CYCLE-GANS FOR DOMAIN ADAPTATION OF ACOUSTIC FEATURES FOR SPEAKER RECOGNITION

Phani Sankar Nidadavolu, Jesús Villalba, Najim Dehak

Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, USA

{snidada1,jvillal7,ndehak3}@jhu.edu

ABSTRACT

It is well known that domain mismatch between the training and evaluation data hinders the performance of any machine learning system. Various factors contribute to domain mismatch. In speaker recognition systems, it mainly occurs due to the mismatch in recording conditions and language. Most speaker recognition corpora are telephone speech. Meanwhile, a few evaluation data sets like Speakers In The Wild (SITW) are microphone speech. In this work, we explore domain adaptation at acoustic feature level by learning feature mappings between domains using cycle consistent generative adversarial networks (cycle-GANs), without any parallel data between domains. Microphone features mapped to telephone domain are used to evaluate speaker recognition system trained only on telephone data. We achieved 9.37% and 2.82% relative improvement in equal error rate (EER) and detection cost function (DCF) on SITW eval set.

***Index Terms*—** Unsupervised domain adaptation, speaker recognition, cycle-GANs, generative adversarial neural networks (GANs)

1. INTRODUCTION

NIST speaker recognition evaluations have mainly driven speaker recognition research in the past few years. Because of the target application of these evaluations, most data available to train speaker recognition systems is telephone speech. The growing interest to apply speaker recognition to microphone speech has promoted the emergence of new databases like Speakers In The Wild (SITW) [1]. This implies a new challenge for speaker recognition trained on telephone speech because of the mismatch between the train and evaluation domains. The authors in [2] approached this problem by augmenting the telephone data with microphone speech from the VoxCeleb dataset [3] and obtained a significant improvement on SITW. However, this approach requires access to labelled in domain data sets, a time consuming and an expensive approach. On the other hand, getting access to unlabelled microphone speech is relatively easy.

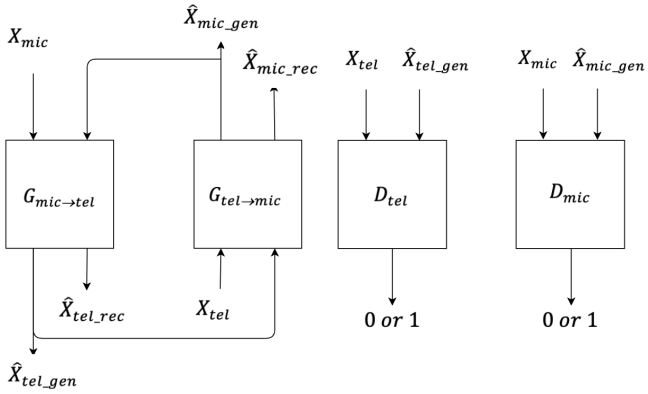
In this work, we assumed we have access to unlabelled nonparallel microphone speech data for adaptation. We leveraged unlabelled microphone data along with the telephone data, used to train speaker recognition systems, to learn feature mappings between microphone and telephone domains. Following that, we mapped the microphone speech features– from SITW– to telephone domain. Finally, these mapped features were used in the evaluation phase (enrollment and test), with a x-vector system[2], trained on actual telephone data. The feature mapping functions were implemented by using recently proposed cycle-GANs [4], variant of the original generative adversarial networks (GANs) [5]. In computer vision, cycle-GANs were proposed to learn mapping functions of images between domains with non parallel data. Cycle-GANs soon found their way to speech research where they were used for adapting automatic speech recognition (ASR) trained on clean speech to noisy speech [6, 7], voice conversion [8, 9, 10], and gender adaptation [11]. To the best of our knowledge, we are the first ones to use cycle-GANs for telephone-microphone channel domain adaptation for improved speaker recognition.

The rest of the paper is organized as follows. In Section 2, we introduce our feature adaptation mechanisms which use cycle-GANs. Section 3 shows several alternate architectures for the generator. In Section 4, we discuss the experimental setup and in Section 5, we show the results. We conclude with a summary of the paper and future work in Section 6.

2. CYCLE-GANS FOR DOMAIN ADAPTATION

Data for cycle-GANs training consists of features $\mathbf{X}_{\text{tel}} = \{\mathbf{x}_{\text{tel},i}\}_{i=1}^{N}$ and $\mathbf{X}_{\text{mic}} = \{\mathbf{x}_{\text{mic},i}\}_{i=1}^{M}$, which are drawn from two different distributions $\mathbf{x}_{\text{tel},i} \sim p_{\text{tel}}(\mathbf{x})$ and $\mathbf{x}_{\text{mic},i} \sim p_{\text{mic}}(\mathbf{x})$. No speaker labels from either domains are needed to train the feature mapping system.

Cycle-GANs architecture in Figure 1 comprises two generators and two discriminators. The generator $G_{\text{tel}\to\text{mic}}$ transforms telephone domain features $\mathbf{X}_{\text{tel}}$ to microphone domain, producing features $\hat{\mathbf{X}}_{\text{mic\_gen}}$. The discriminator D_{mic}, is paired to $G_{\text{tel}\to\text{mic}}$ to discriminate between the generated $\hat{\mathbf{X}}_{\text{mic\_gen}}$ and original $\mathbf{X}_{\text{mic}}$ microphone features. Equivalently, the other generator-discriminator ($G_{\text{mic}\to\text{tel}}$, D_{tel}) pair is intended to transfer features from microphone to telephone domain.



(a) Generator Pair (b) Discriminator Pair

Fig. 1: Cycle-GANs architecture

In the Cycle-GANs framework, the generators and discriminators are trained using a combination of loss functions, i.e., adversarial, cycle consistency and identity loss. In the adversarial loss, the discriminator minimizes the classification error between real and transferred samples, while the generator tries to maximize it. For the discriminator loss, we observed that mean square error provided better performance than the typical cross-entropy, as shown in [12]. Thus, to train $G_{\text{mic}\to\text{tel}}$ and D_{tel}, we optimize

$$L_{\text{GAN}}(G_{\text{mic}\to\text{tel}}, D_{\text{tel}}, \mathbf{X}_{\text{mic}}, \mathbf{X}_{\text{tel}}) = \quad (1)$$
$$\mathbb{E}_{\mathbf{x}\sim p_{\text{tel}}}[(D_{\text{tel}}(\mathbf{x})-1)^2] + \mathbb{E}_{\mathbf{x}\sim p_{\text{mic}}}[D_{\text{tel}}(G_{mic\to\text{tel}}(\mathbf{x})^2)] \ ,$$

where we minimize w.r.t. D_{tel} and maximize w.r.t. $G_{\text{mic}\to\text{tel}}$ Equivalently, for $G_{\text{tel}\to\text{mic}}$ and D_{mic}, we optimize $L_{\text{GAN}}(G_{\text{tel}\to\text{mic}}, D_{\text{mic}}, \mathbf{X}_{\text{tel}}, \mathbf{X}_{\text{mic}})$.

A single generator-discriminator pair, trained with adversarial loss, would suffice to transfer features from microphone to telephone domain. However, this leads to an ill poised problem with adversarial loss putting a weak constraint on the generators. Thus, the generator could create many possible features which appear to be drawn from the true distributions but that differ from it significantly. To restrict the space of possible mappings from the generator, cycle-GANs enforce cycle consistency constraint on the generators. Cycle consistency consists of reconstructing the original features, e.g. $\mathbf{X}_{\text{mic}}$, from the adapted features in the opposite domain, e.g., $\hat{\mathbf{X}}_{\text{tel\_gen}}$. That means to minimize the error between $\mathbf{X}_{\text{mic}}$ and $\hat{\mathbf{X}}_{\text{mic\_rec}} = G_{\text{tel}\to\text{mic}}(\hat{\mathbf{X}}_{\text{tel\_gen}})$. Considering cycle consistency in both directions, the loss is

$$L_{\text{cyc}}(G_{\text{mic}\to\text{tel}}, G_{\text{tel}\to\text{mic}}) = \quad (2)$$
$$\mathbb{E}_{\mathbf{x}\sim p_{\text{tel}}}[||G_{\text{mic}\to\text{tel}}(G_{\text{tel}\to\text{mic}}(\mathbf{x}))-\mathbf{x}||_1]$$
$$+ \mathbb{E}_{\mathbf{x}\sim p_{\text{mic}}}[||G_{\text{tel}\to\text{mic}}(G_{\text{mic}\to\text{tel}}(\mathbf{x}))-\mathbf{x}||_1] \ ,$$

where we used L1 distance as metric.

Finally, we added another loss taking into account that when samples from the output domain are presented as input to the generators, the generators should give an identity mapping [13]. We found this to be a good regularizer. Identity loss is expressed as

$$L_{\text{idt}}(G_{\text{mic}\to\text{tel}}, G_{\text{tel}\to\text{mic}}) = \quad (3)$$
$$\mathbb{E}_{\mathbf{x}\sim p_{\text{tel}}}[||G_{\text{mic}\to\text{tel}}(\mathbf{x})-\mathbf{x}||_1]$$
$$+ \mathbb{E}_{\mathbf{x}\sim p_{\text{mic}}}[||G_{\text{tel}\to\text{mic}}(\mathbf{x})-\mathbf{x}||_1]$$

Combining all the objectives, we have

$$L(G_{\text{mic}\to\text{tel}}, G_{\text{tel}\to\text{mic}}, D_{\text{mic}}, D_{\text{tel}}) = \quad (4)$$
$$L_{\text{GAN}}(G_{\text{mic}\to\text{tel}}, D_{\text{tel}}, X_{\text{mic}}, X_{\text{tel}})$$
$$+ L_{\text{GAN}}(G_{\text{tel}\to\text{mic}}, D_{\text{mic}}, X_{\text{tel}}, X_{\text{mic}})$$
$$- \lambda_c L_{\text{cyc}}(G_{\text{mic}\to\text{tel}}, G_{\text{tel}\to\text{mic}})$$
$$- \lambda_i L_{\text{idt}}(G_{\text{mic}\to\text{tel}}, G_{\text{tel}\to\text{mic}})$$

where λ_c and λ_i control the relative importance given to their respective objectives. The discriminators are trained to minimize the objective function where as the generators are trained to maximize it.

3. GENERATOR AND DISCRIMINATOR ARCHITECTURES

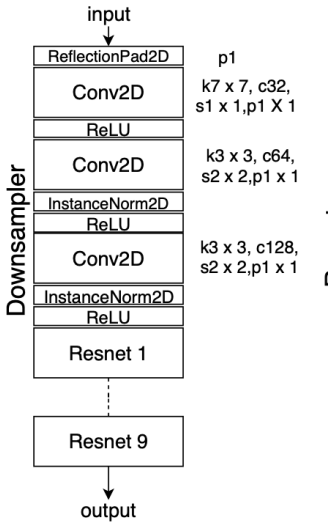
We experimented with three different cycle-GANs architectures. They differ on how the generators were built. All the generators used in this work were built using two building blocks: a downsampler and an upsampler. The downsampler was realized using a series of 2D convolutional operations, non linear layers and few residual blocks. We used 9 residual blocks in this work. The second and third convolutional layers have stride 2 on both axes, which reduces the dimension of output feature maps. Hence, the name downsampler. The upsampler block is realized using two deconvolutional layers followed by a final convolution layer. The deconvolutional layers increase the dimension of the feature maps by applying a stride $\frac{1}{2}$. The architectures for the upsampler, residual block and the downsampler are given in Figure 2. For all the three architectures the discriminator architecture remains unchanged. We explain below the architectures of the three cycle-GANs used in this work.

3.1. Cycle-GANs system A

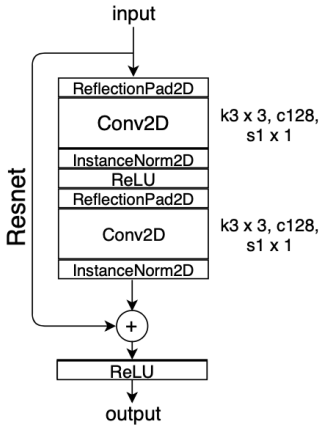
For this generator, the configuration in Figure 3a was used. This is the most common architecture used in cycle-GANs. The generator accepts inputs from one domain which by passing through the downsampler and upsampler maps those features to the opposite domain.

3.2. Cycle-GANs system B

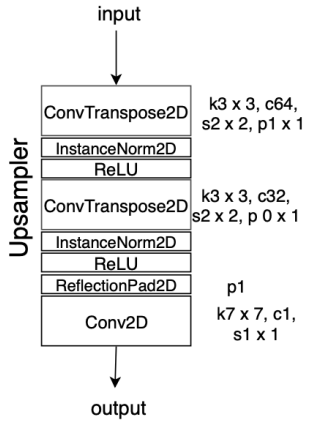
For this generator, the configuration in Figure 3b was used. The output $\mathbf{y}$ of the generator given the input $\mathbf{x}$ is $\mathbf{y} = \alpha\mathbf{x} +$



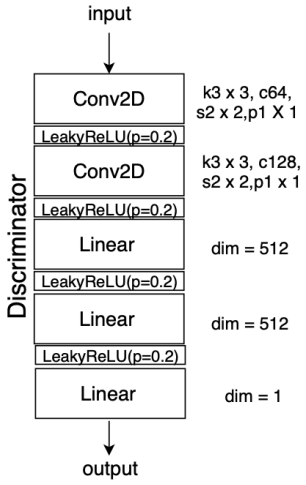
(a) Downsampler network.



(b) Residual Network



(c) Upsampler Network.



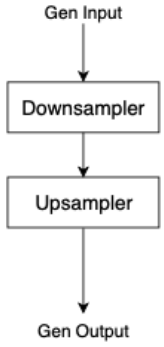
(d) Discriminator Network.

Fig. 2: Architectures of individual blocks of cycle-GAN.

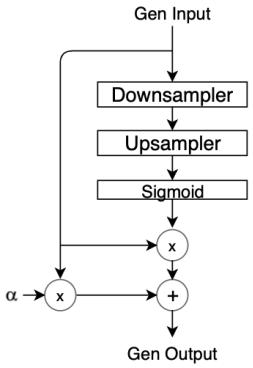
$\text{sigmoid}(\mathbf{u}) \circ \mathbf{x}$, with $0 < \alpha < 1$, where $\mathbf{u}$ denotes the output of the upsampler block. $\circ$ denotes element wise multiplication. Here, the purpose of the downsampler/upsampler blocks is to compute a mask, which acts as a kind of filter. The addition with the scaled input is to avoid the possibility of obtained null components of the spectrum.

3.3. Cycle-GANs system C

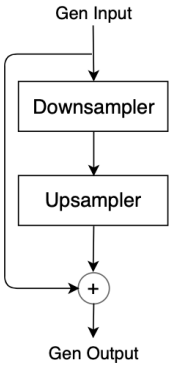
For this generator, the configuration in Figure 3c was used. The generator input is added to the output of the upsampler which becomes the final output of the generator. This will ensure that structure in the input data is preserved at the generator output. This architecture was inspired by the work done for speech enhancement by [6].



(a) Cycle-GANs System A.



(b) Cycle-GANs System B.



(c) Cycle-GANs System C.

Fig. 3: Different Generators for cycle-GANs system

4. EXPERIMENTAL SETUP

4.1. Datasets

We experimented adapting the microphone speech of the SITW evaluation set to telephone domain. Then, we evaluated the transformed SITW using an x-vector system trained on telephone speech. The telephone data consisted of recordings from SRE04-10, Mixer6 and Switchboard 1-Phase 1,2 and 3. Together, they account to 90946 utterances from 6986 number of speakers. The microphone data used for adaptation consisted of the SITW development set and VoxCeleb database. Together they consist of 24581 utterances. There is no speaker overlap between the development data and evaluation data for SITW. The cycle-GANs training did not involve any speaker label information.

4.2. Cycle-GANs training

The cycle-GANs are implemented using PyTorch [4]. Two mini batches of features, one from telephone speech and other from microphone speech, were sampled randomly from their respective data sets during each training step. The minibatch sizes were set to 256 and the number of contiguous frames sampled from each utterance was set to 11. Since we used 2D convolutional neural networks, all the mini batches were

Table 1: Comparison of Cycle-GANs domain adaptation on the SITW eval set.

	SITW Core			SITW Assist-Multi		
	EER	**DCF(1E-2)**	**DCF(1E-3)**	**EER**	**DCF(1E-2)**	**DCF(1E-3)**
Baseline	10.14	0.6842	0.8171	12.72	0.6941	0.8179
Cycle-GAN						
System A	10.81	0.6852	0.8214	13.48	0.6930	0.8242
System B	9.63	0.6758	0.8212	12.19	0.6822	0.8119
System C	**9.19**	**0.6649**	**0.8170**	**11.51**	**0.6797**	**0.8105**

arranged as four dimensional tensors of size (256, 1, 11, 40), 40 being the dimension of filter bank features. The model was trained for 100 epochs. Each epoch is set to be complete when all the telephone utterances have appeared once in that epoch. Utterances were sampled in random order. Adam Optimizer was used with momentum $\beta_1 = 0.5$ as suggested by [14]. The learning rates for the generators and discriminators were set to 0.0003 and 0.0001 respectively. The learning rates were kept constant for the first 15 epochs and, then, linearly decreased until they reach the minimum learning rate (1e-6). For cycle-GANs system B and C loss weights λ_c and λ_i from (4) were set to 2.5 and 0.0 respectively. For cycle-GANs system A λ_c and λ_i were set to 10.0 and 5.0 respectively. α value for training cycle-GANs system B was set to 0.7.

4.3. x-Vector system

The x-vector system was based on Kaldi [15]. We used the same setup as in SRE16 Kaldi recipe[1] but without any data augmentation. The data augmentation is done using artificially adding noise to speech which introduces new domains (noise types). Our main goal of interest is to do domain adaptation across microphone and telephone domains. Hence, we decided not to use data augmentation. The system used 40 dimensional Mel filter-bank features with short-time centering. Microphone speech was downsampled from 16kHz to 8kHz. For the systems with adaptation microphone speech filter-banks were transformed to telephone domain using the $G_{\mathrm{mic}\to\mathrm{tel}}(\mathbf{x})$ generator. The x-vectors were centered, projected to 150 dimension using linear discriminant analysis (LDA) and length normalized. Full-rank probabilistic linear discriminant analysis (PLDA) [16] was used to get the scores. Finally, scores were normalized using adaptive symmetric norm (S-Norm) [17].

5. RESULTS

5.1. Comparison of different feature mapping systems

Table 1 gives a comparison of speaker ID results for all the systems. All the systems were evaluated on the core and the assist-multi (multiple speakers in enroll and test) conditions of SITW database [1]. Domain adaptation with system A did not improve the performance w.r.t. the baseline. System B achieved relative improvement of 5.03% and 1.23% over the baseline in terms of EER and DCF(1E-2). System C achieved relative improvement of 9.37% and 2.82% in terms of EER and DCF(1E-2) over the baseline.

[1] https://github.com/kaldi-asr/kaldi/tree/master/egs/sre16/v2

Table 2: Adapting SITW vs Adapting telephone training data.

	SITW Core			SITW Assist-Multi		
	EER	**DCF(1E-2)**	**DCF(1E-3)**	**EER**	**DCF(1E-2)**	**DCF(1E-3)**
Cycle-GANs system C						
Adapt SITW with $G_{\mathrm{mic}\to\mathrm{tel}}$	9.19	0.6649	0.8170	11.51	0.6797	0.8105
Adapt training data with $G_{\mathrm{tel}\to\mathrm{mic}}$	**9.11**	**0.6631**	**0.8072**	**11.47**	**0.6736**	**0.8071**

5.2. Adapting SITW features vs adapting telephone training features

So far we have discussed about mapping the features from microphone to telephone domain and evaluating on the x-vector model trained on telephone data. However, we can also use the generator $G_{\mathrm{tel}\to\mathrm{mic}}$ to map all the telephone features to microphone domain and train the x-vector model on the mapped features. SITW features remains unchanged when evaluating the model. Table 2 compares both experiments. Adapting telephone data to microphone domain obtained larger improvements (10.16% reduction in EER and 3.08% reduction in DCF(1E-2)). However, since the same telephone data used to train the mapping system is used to train the x-vector system this could result in over-fitting. Hence, this requires further investigation by training the feature mapping systems and x-vector on different telephone data.

6. SUMMARY

We explored the usage of cycle-GANs for learning feature mapping functions across telephone and microphone domains without any parallel data between both domains. We explored three different configurations for the cycle-GANs, each different from the way the generators are defined. The main challenge that we observed was to transfer features to another domain while preserving the structure (like lingusitic information, speaker information, etc). The best results were obtained when the generators were trained to learn a residual mapping between its input and output (cycle-GANs system C). By mapping the features of SITW–a microphone speech corpus– to telephone domain and testing those features on a speaker recognition system trained completely on telephone data, we obtained a reduction of 9.37% and 2.82% in terms of EER and DCF(1E-2) w.r.t. the baseline system without adaptation. Furthermore, by mapping the telephone data to microphone domain and retraining the x-vector extractor, we obtained a 10.16% and 3.08% respectively. In the future, we intend to incorporate attention mechanisms in the feature mapping functions and using speaker labels from telephone domain while training cycle-GANs.

7. REFERENCES

[1] Mitchell McLaren, Luciana Ferrer, Diego Castan, and Aaron Lawson, "The speakers in the wild (sitw) speaker recognition database.," in *Interspeech*, 2016, pp. 818–822.

[2] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *Submitted to ICASSP*, 2018.

[3] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[4] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint*, 2017.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[6] Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara, "Cross-domain speech recognition using nonparallel corpora with cycle-consistent adversarial networks," in *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*. IEEE, 2017, pp. 134–140.

[7] Zhong Meng, Jinyu Li, Yifan Gong, et al., "Cycle-consistent speech enhancement," *arXiv preprint arXiv:1809.02253*, 2018.

[8] Takuhiro Kaneko and Hirokazu Kameoka, "Parallel-data-free voice conversion using cycle-consistent adversarial networks," *arXiv preprint arXiv:1711.11293*, 2017.

[9] Fuming Fang, Junichi Yamagishi, Isao Echizen, and Jaime Lorenzo-Trueba, "High-quality nonparallel voice conversion based on cycle-consistent adversarial network," *arXiv preprint arXiv:1804.00425*, 2018.

[10] Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion with star generative adversarial networks," *arXiv preprint arXiv:1806.02169*, 2018.

[11] Ehsan Hosseini-Asl, Yingbo Zhou, Caiming Xiong, and Richard Socher, "A multi-discriminator cyclegan for unsupervised non-parallel speech domain adaptation," *arXiv preprint arXiv:1804.00522*, 2018.

[12] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley, "Least squares generative adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2813–2821.

[13] Yaniv Taigman, Adam Polyak, and Lior Wolf, "Unsupervised cross-domain image generation," *arXiv preprint arXiv:1611.02200*, 2016.

[14] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[15] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[16] Niko Brümmer and Edward De Villiers, "The speaker partitioning problem.," in *Odyssey*, 2010, p. 34.

[17] Niko Brümmer and Albert Strasheim, "Agnitios speaker recognition system for evalita 2009," in *The 11th Conference of the Italian Association for Artificial Intelligence*. Citeseer, 2009.