMIT dataset. To ensure enough training data, we use 8 of the 10 sentences for each speaker for training. The remaining 2 sentences, corresponding to approximately 5 seconds, are used as test data. We do not use any validation data in this experiment. We split the test data for one speaker into non-overlapping snippets of one second length (if the remainder of an utterance is shorter than one second, it is ignored). These snippets are then fed to the network, and the arithmetic mean over all output vectors is calculated. The element with the highest value corresponds with the assigned speaker.

We achieve an accuracy of 97.0%, corresponding to 19 misidentified speakers. Using the geometric instead of the arithmetic mean yields similar results, whereas using the maximum performs slightly worse. To our knowledge, this is the highest accuracy achieved using neural networks for this task on this dataset. However, other works achieve nearly perfect accuracy by using GMMs [28][29].

3.3. Speaker clustering

Based on our identification network, we evaluate here whether it is possible to cluster unknown speakers using the activations of an upper (dense or softmax) layer of a pre-trained identification CNN as a feature vector. Fig. 2 visualizes the individual output vectors produced by the snippets from 5 unknown speakers (i.e., never encountered during original indentification-targeted training) for the first dense layer L5 and the softmax layer L8 in a network trained to recognize 100 speakers, using t-SNE [30] with cosine metric.

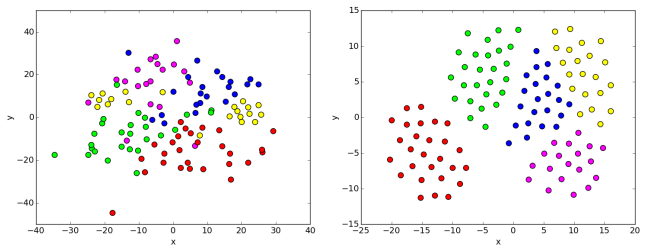

Fig. 2. t-SNE plots based on the output vectors of the softmax layer L8 (left) and the first dense layer L5 (right). Different colors correspond to different speakers.

For each speaker we have about 25 seconds of audio corresponding to 25 snippets. It is clearly visible that segments of the same speaker cluster together, and especially for the first dense layer L5 a good separation is observed.

Next, we examine the quality of this clustering approach when applying hierarchical clustering to a higher number of speakers. We use the misclassification rate (MR) to evaluate cluster quality [31]:

$$MR = \frac{1}{N} \sum_{j=1}^{N_s} e_j. \quad (1)$$

Where N is the total number of audio segment, N_s the number of speakers and e_j the number segments of speaker j that are assigned incorrectly. The MR takes ranges from 0 (perfect assignment) to 1 (all segments are wrongly assigned). We train identification networks on a training set of 100 and 590 speakers for comparison, and calculate the output vectors for all non-overlapping snippets of sentences for never encountered during training test sets of 20 and 40 speakers. In order to incorporate the knowledge of snippets belonging to certain sentences, we build the mean over all segments of 8 (=first utterance) and 2 (=second utterance) sentences for each speaker, which results in 2 feature vectors for each speaker. This is in accordance with [2], whose experimental setup we adopt in detail in order to be comparable.

Agglomerative hierarchical clustering is then applied to these speaker representations using complete linkage with the cosine metric. The results in Tab. 1 are obtained for the optimal cut-off performing a stepwise hierarchical clustering. The best result for 20 speakers with a MR of 0.1 is visualized using a dendrogram in Fig. 3.

	20 speakers		**40 speakers**	
Layer	**MR 100**	**MR 590**	**MR 100**	**MR 590**
L5: dense	0.100	0.100	0.300	0.125
L7: dense	0.100	**0.100**	0.325	**0.050**
L8: softmax	0.450	0.250	0.700	0.450

Table 1. Results of the clustering experiment showing the misclassification rate for representations taken from different upper layers of two networks trained on 100 and 590 speakers. The clustered speakers were unknown to both networks.

As suggested already by the results in Fig. 2, MR is optimized when using the output of dense layers above the convolution and below the final softmax. We attribute this to the fact that the hidden layers L5 and L7 portray a general representation of a speaker (because of the discriminative training to segregate speakers), and L7 is the highest-level such representation. The softmax layer L8 on the other hand represents a speaker by a probability distribution that can be likened to a linear combination of a multitude of unrelated speakers spanning the speaker space. The best results are obtained by

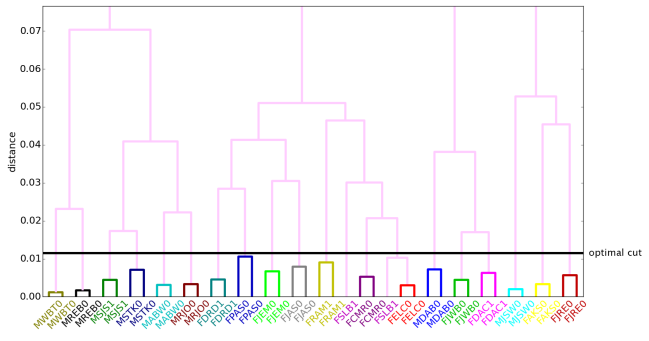

Fig. 3. The dendrogram shows the hierarchical clustering performed on 20 speakers based on the activations of the second hidden layer L7 of an identification network trained on a training set of 590 speakers. Each leaf node corresponds to the mean over all segments of either 8 or 2 sentences of the speaker. The optimal cut is set where the MR reaches its lowest value (in this case $MR = 0.1$). A label starting with *M* denotes a male speaker, *F* denotes female, and the colors correspond to different speakers.

clustering on the outputs of the second dense layer L7 of a network trained for classifying many (590) speakers: On the dataset with 20 speakers we achieve a MR of 0.1 and with 40 speakers 0.05. In [2] a MR of 0.0 and 0.065 was reported respectively using a MFCC-GMM baseline of 0.125, showing that our results using the speaker embedding (L7) are on par, and the cohort modeling (L8) clearly worse.

4. CONCLUSION AND FUTURE WORK

In this paper, we have investigated whether it is feasible to identify speakers using features generated by a CNN, and to cluster unknown speakers using the activations of post-convolutional layers of pre-trained speaker identification CNNs. The clustering performance of this approach is on par with todays best systems, when using the output of the high level dense layers (speaker embedding) instead of the softmax layer (cohort modeling). This is remarkable because no speech-specific preprocessing has been applied (e.g., silence removal, detection and removal of unvoiced speech): the presented approach is rather un-tuned. We thus effectively showed that the learned features by the CNN are relevant to recognize unknown speakers, with potential for future improvements.

Future work will consist of further exploring speaker recognition as a sequence learning task using representation learning approaches like recurrent neural networks. It is particularly interesting how discriminative training can be applied to the per se unsupervised learning task of speaker clustering, e.g. using a siamese architecture. Additionally, further investigations concerning the directions of the convolutional filters, pooling and stride are valuable to determine why 1D and 2D operations seem to perform comparable to each other, and how such filters could be interpreted in terms of auditory processing of the time-evolution of speech.

5. REFERENCES

[1] Homayoon Beigi, *Fundamentals of speaker recognition*, Springer Science & Business Media, 2011.

[2] Thilo Stadelmann and Bernd Freisleben, "Unfolding speaker clustering potential: a biomimetic approach," in *Proceedings of the 17th ACM international conference on Multimedia*. ACM, 2009, pp. 185–194.

[3] Mark Sinclair and Simon King, "Where are the challenges in speaker diarization?," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7741–7745.

[4] Marwan Al-Akaidi, *Fractal speech processing*, Cambridge University Press, 2004.

[5] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.

[6] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[7] Hung-Shin Lee, Yu Tsao, Hsin-Min Wang, and Shyh-Kang Jeng, "Clustering-based i-vector formulation for speaker recognition.," in *INTERSPEECH*, 2014, pp. 1101–1105.

[8] Martin F McKinney and Jeroen Breebaart, "Features for audio and music classification.," in *ISMIR*, 2003, vol. 3, pp. 151–158.

[9] Jitendra Ajmera and Chuck Wooters, "A robust speaker clustering algorithm," in *Automatic Speech Recognition and Understanding, 2003. ASRU'03. 2003 IEEE Workshop on*. IEEE, 2003, pp. 411–416.

[10] Gerald Friedland, Adam Janin, David Imseng, Xavier Anguera Miro, Luke Gottlieb, Marijn Huijbregts, Mary Tai Knox, and Oriol Vinyals, "The icsi rt-09 speaker diarization system," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 2, pp. 371–381, 2012.

[11] I Kapsouras, A Tefas, N Nikolaidis, G Peeters, L Benaroya, and I Pitas, "Multimodal speaker clustering in full length movies," *Multimedia Tools and Applications*, pp. 1–20, 2016.

[12] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[14] Yann LeCun and Yoshua Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, pp. 1995, 1995.

[15] Ossama Abdel-Hamid, Li Deng, and Dong Yu, "Exploring convolutional neural network structures and optimization techniques for speech recognition.," in *INTERSPEECH*, 2013, pp. 3366–3370.

[16] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in neural information processing systems*, 2009, pp. 1096–1104.

[17] Mitchell McLaren, Yun Lei, Nicolas Scheffer, and Luciana Ferrer, "Application of convolutional neural networks to speaker recognition in noisy conditions.," in *INTERSPEECH*, 2014, pp. 686–690.

[18] Ke Chen and Ahmad Salman, "Learning speaker-specific characteristics with a deep neural architecture," *Neural Networks, IEEE Transactions on*, vol. 22, no. 11, pp. 1744–1756, 2011.

[19] Sree Harsha Yella, Andreas Stolcke, and Malcolm Slaney, "Artificial neural network features for speaker diarization," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 402–406.

[20] Mickael Rouvier, Pierre-Michel Bousquet, and Benoit Favre, "Speaker diarization through speaker embeddings," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 2015, pp. 2082–2086.

[21] Sander Dieleman and Benjamin Schrauwen, "End-to-end learning for music audio," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6964–6968.

[22] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[23] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference*, 2015.

[24] Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Sren Kaae Snderby, Daniel Nouri, Daniel Maturana, Martin Thoma, Eric Battenberg, Jack Kelly, Jeffrey De Fauw, Michael Heilman, diogo149, Brian McFee, Hendrik Weideman, takacsg84, peterderivaz, Jon, instagibbs, Dr. Kashif Rasul, CongLiu, Britefury, and Jonas Degrave, "Lasagne: First release.," Aug. 2015.

[25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[26] Aaron E Rosenberg, Joel DeLong, Chin-Hui Lee, Biing-Hwang Juang, and Frank K Soong, "The use of cohort normalized scores for speaker verification," in *Second international conference on spoken language processing*, 1992.

[27] William M Fisher, George R Doddington, and Kathleen M Goudie-Marshall, "The darpa speech recognition research database: specifications and status," in *Proc. DARPA Workshop on speech recognition*, 1986, pp. 93–99.

[28] Douglas A Reynolds, "Speaker identification and verification using gaussian mixture speaker models," *Speech communication*, vol. 17, no. 1, pp. 91–108, 1995.

[29] Andrew Morris, Dalei Wu, and Jacques Koreman, "Gmm based clustering and speaker separability in the timit speech database," *IEICE Transactions on Fundamentals of Communications, Electronics, Informatics and Systems*, vol. 85, 2004.

[30] Laurens Van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, pp. 85, 2008.

[31] Margarita Kotti, Vassiliki Moschou, and Constantine Kotropoulos, "Speaker segmentation and clustering," *Signal processing*, vol. 88, no. 5, pp. 1091–1124, 2008.