Similarity Metric Based on Siamese Neural Networks for Voice Casting

Adrien Gresse, Mathias Quillot, Richard Dufour, Vincent Labatut, Jean-François Bonastre

▶ **To cite this version:**

Adrien Gresse, Mathias Quillot, Richard Dufour, Vincent Labatut, Jean-François Bonastre. Similarity Metric Based on Siamese Neural Networks for Voice Casting. International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 2019, Brighton, United Kingdom. pp.6585-6589, 10.1109/ICASSP.2019.8683178 . hal-02004762

HAL Id: hal-02004762
https://hal.science/hal-02004762

Submitted on 2 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SIMILARITY METRIC BASED ON SIAMESE NEURAL NETWORKS FOR VOICE CASTING

Adrien Gresse, Mathias Quillot, Richard Dufour, Vincent Labatut, Jean-Francois Bonastre

LIA, University of Avignon, France

firstname.lastname@univ-avignon.fr

ABSTRACT

Dubbing contributes to a larger international distribution of multimedia documents. It aims to replace the original voice in a source language by a new one in a target language. For now, the target voice selection procedure, called voice casting, is manually performed by human experts. This selection is not exclusively based on acoustic similarity between the two voices. Actually, it is also supported by more subjective criteria such as the "color" of the voice, socio-cultural choices... The objective of this work is to model a voice similarity metric able to embed all the concerned voice characteristics, including the observers' receptive interests. In this paper, we propose a Siamese Neural Networks-based approach, measuring proximity between the original and dubbed voices. We propose an adapted *jackknifing* cross-validation method to evaluate our similarity model on unseen voices. The results show that we successfully capture information allowing two voices to be associated, with respect to the character's or role's abstract dimension.

Index Terms— Voice casting, Similarity metric, Siamese networks, i-vector

1. INTRODUCTION

The voice is an important medium in various multimedia documents, allowing for example an immersion of the spectators in cultural works (*e.g.* movies, video games). For international distribution, companies need to adapt the voices to reach the widest possible audience. Subtitling is the simpler and cheaper option, but not the most practical one for spectators: a large audience is more comfortable with hearing speech, in general in his mother tongue, rather than reading subtitles and at the same time hearing speech in another language. The localization process requires the original voices to be replaced by new voices in the target language. This is called *dubbing* or *revoicing*. It involves a voice selection process, which means choosing among several candidate voices in a target language with respect to the source voice. This selection is referred to as *voice casting* and it is currently carried out by human operators. The choice for the target language voice could be simply based on its acoustic resemblance with the original one, but it may relies on previous usages of the target actor's voice too. We suppose the existence of stereotypical voices that depends on cultural factors. Sociologists talk about "reception" to qualify long-term effects involved in the voice perception process. The point is, artistic directors want to find a voice that will induce a similar effect according to the target culture, more than a voice that just sounds like the original one. Voice casting process suffers from different problems. First, the human operator is exposed to his subjectivity. In fact, he makes his choice among the voices he is used to work with. Second, there is a huge amount of available voices and it is impossible to listen to all of them. As a consequence, voice selection remains essentially a subjective process which favors already known voices.

Given the difficulty of the voice casting process, automatic recommendations could help the operator in his quest for new voices. Thus, we propose a method to evaluate the similarity between two voices, beyond the simple acoustical level. The classic notion of voice similarity refers to a phonic comparison, what is usually sought in the automatic speaker recognition field. Speaker recognition systems [1, 2, 3] are able to measure efficiently the probability that two speech extracts were pronounced by the same person or not. However, the way humans perceive voices as "similar" is still an open question. There is a large amount of research work about perceptual voice similarity that derives from [4]. The author introduces the notion of *voice quality*, what we understand as auditory characteristics that color an individual voice. Several works explore the voice similarity level assessment that we can perceive among a group of voices [5, 6, 7, 8, 9, 10, 11]. They show the existence of correlations between particular acoustic characteristics (mainly on formants) and the way we figure out that two voices are perceptually similar. However there is no well-established method to automatically estimate this similarity.

The notion of perceptual similarity has been studied on professional acted voices in [12, 13, 14], presenting different perspectives, including a speaker recognition-based similarity metric and a paralinguistic-based classification, in addition to a subjective experiment. The main challenge here is tied with this notion of similarity, especially when comparing multilingual voices. For example, an acoustically similar voice could be inconsistent to a very different culture indeed. Given the operator's choices, we wish to learn this abstract notion and to capture the expected latent information into the voice (*e.g.* voice inflections, rhythm, timbre ...).

In order to delve into the role-specific dimension perceived throughout the voice, we focus on multilingual video games, as we expect more prototypical characters there than in other multimedia productions such as movies. We restrain this work to two languages, where English is the source and French is the target language. In [14], we proposed an i-vector/PLDA approach. In this work, we move to *Siamese Neural Networks* (SNN) since PLDA tends to focus on the speaker identity. Additionally, that method could be understood as a mapping from original to target language speakers, more than a vocal similarity estimation. Our intuition is that SNN and the usage of a pair-based learning, are more suitable to voice similarity estimation taken in its abstract assumption.

This paper is organized as follows. In Section 2, we give details on our approach. Methodology and experimental protocol are described in Section 3. Then we present our results in Section 4, and finally, we make a conclusion and lay down our perspectives for future work in Section 5.

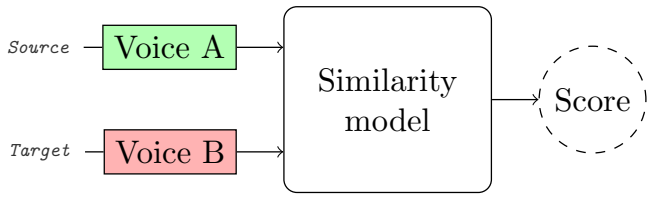

Fig. 1. *Overview of the automatic voice casting system.*

2. PROPOSED APPROACH

In Figure 1, we present a simple illustration of the automatic voice casting system. It accepts a pair of inputs corresponding to two voice excerpts. In addition, it gives a single output corresponding to the similarity between the two voices, beyond the plain acoustic resemblance. In other words, this score denotes the ability of the target voice to replace the original one. What makes the hearth of our system is our similarity model which is learnt from a set of voices in two different languages.

In Section 2.1, we present the concept and our motivation for the usage of a Siamese architecture, which constitutes the first novelty of this paper. The input data are represented with i-vectors, succinctly presented in Section 2.2.

2.1. Siamese Neural Networks (SNN)

Intuitively, this particular neural architecture provides a way to learn a similarity metric from pairwise relations between two inputs that share an abstract notion of similarity. The first work using SNN referred to [15] and takes place in the context of automatic signatures verification. This kind of architecture involves two identical neural networks. Thus, it takes two independent inputs that are projected into a latent space and finally merges them in a final layer by computing a penalty function which is named *contrastive loss*. Both networks' high-level representations are used for the computation of a distance used in the penalty function, and which is inversely proportional to the similarity between the pair of inputs. What makes the particular aspect of this architecture is that both networks share their parameters, allowing a comparison to be made. In this paper, we use an Euclidean distance.

In this work, we set up a Siamese architecture like in [16, 17, 18]. It promises two things:

- Due to the shared parameters, two strongly similar inputs cannot be mapped to very different places in the latent representational space. Conversely, a pair of different inputs cannot be close.
- The penalty function makes no distinction on inputs order contained in the processed pair (*i. e.* similarity function is symmetric).

In [16], the authors use an energy-based penalty function defined by the following equation:

$$E_W(I_1,\, I_2) = (\parallel G_W(I_1) - G_W(I_2) \parallel_2)^2 \tag{1}$$

Here, I_1 and I_2 refer to the inputs. G is a function that performs a mapping from the inputs space to a new space called the embeddings space. This function is parameterized by a matrix denoted W. By playing with W, the objective is to minimize the energy (Equation 1) when the two inputs are similar, but it also must verify that the distance denoted by E_W is still large enough for different inputs. Alternatively, in [19] the authors use a cosine similarity instead of the Euclidean distance. In our context we do not observe any difference with other distances, thus we do not use cosine in order to avoid negative values for convenience. We consider a binary variable denoted T, such as $T = 0$ when the inputs are similar and $T = 1$ otherwise. We also have a positive constant referred as m, that we can interpret as a margin such as $E_W(I_1, I_2) + m < E_W(I_1', I_2)$ with two similar inputs I_1, I_2 and where I_1' is a different input. The contrastive loss function is defined as follows:

$$\begin{aligned} L(I_1,\, I_2,\, T) = (1 - T) \times E_W(I_1,\, I_2) \\ + T \times max\{0,\, m - E_W(I_1,\, I_2)\} \end{aligned} \tag{2}$$

In this paper, we use Convolutional Neural Networks (CNN) ended by Fully Connected Layers (FCL) in order to compute the G function. The shared weights allow the twin networks to compute the same function G_W. Furthermore, the final layer units are combined with an hyperbolic tangent activation function, so output values belong to interval $[-1,\, 1]$. We give more details in Section 3.3.1 about the architecture.

2.2. I-vector based input representation

In general, the representational choice of the input data (*i. e.* audio segments) has a significant impact on the global system performance. In order to represent a length variable speech extract, we use here an i-vector based input representation.

I-vectors have been initially presented in the speaker recognition domain by [20] and are also used in other tasks (*e.g.* language identification, emotion recognition...). A large amount of data from many speakers over various sessions and channels are used to build the *total variability space*. Audio segments are projected onto this space and characterized by i-vectors [21]. They contain, the speaker individual characteristics. This representation can be extracted from sequences of different possible lengths. More generally, we see i-vectors as a compact representation of acoustic parameters sequences.

3. EXPERIMENTAL PROTOCOL

In this section, we first describe the data used for our voice casting problem (Section 3.1). Then, we detail the sequence extraction in (Section 3.2) and define our learning protocol in (Section 3.3). Finally, we propose a protocol in order to evaluate our approach that mostly focus on the similarity metric quality and on the relevance of the Siamese architecture (Sections 3.4 and 3.5).

3.1. Corpus

Our experimental data comes from a video-game called *Mass Effect 3*. The different characters may interact with their voice. The game was mostly dubbed in several languages, which means we have access to the original voice audio segments (here, in English) and their dubbed versions (only French in our experiments). More than that, we ensured that each segment from the original version has exactly one equivalent in the French version. That correspondence allows us to conduct a pair-based training with the pairs containing both English and French voices. Our goal consists in learning –such as the voice casting operator– the right way to automatically gather the original and dubbed voices.

A character is then defined by two different actors: one speaking in English and one in French. We ensured that all speakers (actors) are associated to a maximum of one character, in order to avoid any

bias. Moreover, we payed attention to characters' segments distribution, which is not uniformly shaped as it depends on the respective place of the character. We designed our experimental protocol in order to reduce this bias as much as possible. Both English and French sets contain $10,000$ voice segments that are high-quality studio recorded audio files. The corpus contains a total of 7.5 hours of speech in each language, and audio segments are 3.5 seconds long on average.

3.2. Sequences extraction

We transform the audio signal into 60-dimensional features vectors. It consists in 20 MFCC parameters including log energy, in addition to the 20 first and second order derivatives ($\Delta + \Delta\Delta$). We compute the parameters on a sliding Hamming window of 20 ms with a shift sets to 10 ms. We perform a cepstral mean normalization and we remove the low-energy frames that mainly correspond to silence. We train a language-independent i-vector system [22] for English on NIST SRE 2004, 2005 and 2006, and French on ESTER-1, ESTER-2, EPAC, ETAPE and REPERE. We train a Universal Background Model (UBM) of $2,048$ components from features vectors. Moreover, we train a total variability matrix T of low-rank 400, allowing us to extract i-vectors.

3.3. Pairwise learning

Voice segment pairs corresponding to a same character are denoted *target*, others are referred as *non-target*. We select 16 characters with at least 90 voice segments for both English and French in order to balance the number of *target* and *non-target* trials. Thus, we perform a 4-fold cross-validation, each fold is comprising 4 of these characters that we keep-out in order to test our hypothesis. We make sure we pull them out from the training set, so that any memorization effect is avoided. The four different train and test cases are denoted A, B, C and D.

Additionally, we presume language and linguistic content could be seen as possible bias. Since i-vectors are sensitive to duration which directly results from the linguistic content, we make sure to avoid pairing an English segment with its direct French equivalent by randomly shuffling the two sets. To ensure that the i-vector extraction is reliable we also reject all segments shorter than one second. Finally, we operate on same gender pairs only, in order to avoid a potential bias related to gender difference. Usually *target* pairs comply with this constraint.

Each fold contains a total of $32,400$ *target* pairs for test by combining 90 random segments in both English and French for each character. We randomly build the same number of *non-target* pairs according to previously stated constraints. This step aims to avoid the bias related to *a priori* probabilities of the 2 classes, by bringing them back to 0.5. We apply the same method to build the four training sets, except that we split the segments into 80% train and 20% validation sets. This step is fully randomized too and we operate on character's segments instead of pairs to ensure a balanced amount of voice segments between all characters in both languages. The total number of pairs available in a training case is $194,400$.

3.3.1. Training

We set up two Convolutional Neural Networks with Keras [23]. The specific network architecture is described in Table 1. We use a Xavier initialization [24] for weights and a normal initialization with $\mu = 0.5$ and $\sigma = 0.1$ for the bias. Also, we use the *Adadelta* optimizer with default configuration and dropout plus $L2$ regularization on each layer. Finally, we fix the mini-batch size to 128 trials. We trained all 4 models during 50 epochs and perform a validation accuracy based monitoring in order to avoid overfitting.

Input 400 x 1
10 Conv1D(pad-2, str-2)-32-BatchNorm-LeakyReLU MaxPooling + Dropout(0.25)
7 Conv1D(pad-2, str-2)-64-BatchNorm-LeakyReLU MaxPooling + Dropout(0.25)
4 Conv1D(pad-2, str-2)-256-BatchNorm-LeakyReLU MaxPooling + Dropout(0.25)
4 Conv1D(pad-2, str-2)-256-BatchNorm-LeakyReLU MaxPooling + Dropout(0.25)
Flatten Dense-2048-BatchNorm-LeakyReLU + Dropout(0.25)
Dense-256-BatchNorm-Tanh

Table 1. *Network architecture.*

3.4. Baseline NN configuration

We confront the Siamese Neural Networks to classic architectures. We use two different neural networks to compare with SNN. In the first one (called *2in-conc*), we concatenate the two inputs of dimension d into a single $2d$ input vector. In the second one (named *2in-merge*), we use an embedding layer for each d-dimensional input vectors, then we merge them into a single layer. We build these two networks according to Table 1, they just differ on their way of managing inputs, that is to say, concatenated inputs against multi-inputs model.

3.5. Evaluation

In order to evaluate the reliability of the learned similarity metric, we refer to the scores obtained on each pair. Here, we use the Euclidean distance computed in the embedding space as a similarity score. According to the nature of the two groups (*target* and *non-target*), we hope to observe a neat distinction between respective scores. Hence, it is relevant to perform a statistical hypothesis test: the t-test, also called *Student* test. The intuition is to compare the average scores of the two groups. It is a two-tailed test where the null hypothesis says the mean score of two classes are equals. This statistic is denoted t-score. In addition we also compute pure system performance evaluation metrics such as accuracy.

4. RESULTS & DISCUSSION

We present our results in Table 2. Considering the SNN architecture only, accuracy is slightly varying above 0.70% on the four development sets. As we expected, results on our different testing sets are not as good as in development sets. However, we come to a 0.62% accuracy score (on case C), A and B varying slightly below 0.6% while D does not perform better than random. This result shows us the model sensibility to the different characters. We note that t-score is higher on test than on development set. Since it corresponds to a ratio between *inter* and *intra* variance it means the difference between *target* and *non-target* is more important. We illustrate on Figure 3 the differences between *target* and *non-target* scores. According to *target*, we see a larger concentration on test which explains the

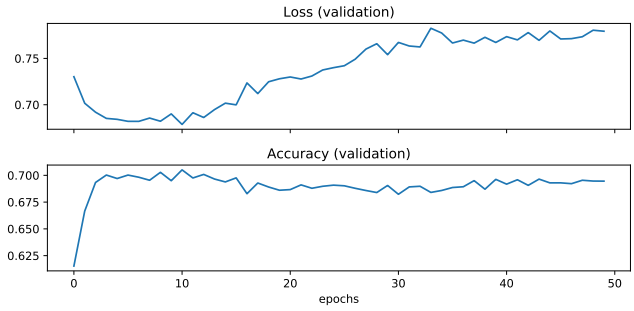

Fig. 2. *Averaging the 4 cases loss (top) and accuracy (bottom) on development set after each epoch on the SNN architecture.*

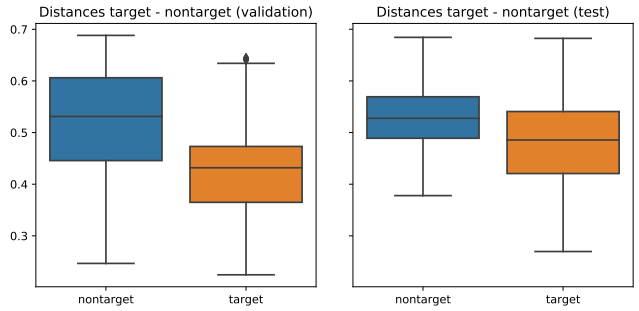

Fig. 3. *Target (blue) and non-target (orange) scores for case C for development (left) and test (right) with the SNN architecture.*

	2in-conc		2in-merge		**siamese-net**	
	acc.	*tscore*	*acc.*	*tscore*	*acc.*	*tscore*
A (test)	0.49	~~0.71~~	0.52	17.66	**0.55**	52.18
B (test)	0.49	5.34	0.50	4.53	**0.59**	77.99
C (test)	0.51	7.82	0.53	18.37	**0.62**	86.17
D (test)	**0.53**	17.30	0.52	14.50	0.50	~~1.87~~
A (dev)	0.94	185.72	0.93	169.93	0.72	47.90
B (dev)	0.96	211.32	0.94	190.68	0.71	52.77
C (dev)	0.93	161.16	0.93	160.16	0.70	45.18
D (dev)	0.96	227.85	0.96	212.80	0.71	44.46

Table 2. *Architectures comparison: concatenated inputs, merged inputs embedding and Siamese. Values refer to the accuracy (left) and t-score (right) obtained on the four different cases. Ruled out values stand for a p-value above the rejecting threshold.*

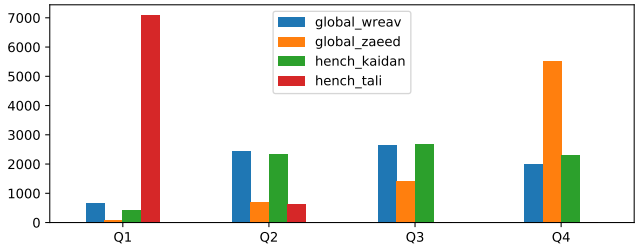

Fig. 4. *Characters (cases C) appearance count in the different quartiles of prediction error.*

higher t-score. Results show, we successfully learnt to make the distinction between *target* and *non-target* trials from new utterances on all cases, excluding D. By the way their respective associated probabilities are under the null hypothesis rejecting threshold. That is to say, there are few risks having the same mean between *target* and *non-target* scores. Figure 2 illustrates how fast the loss is evolving and the intrinsic quality of predictions are improving until convergence. After *10-th* epoch the model starts overfitting even with the use of regularization techniques such as dropout.

Given all the constraints we set up, there is no chance to have such an accuracy on test sets due to randomness. In these cases, the model found enough information to estimate a similarity between unseen characters/speakers, which is very interesting. As we illustrate in Figure 4, some characters (e.g. *hench_tali*) are easier to recognize than others (e.g. *global_zaeed*). We could hypothesize that female speakers may overemphasize on their voice in order to act like a soldier. That could explain why they get better results. On the contrary male voices may be more natural in the role of a soldier and consequently be harder to distinguish.

As expected the SNN gives better performance than the two other architectures for three out of the four cases. This is true only by considering the test. The single-network architectures obtain far better results on development set. We suppose single-networks (particularly when inputs are concatenated) are able to memorize the couple of speakers (source, target). In fact, the model seems to recognize the speakers over the development segments (which are unseen utterances from same training speakers). However, it dramatically fails on test when confronting to unseen speakers. On the contrary, the SNN seems to better generalize.

5. CONCLUSION & PERSPECTIVES

In this paper we hypothesize that we can benefit from pairwise relationship between different voices to learn a similarity metric for professional acted voices. Our experimental results show that it is indeed possible to use Siamese Neural Networks to model this abstract notion of similarity. In addition, we show that this architecture gives better generalization results on unseen voices, by comparison to classic architectures. The learnt metric is able to discriminate *target* and *non-target* pairs on new speakers/characters. It means that the built latent representational space highlights the "abstract" information that we were looking for.

Nevertheless, it is likely to depend on characters involved in both train and test pairs. Further works could investigate how the system reacts to particular pairs of characters, and try to take advantages from it in the learning process. Also, a larger dataset with increased variability could be very helpful in this way. Furthermore, we made usage of i-vectors in this very work, because it is a convenient representation which had shown its robustness many times. However, we think that we could benefit from a dedicated representation. In this approach, we detach the information representation process from the effective learned task by using the i-vectors space. Future work would adopt deeper neural architectures in order to jointly learn a task specific representation. Finally, we could consider the different classes of characters (*e.g.* soldier, alien, scientist...) in order to leverage the latent space on particular attributes of data. Attention based model or Conditional-GAN seem to be a promising lead for further work.

6. REFERENCES

[1] Finnian Kelly, Anil Alexander, Oscar Forth, Samuel Kent, Jonas Lindh, and Joel Åkesson, "Identifying perceptually similar voices with a speaker recognition system using auto-phonetic features.," in *INTERSPEECH*, 2016, pp. 1567–1568.

[2] Cuiling Zhang and Tiejun Tan, "Voice disguise and automatic speaker recognition," *Forensic science international*, vol. 175, no. 2, pp. 118–122, 2008.

[3] Jonas Lindh and Anders Eriksson, "Voice similarity-a comparison between judgements by human listeners and automatic voice comparison," in *Proceedings from FONETIK*, 2010, pp. 63–69.

[4] John Laver, "The phonetic description of voice quality," *Cambridge Studies in Linguistics London*, vol. 31, pp. 1–186, 1980.

[5] Kirsty McDougall, "Assessing perceived voice similarity using multidimensional scaling for the construction of voice parades.," *International Journal of Speech, Language & the Law*, vol. 20, no. 2, 2013.

[6] Phil Rose, "Differences and distinguishability in the acoustic characteristics of hello in voices of similar-sounding speakers," *Australian Review of Applied Linguistics*, vol. 22, no. 1, pp. 1–42, 1999.

[7] Deborah Loakes, *A forensic phonetic investigation into the speech patterns of identical and non-identical twins*, Ph.D. thesis, University of Melbourne, School of Languages, 2006.

[8] Francis Nolan, Peter French, Kirsty McDougall, Louisa Stevens, and Toby Hudson, "The role of voice quality settings in perceived voice similarity," *International Association for Forensic Phonetics and Acoustics, Vienna, Austria*, 2011.

[9] Oliver Baumann and Pascal Belin, "Perceptual scaling of voice identity: common dimensions for different vowels and speakers," *Psychological Research PRPF*, vol. 74, no. 1, pp. 110, 2010.

[10] Yusuke Ijima and Hideyuki Mizuno, "Similar speaker selection technique based on distance metric learning using highly correlated acoustic features with perceptual voice quality similarity," *IEICE TRANSACTIONS on Information and Systems*, vol. 98, no. 1, pp. 157–165, 2015.

[11] Eugenia San Segundo and Jose A Mompean, "A simplified vocal profile analysis protocol for the assessment of voice quality and speaker similarity," *Journal of Voice*, vol. 31, no. 5, pp. 644–e11, 2017.

[12] Nicolas Obin, Axel Roebel, and Grégoire Bachman, "On automatic voice casting for expressive speech: Speaker recognition vs. speech classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 950–954.

[13] Nicolas Obin and Axel Roebel, "Similarity search of acted voices for automatic voice casting," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1642–1651, 2016.

[14] Adrien Gresse, Mickael Rouvier, Richard Dufour, Vincent Labatut, and Jean-Francois Bonastre, "Acoustic pairing of original and dubbed voices in the context of video game localization," 2017.

[15] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in Neural Information Processing Systems*, 1994, pp. 737–744.

[16] Sumit Chopra, Raia Hadsell, and Yann LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 1, pp. 539–546.

[17] Raia Hadsell, Sumit Chopra, and Yann LeCun, "Dimensionality reduction by learning an invariant mapping," in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*. IEEE, 2006, vol. 2, pp. 1735–1742.

[18] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML Deep Learning Workshop*, 2015, vol. 2.

[19] Neil Zeghidour, Gabriel Synnaeve, Nicolas Usunier, and Emmanuel Dupoux, "Joint learning of speaker and phonetic similarities with siamese networks.," in *INTERSPEECH*, 2016, pp. 1295–1299.

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

[21] Achintya Kumar Sarkar, Jean-François Bonastre, and Driss Matrouf, "A study on the roles of total variability space and session variability modeling in speaker recognition," *International Journal of Speech Technology*, vol. 19, no. 1, pp. 111–120, 2016.

[22] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.

[23] François Chollet et al., "Keras," 2015.

[24] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.