INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Modeling Labial Coarticulation with Bidirectional Gated Recurrent Networks and Transfer Learning

Théo Biasutto--Lervat, Sara Dahmani, Slim Ouni

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
theo.biasutto-lervat@loria.fr, sara.dahmani@loria.fr, slim.ouni@loria.fr

Abstract

In this study, we investigate how to learn labial coarticulation to generate a sparse representation of the face from speech. To do so, we experiment a sequential deep learning model, bidirectional gated recurrent networks, which have reached nice result in addressing the articulatory inversion problem and so should be able to handle coarticulation effects. As acquiring audiovisual corpora is expensive and time-consuming, we designed our solution to counteract the lack of data. Firstly, we have used phonetic information (phoneme label and respective duration) as input to ensure speaker independence, and in second hand, we have experimented around pretraining strategies to reach acceptable performances. We demonstrate how a careful initialization of the last layers of the network can greatly ease the training and help to handle coarticulation effect. This initialization relies on dimensionality reduction strategies, allowing injecting knowledge of useful latent representation of the visual data into the network. We focused on two data-driven tools (PCA and autoencoder) and one hand-crafted latent space coming from animation community, blendshapes decomposition. We have trained and evaluated the model with a corpus consisting of 4 hours of French speech, and we have gotten an average RMSE close to $1.3mm$.

Index Terms: coarticulation modeling, visual speech synthesis, recurrent neural network, pretraining strategy

1. Introduction

Research in audiovisual speech intelligibility has shown the importance of the information provided by the face especially when audio is degraded [1, 2, 3]. Hence, a body of studies has worked on audiovisual speech synthesis, i.e. the ability to animate the mouth of a virtual character synchronously with acoustic speech (see for instance [4] for a comprehensive literature review).

Intelligible virtual avatar with accurate lip motion could have a great impact on second language acquisition, on assistance for the hard-of-hearing population, on video-games and film industries, or more generally for intuitive and natural human/machine interfaces. However, humans are very sensitive to any incoherence between audio and visual animation, such as an asynchrony between audio and visual channels [5] or small phonetic distortion [6, 7, 8]. The mismatch between audio and visual information can even lead to important effect on perception, as assessed by the McGurk effect [9].

While humans do not tolerate incongruity between audio and visual channels, generating these visual trajectories is incredibly complicated by the coarticulation effects. Coarticulation generally refers to the influence of the phonetic context on the realization of a phone [10]. This influence can be carryover due to the mass and inertia of each articulator (e.g. tongue, lips, jaw), or anticipatory due to an underlying high-level planning of speech production. Recurrent neural networks have been shown to properly deal with tongue and jaw coarticulation [11, 12, 13]. Thus, we assume they could also handle labial coarticulation present in visual data.

In a traditional two-step audiovisual synthesis, a text-to-speech system is used to generate both the acoustic signal and the phonetic sequences (the label and duration of each phoneme), then this phonetic information is used to generate the visual speech. This approach has the great advantages to benefit from the latest progress in acoustic speech synthesis, but may introduce some asynchrony between audio and visual channels.

In this paper, we address audiovisual speech synthesis with a new modular approach. We try to generate a sparse representation of the face from a phonetic sequence, and also benefit from the last progress in retargeting system [14, 15, 16, 17]. Nonetheless, audiovisual corpora are typically small for a deep learning approach. Even though one hour of speech was sufficient to learn articulatory coarticulation, the number of output point is far greater for our visual corpora (6 or 7 2D points for classic articulatory corpus like MNGU0 [18] or MOCHA-TIMIT [19], versus 44 3D points for our in-house corpus). Thus, we investigate pretraining strategy based on dimensionality reduction to ease the training and to improve the network's performance. We have compared two data-driven methods, principal component analysis and autoencoders, and a hand-crafted method, i.e., Blendshape's decomposition.

2. Data Acquisition

A motion-capture system based on eight Optitrack™ Flex-13 cameras have been used to record a professional female actress at $120fps$. Since conducting audiovisual acquisition and preprocessing them are greatly time-consuming, we carefully selected 2000 French sentences, about 4 hours of speech, that is

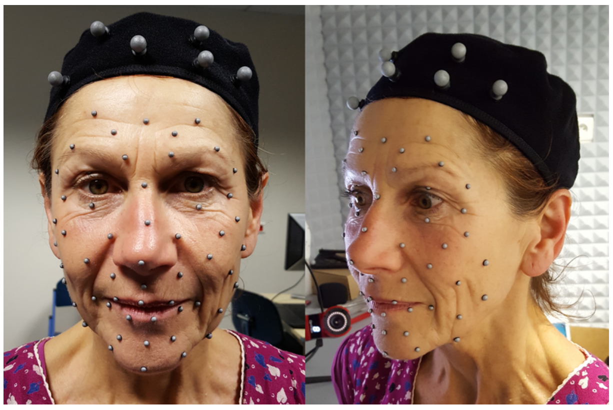
Figure 1: *Sensors location of our audiovisual corpus.*

linguistically rich. Then, sixty-three sensors have been glued on the actress face (fig. 1). In this work, we have used, however, only 44 sensors corresponding to the lower part of the face, and discarded information related to eyelids, forehead or eyebrows movements. Head movements have been removed using $9mm$ sensors glued on a beanie. Finally, forced alignment has been conducted using Kaldi and a DNN-based acoustic model trained on about 250 hours of French speech from TV and Radio (ESTER [20], EPAC [21] and ETAPE [22]).

3. Neural Architectures

3.1. Motivations

We have designed our neural networks according to three main considerations. First, we aim to design a speaker independent system. In speech recognition, this is generally achieved by collecting speech from a lot of different speakers. However, audiovisual corpora are expensive and time-consuming, preventing us from recording hundreds of speakers. Hence, we have explored the use of phonetic information (phoneme label and respective duration) which has already been successfully used in speech animation [23, 24] or articulatory inversion [13]. These high-level features properly represent speech while being totally speaker-independent. Secondly, there is absolutely no consensus on the maximum duration of anticipatory and retentive coarticulation effects. Although one can argue that its influence is limited in time, we would prefer avoiding the use of a temporal window to ensure the model take into account long-range coarticulation effect. Thus, we have used a well-known sequential model : the recurrent neural networks. Moreover, coarticulation should not exhibit very long-range correlation either, as RNNs usually do not perform very well. Finally, RNNs are effective at producing smooth trajectories, and they should alleviate the need for post-processing. This final filtering step is usually mandatory when using feed-forward neural networks for articulatory inversion or visual speech synthesis, as they tend to produce jagged trajectories.

3.2. Bidirectional Gated RNN

While feedforward neural networks are universal approximators (see for instance [25]), RNNs have been shown to be Turing complete, and thus should be able to approximate any dynamic system [26]. RNNs are able to summarize the input sequence into an internal state using cyclical connection, giving them the ability to learn temporal relationship and correlation between data points. However, these recurrent networks are limited to the use of *past* information, although knowledge of *future* information could improve the prediction. This is particularly true when dealing with speech production, for which it is well established that future phonemes influence on the production of the current phoneme and even the previous ones (anticipatory coarticulation effect). Bidirectional RNNs (BRNNs) [27] overcome this limitation using two layers simultaneously trained in positive and negative time direction.

Classic equations for BRNN is:

$$\begin{aligned}\overleftarrow{h}_t &= \overleftarrow{\mathcal{H}}(x_t, \overleftarrow{h}_{t+1}; \overleftarrow{\theta})\\ \overrightarrow{h}_t &= \overrightarrow{\mathcal{H}}(x_t, \overrightarrow{h}_{t-1}; \overrightarrow{\theta})\\ y_t &= W_{output}.[\overleftarrow{h}_t, \overrightarrow{h}_t] + b_{output}\end{aligned} \tag{1}$$

where h_t is the network internal state, x_t the network input and y_t the corresponding output at time t. W_{output} is the internal-

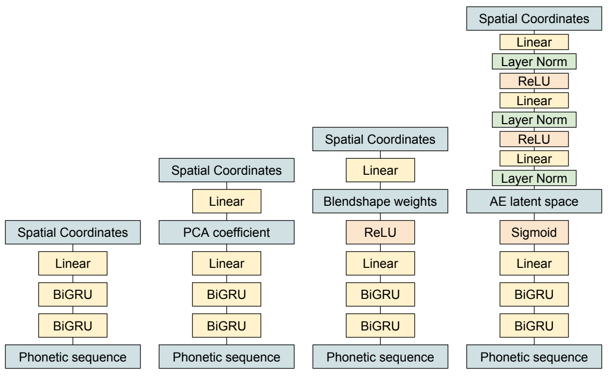

Figure 2: *Neural Architectures - From left to right: baseline model, injecting PCA eigenvectors, injecting blendshape knowledge and using autoencoder's decoder.*

to-output weight connection, and b_{output} the associated bias vector. $\mathcal{H}$ is the recurrent hidden layer transfer function parameterized by the θ set. $\overleftarrow{\cdot}$ denotes elements related to the backward layer and $\overrightarrow{\cdot}$ to the forward layer, finally $[\dots]$ denotes concatenation.

Despite its theoretical abilities, training vanilla RNN to learn long-range dependencies using a gradient descent algorithm is still a difficult task due to the *vanishing/exploding* gradient issue [28]. Long Short Term Memory (LSTM) network [29] gets around this issue by computing *increments* to the internal state and so encouraging information to stay for much longer, and by adding to each neural unit three gates which act as weight adjusters in function of inputs and hidden states. Among several variants of LSTM, the Gated Recurrent Unit (GRU) proposed by [30] has become quite popular and successful. GRU reduces the complexity of LSTM, by removing one gate and the cell memory and so decreasing the number of parameters, which should simplify the training.

3.3. Transfer learning

For a deep learning approach, 4 hours of speech is quite a small dataset, and may not be sufficient for a basic end-to-end training. To counteract this lack of data, we explore in this study the use of transfer learning. When working with neural networks, transfer learning generally refers to the pretraining of a network on related tasks, discard the last layer, and exploit the features learned for the main task. Here, we actually exploit the pretrained last layers, so one can argue our method is actually just a pretraining strategy and not really transfer learning.

We exploit dimensionality reduction tools able to compress the visual data into a latent space to initialize specific part of our networks. Features learned for this compression task are then used to learn a mapping between phonetic and articulation. The main idea is to force the network to use an previously computed latent representation of the sparse representation of the face, while still being trained from the raw spatial trajectories. To do so, we simply append the latent decoder at the top of the recurrent layers.

We compare in this paper three dimensionality reduction tools: blendshape decomposition, principal component analysis and autoencoder. Blendshape decomposition is a animation technique allowing a mesh to deform from a base shape to numerous pre-defined shape (keyshapes) through linear morphing. At any time, the targeted mesh b can be defined as a set of coef-

Table 1: *PCA, Blendshape and Autoencoder errors when compressing and reconstructing the test set.*

Method	PCA	Blendshape	Autoencoder
RMSE (mm)	0.33	0.95	**0.32**

ficients x_i, one for each keyshape b_i.

$$b = B.x + b_0 \tag{2}$$

where b_0 is a vector representing mesh coordinate of the base shape, B is a matrix containing all keyshapes vector $b_i - b_0$ and x is the keyshape weights vector. This equation clearly exhibits how easy it is to transfer this knowledge of useful keyshape into a neural network, by carefully initializing a dense layer with b_0 as bias and B as weight's matrix. The same trick can be applied for PCA, where a dense layer will be initialized with the computed eigenvectors as weight's matrix. The PCA approach has already been successfully used for audiovisual speech synthesis [31], and we carefully selected 13 blendshapes greatly inspired by the work of Benot and Govokhina [32, 33].

Extending equation 1 for our pretraining strategy gives

$$\begin{aligned}
\overleftarrow{h}_t &= \overleftarrow{\mathcal{H}}(x_t, \overleftarrow{h}_{t+1}; \overleftarrow{\theta}) \\
\overrightarrow{h}_t &= \overrightarrow{\mathcal{H}}(x_t, \overrightarrow{h}_{t-1}; \overrightarrow{\theta}) \\
z_t &= W_{output}.[\overleftarrow{h}_t, \overrightarrow{h}_t] + b_{output} \\
y_t &= f_{decoder}(z_t)
\end{aligned} \tag{3}$$

where z_t is the precomputed latent space and $f_{decoder}$ the associated decoding function. Note that our procedure is compatible with any dimensionality reduction tools, as long as $f_{decoder}$ exists and is differentiable.

4. Experiments

4.1. Methodology

We compare in this study four architectures, illustrated in figure 2. The first one is a simple baseline model trying to predict the sensor positions directly from two layers of bidirectional GRU. The three following architectures correspond to different latent space representations and their associated decoders, from left to right we respectively have PCA eigenvalues and eigenvectors, blendshape weights and keyshapes, and finally autoencoder latent space and its decoder. To ensure that the performance differences between baseline and the other models come from the initialization and not from the architecture differences, each of these three architectures has been evaluated twice, once using the pretraining procedure and once with a random initialization. We also pay attention to a fair comparison between latent spaces, so they all have a size of 13, which correspond to both the number of keyshapes and the number of principal components covering slightly more than 98% of the variance. As training is stochastic, each network has been independently initialized and trained 10 times. Each GRU layer has 128 neurons in each direction. The autoencoder are composed of two networks, an encoder with two hidden layers of 64 units with ReLU activation, and a symmetric decoder. The latent space is bounded between zero and one with sigmoid, and each layer is normalized using layer normalization [34]. The upper part of the right most figure in 2 illustrates the decoder architecture.

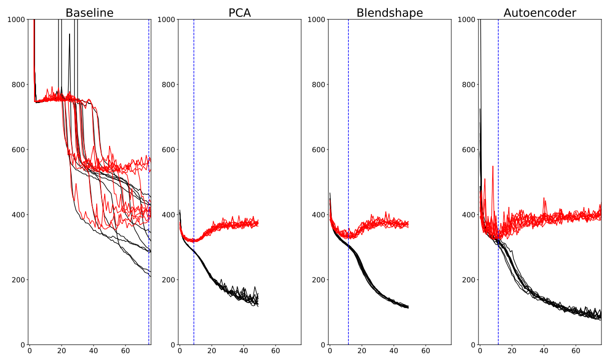

Figure 3: *Average loss per epoch - Black lines are the training losses and red lines are validation losses. Dashed lines represent average location of minimal validation loss.*

4.2. Training Procedure

We trained our networks in a framewise regression scheme where inputs and outputs are synchronized, a procedure successfully used for articulatory inversion [13]. The target output is a sequence of n-dimensional vectors representing the stacked spatial coordinates of each sensor, while the input is a sequence of one-hot vector representing the articulated phoneme at each time step.

As in typical regression task, we used the mean squared error as loss function, and defined the error as the Euclidean distance between prediction and target vector. The partial derivatives according to the BRNN parameters were computed with Backpropagation Through Time (BPTT) [35], and the network was fully unfolded for each training sequence. We used Adam [36] and its recommended parameters to finally update the network's parameters. Each GRU layer has 128 neurons in each direction, and we kept very small minibatches of size 2. The corpus split is a traditional 80/10/10, where 80% of the data go to the training set (about 200'), 10% to the validation set, and 10% to the testing set (about 20' each).

Baseline has been trained for 150 epoches, architectures initialized with PCA and Blendshape 50 epoches, and architectures initialized with Autoencoder 100 epoches.

4.3. Training Time

Figure 3 presents the average loss per epoch, for both the training set and the validation set. It clearly exhibits two major points, firstly the baseline method seems to encounter two local minimums or saddle points, located around a loss of 750 and 550. Resulting in uneven training time needed to reach minimal value on validation set. Then, all injections of precomputed latent space help the network to go through these training difficulties, and so greatly speed up the average number of epochs needed to reach the minimal value on the validation set. Autoencoder seems not to be as efficient as PCA and Blendshape's decomposition to speed up the training, but this difference may be explained by the depth of the architecture.

4.4. Performances Evaluation

Figure 4 presents average RMSE (in mm) and correlation for the four architectures and different initialization strategies.

A first insight that this figure provides is similar performances PCA, Blendshapes and Autoencoder architectures get

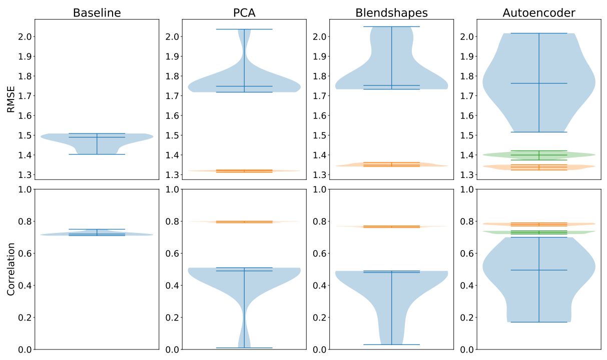

Figure 4: *Violin plot of performances per architecture - Blue corresponds to a random initialization without normalization, green to a random initialization with normalization and orange to an architecture with parameters injection. Upper plots present the RMSE in mm and lower plots the Pearson's correlation.*

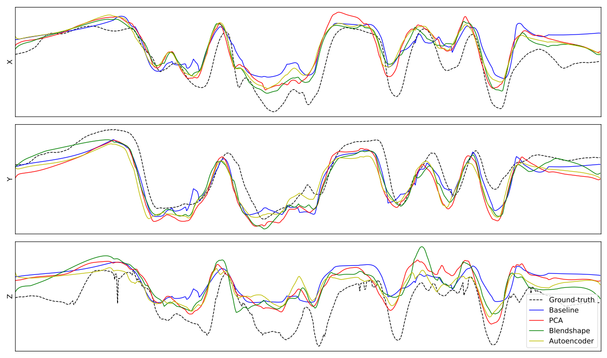

Figure 5: *Sample trajectories for the sensors located at the center of the upper lips.*

when randomly initialized, they actually are all stuck into the plateau presented in section 4.3. When looking at the generated trajectories of architectures stuck with a loss function around 750, with near 0 correlation and $2mm$ RMSE, it appears that they are all mainly stationary, so the network has only learned to reconstruct a face and completely fails to learn the labial coarticulation. This explains the poor correlation with relatively low RMSE. We suspect that these randomly initialized architectures could perform as well as the baseline with a longer training time, as this seems to be the case with Autoencoder.

A second insight is that all parameters injection actually helps the network to reach new levels of performance, even for hand-crafted features like keyshapes. Moreover, better dimensionality reduction tools (PCA and autoencoder in this paper, see table 1) seems to yield slightly to better overall performances.

Finally, even without parameters injection, the presence of layer normalization greatly helps the training. This clearly indicates that working on the training procedure could also improve overall performances. For example, further investigation of applying this normalization directly to RNN layers should be considered.

5. Conclusion

We have presented in this paper two main ideas to work around the small size of audiovisual database. First, we ensure that phonetic sequences, a high-level spearker-independent feature, is actually rich enough to properly infer the labial movements, while taking into account coarticulation, using bidirectional gated networks. Second, we demonstrate how a simple initialization strategy can greatly help such networks to learn coarticulation. Indeed, mapping a phonetic sequence to its corresponding labial articulation is not trivial. Without parameters injection, the network's loss function during training is quickly confronted to a plateau requiring a long training time.

The pretraining procedure consists of injecting precomputed latent space into the network, so the neural architecture already has an internal representation of the visual data to work with at the beginning of the training. From a certain point of view, we actually inject knowledge about the articulation into the network, as we provide a latent space able to represent the many different shapes of the lower part of the face during speech. This knowledge injection seems to greatly help the network to deal with labial coarticulation. Actually, this strategy improves both the training time and the performances of the network. The number of epochs needed to reach the best validation loss is divided by about seven (about 70 epoches for the baseline, about 10 for PCA, Blendshape and Autoencoder), there is up to a 35% relative RMSE improvements with the same training time between networks using our initialization and the same architecture with random initialization. Moreover there is far less variability in the network final performances. Surprisingly, performances using hand-crafted latent space (blendshape) are really close to performances using latent space computed with data-driven methods (PCA and Autoencoder).

In a similar way, it could be now really interesting to inject knowledge about the phonetic sequences. To do so, our future work will take inspiration from the NLP community by conducting language modeling at the phoneme level. These features at word-level have been shown to be really robust and useful for a lot of NLP tasks (e.g. ELMo [37] or BERT [38]), and we hope that the pre-trained contextualized features of phonemes could bring useful information to learn labial coarticulation. Moreover, such an approach could allow us to benefit from the great amount of data already available in classic speech datasets, as we only need phonetic sequences without visual data. Finally, these labial articulations are actually a sparse representation of the human face acquired with a motion-capture system, and so could benefit from any retargeting method to animate the face of an arbitrary 3D characters. In addition, we are planning to conduct a perceptual study to assess the perceptual quality of our coarticulation model.

6. Acknowledgements

Authors would like to thanks Valérian Girard and Vincent Colotte for their great contribution regarding to data acquisition and blendshape's decomposition. We also would like to thanks the anonymous reviewers for theirs relevant and useful comments.

This work was supported by PIA2 E-Fran, within the METAL project. It was also supported in part by Contrat de plan État / Région Lorraine - LCHN.

7. References

[1] W. Sumby and I. Pollack, "Visual contribution to speech intelligibility in noise," *Journal of the Acoustical Society of America*, vol. 26, p. 212, 1954.

[2] Le Goff, T. Guiard-marigny, M. Cohen, and C. Benoit, "Real-time analysis-synthesis and intelligibility of talking faces," in *In 2nd International conference on Speech Synthesis*, 1994, pp. 53–56.

[3] S. Ouni, M. M. Cohen, H. Ishak, and D. W. Massaro, "Visual contribution to speech perception: measuring the intelligibility of animated talking heads," *EURASIP J. Audio Speech Music Process.*, vol. 2007, no. 1, pp. 3–3, Jan. 2007.

[4] W. Mattheyses and W. Verhelst, "Audiovisual speech synthesis: An overview of the state-of-the-art," *Speech Communication*, vol. 66, pp. 182–217, 2015.

[5] N. F. Dixon and L. Spitz, "The detection of audiovisual desynchrony," *Perception*, vol. 9, pp. 719–721, 1980.

[6] K. P. Green and P. K. Kuhl, "The role of visual information in the processing of place and manner features in speech perception," *Perception and Psychophysics*, vol. 45, pp. 34–42, 1989.

[7] ——, "Integral processing of visual place and auditory voicing information during phonetic perception," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 17, pp. 278–288, 1991.

[8] J. Jiang, A. Alwan, P. Keating, E. Auer, and L. Bernstein, "On the relationship between face movements, tongue movements, and speech acoustics," *EURASIP Journal on Applied Signal Processing*, vol. 11, pp. 1174–1188, 2002.

[9] H. McGurk and J. MacDonald, "Hearing lips and seeing voices," *Nature*, vol. 264, pp. 746–748, 1976.

[10] W. J. Hardcastle and N. Hewlett, *Coarticulation: Theory, data and techniques*. Cambridge University Press, 2006.

[11] P. Liu, Q. Yu, Z. Wu, S. Kang, H. Meng, and L. Cai, "A deep recurrent approach for acoustic-to-articulatory inversion," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4450–4454.

[12] P. Zhu, L. Xie, and Y. Chen, "Articulatory movement prediction using deep bidirectional long short-term memory based recurrent neural networks and word/phone embeddings," in *Proc. of Interspeech*. ISCA, 2015, pp. 2192–2196.

[13] T. Biasutto-Lervat and S. Ouni, "Phoneme-to-Articulatory Mapping Using Bidirectional Gated RNN," in *Proc. of Interspeech*. ISCA, Sep. 2018, pp. 3112–3116.

[14] C. Curio, M. Breidt, M. Kleiner, Q. C. Vuong, M. A. Giese, and H. H. Bülthoff, "Semantic 3d motion retargeting for facial animation," in *Proc. of the 3rd symposium on Applied perception in graphics and visualization*. ACM, 2006, pp. 77–84.

[15] Y. Seol, J. Seo, P. H. Kim, J. Lewis, and J. Noh, "Artist friendly facial animation retargeting," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 6. ACM, 2011, p. 162.

[16] E. Chuang and C. Bregler, "Performance driven facial animation using blendshape interpolation," *Computer Science Technical Report, Stanford University*, vol. 2, no. 2, p. 3, 2002.

[17] L. Dutreve, A. Meyer, and S. Bouakaz, "Feature points based facial animation retargeting," in *Proc. of the ACM symposium on Virtual reality software and technology*. ACM, 2008, pp. 197–200.

[18] K. Richmond, P. Hoole, and S. King, "Announcing the electromagnetic articulography (day 1) subset of the mngu0 articulatory corpus," in *Proc. of Interspeech*. ISCA, 2011.

[19] A. A. Wrench, "A multichannel articulatory database and its application for automatic speech recognition," in *In Proceedings 5 th Seminar of Speech Production*. Citeseer, 2000.

[20] S. Galliano, E. Geoffrois, G. Gravier, J.-F. Bonastre, D. Mostefa, and K. Choukri, "Corpus description of the ester evaluation campaign for the rich transcription of french broadcast news." in *LREC*, 2006, pp. 139–142.

[21] Y. Esteve, T. Bazillon, J.-Y. Antoine, F. Béchet, and J. Farinas, "The epac corpus: Manual and automatic annotations of conversational speech in french broadcast news." in *LREC*. Citeseer, 2010.

[22] G. Gravier, G. Adda, N. Paulson, M. Carré, A. Giraudel, and O. Galibert, "The etape corpus for the evaluation of speech-based tv content processing in the french language," in *LREC-Eighth international conference on Language Resources and Evaluation*, 2012, p. na.

[23] S. Taylor, T. Kim, Y. Yue, M. Mahler, J. Krahe, A. G. Rodriguez, J. Hodgins, and I. Matthews, "A deep learning approach for generalized speech animation," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 93:1–93:11, Jul. 2017.

[24] B. Fan, L. Xie, S. Yang, L. Wang, and F. K. Soong, "A deep bidirectional LSTM approach for video-realistic talking head," *Multimedia Tools and Applications*, vol. 75, no. 9, pp. 5287–5309, May 2016.

[25] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks Are Universal Approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Jul. 1989.

[26] H. Siegelmann and E. Sontag, "On the Computational Power of Neural Nets," *J. Comput. Syst. Sci.*, vol. 50, no. 1, pp. 132–150, Feb. 1995.

[27] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[28] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[29] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[30] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," in *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Oct. 2014.

[31] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen, "Audio-driven Facial Animation by Joint End-to-end Learning of Pose and Emotion," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 94:1–94:12, Jul. 2017.

[32] C. Benoit, T. Lallouache, T. Mohamadi, and C. Abry, "A set of french visemes for visual speech synthesis," pp. 485–501, 1992.

[33] O. Govokhina, "Modèles de trajectoires pour lanimation de visages parlants," Ph.D. dissertation, Thèse de lInstitut National Polytechnique de Grenoble, 2008.

[34] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[35] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[36] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. of the 3rd International Conference for Learning Representations (ICLR)*, 2015.

[37] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. of NAACL*, 2018.

[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.