INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Phoneme-to-audio alignment with recurrent neural networks for speaking and singing voice

Yann Teytaut, Axel Roebel

STMS Lab, UMR 9912 (IRCAM, CNRS, Sorbonne University), Paris, France
yann.teytaut@ircam.fr, axel.roebel@ircam.fr

Abstract

Phoneme-to-audio alignment is the task of synchronizing voice recordings and their related phonetic transcripts. In this work, we introduce a new system to forced phonetic alignment with Recurrent Neural Networks (RNN). With the Connectionist Temporal Classification (CTC) loss as training objective, and an additional reconstruction cost, we learn to infer relevant per-frame phoneme probabilities from which alignment is derived. The core of the neural architecture is a context-aware attention mechanism between mel-spectrograms and side information. We investigate two contexts given by either phoneme sequences (model PHATT) or spectrograms themselves (model SPATT). Evaluations show that these models produce precise alignments for both speaking and singing voice. Best results are obtained with the model PHATT, which outperforms baseline reference with an average imprecision of 16.3ms and 29.8ms on speech and singing, respectively. The model SPATT also appears as an interesting alternative, capable of aligning longer audio files without requiring phoneme sequences on small audio segments.
Index Terms: phoneme-to-audio alignment, recurrent neural network, Connectionist Temporal Classification, voice analysis.

1. Introduction and related work

The general purpose of any alignment system is to determine a precise mapping between several representations that share common underlying information. For instance, while audio-to-score alignment aims to retrieve the start and end times of each event reported in a music score for a given performance [1], lyrics-to-audio alignment is focused on the time location of words pronounced in a recording [2]. In the analysis of speech and singing voice signals, for which linguistics, phonetics and pronunciation are important subjects of interest [3], it is also particularly convenient to have an alignment at the phoneme level, a task known as phonemes-to-audio alignment [4].

Given an audio file with voice, and a phonetic transcript of the text contained in it, a *forced alignment* system is expected to determine time boundaries for each phoneme automatically. Until recently, although some alternatives have been proposed [5], most forced aligners in the literature relied on Hidden Markov Models - Gaussian Mixture Model (HMM-GMM) to infer hidden states from likelihood scores derived from features computed on raw audio or spectral representations [3, 4, 6, 7, 8].

With the advent of deep learning in many fields related to music and audio data, numerous approaches for achieving forced alignment with Deep Neural Networks (DNN) have been reported as a complement to the standard methods. This recent trend comes with the design of a training procedure allowing neural models to learn how to predict phonetic sequences from audio inputs. The predictions can take the form of a phoneme classification or phonetic probability distribution per time unit, which are exploitable towards final alignment retrieval.

For such models to be trained and produce high-quality alignments, a first intuitive strategy is to compare, at each time, the phoneme predicted by the model to the real phoneme thanks to the Categorical Cross-Entropy loss function. From this point of view, the alignment challenge is a classification problem. Though great performances are to be denoted [9, 10], this calls for training data for which precise alignment is already known in advance. It is a major limitation since only few accessible datasets provide perfectly annotated ground truths.

A second approach makes use of specialized cost function, the Connectist Temporal Classification (CTC) [11]. CTC has been mostly used to train Recurrent Neural Networks (RNN) for sequence transcription [12]. A first attempt to apply this algorithm to alignment based on a Wav-U-Net revealed itself successful [13]. However, such a model requires large amount of data, much more than publicly available datasets provide.

A recent development does not face these shortcomings, and directly derives alignment from attention weights of an encoder-decoder that jointly aligns and separates speech [14]. It achieves state-of-the-art alignment precision, and is competitive with a reference Kaldi implementation of a Montreal Forced Alignment (MFA) algorithm [7], making it a strong baseline. While [14] was exclusively limited to speech, we believe that a comparison with singing voice is of interest: sung phonemes may be more challenging to align than spoken ones due to the larger diversity encountered in musical contexts [15].

In this work, we aim to develop a forced phonetic aligner suitable for both speaking and singing voice. We propose to couple the context-aware attention mechanism used in [14], and defined in [16], with the CTC algorithm as training objective. We argue that an additional spectral reconstruction loss helps producing high-quality alignments. After having evaluated our model with two types of attention context, namely spectral and phonetic, our results highlight relevant alignment performances. In the case of phonetic-based attention, we outperform the baseline reference in terms of mean and median time errors (32.2% and 15.2% relative gain, respectively) for speech and singing.

Our main contributions are:

- Two methods, based on the same neural architecture, for exploiting either spectral or phonetic side information during training and inference for phonetic alignment;
- The integration of a spectral reconstruction cost with the CTC to ensure that alignment information is taken into account for the generation of the phonetic posteriogram.

This paper is structured as follows. In section 2, we present our phonetic aligner by covering the various concepts that are implicated in its elaboration. Next, section 3 gives an overview of the experiments to pursue with our newly designed system, thus leading to our results and further discussions in section 4. Finally, section 5 summarizes and concludes our investigations.

http://dx.doi.org/10.21437/Interspeech.2021-1676

2. Phonetic aligner proposal

This section introduces our deep learning-based phonemes-to-audio aligner. The developed model relies on the CTC loss, a context-oriented attention mechanism and temporal coherence reinforcement. Once trained, we use it for alignment retrieval. These notions are explained in upcoming paragraphs.

2.1. Sequence prediction with CTC

Connectionist Temporal Classification (CTC) is an approach to train neural networks for tasks involving sequence labeling and segmentation [11] such as handwriting transcription [17], speech recognition [18] or audio-based keyword detection [19]. Let T and M denote two integers. Given some alphabet $\mathcal{A}$, an input representation $\mathbf{x} = \{x(t)\}, t \in [0, \dots, T-1]$ and its associated transcript $\mathbf{y} = \{y(m)\}, m \in [0, \dots, M-1]$ with $y(m) \in \mathcal{A}$, the CTC algorithm will predict output sequences $\hat{\mathbf{y}} = \{\hat{y}(t)\}, t \in [0, \dots, T-1]$ with $\hat{y}(t) \in \mathcal{A} \cup \{\varepsilon\}$. The extra token ε is referred to as the *blank* label, and means that there is no character from $\mathcal{A}$ concretely specified at time t. It is expected that the output sequence reduces to the target one, *i.e.* $\mathcal{B}(\hat{\mathbf{y}}) = \mathbf{y}$, where $\mathcal{B}$ is an operator merging successive repeated labels then removing all blanks from $\hat{\mathbf{y}}$, *e.g.* $\mathcal{B}(\varepsilon aaa\varepsilon\varepsilon\varepsilon b\varepsilon\varepsilon) = ab$.

As there may exist many sequences $\hat{\mathbf{y}}$ that can reduce to the ground truth $\mathbf{y}$, a CTC-based neural network is trained to produce a *phonetic posteriogram* (per-frame probabilities) $\mathbb{P}(\hat{\mathbf{y}}_t|\mathbf{x})$ over the set $\mathcal{A} \cup \{\varepsilon\}$ while maximizing the CTC conditional probability, with respect to the model's learnable parameters Θ,

$$\mathbb{P}(\mathbf{y}|\mathbf{x};\Theta) = \sum_{\hat{\mathbf{y}},\mathcal{B}(\hat{\mathbf{y}})=\mathbf{y}} \prod_{t=0}^{T-1} \mathbb{P}(\hat{\mathbf{y}}_t|\mathbf{x};\Theta) \tag{1}$$

hence minimizing the negative-log-likelihood loss function

$$\mathcal{L}_{\text{CTC}}(\Theta) = -\log \mathbb{P}(\mathbf{y}|\mathbf{x};\Theta). \tag{2}$$

By nature, the CTC loss does not favour any alignment, and may therefore seem inappropriate for this purpose, yet conclusive results were obtained for lyrics-to-audio alignment [13, 20]. We aim to align at the phoneme level, *i.e.* with higher precision.

2.2. Encodings and attention mechanism

To train the acoustic model on audio data, magnitude log-scaled mel-spectrograms are computed as input features $\mathbf{x}$ and long-term spectral information is encoded by two recurrent layers, namely Bidirectional Long Short-Time Memory (Bi-LSTM). Prior to the encoding itself, convolutional blocks help extracting relevant features from the spectrogram – and notably creating a representation that is suitable for our task.

A second encoder sharing the same architecture is used to process some side information. We investigate two contexts for this other branch: (1) A model with spectral context, denoted SPATT, that re-uses the spectrogram as second input to build a self-spectral attention revealing the relevant spectral segments; (2) A model with phonetic context, denoted PHATT, using non-aligned phonetic phrases as second input, that aims to associate each phoneme with specific spectral regions. These phonetic sequences (to be aligned) are converted into activation matrices seen as succession of one-hot vectors over the alphabet $\mathcal{A}$.

Following the implementation of [14, 16], we put forward a context-oriented attention mechanism. Let $\mathcal{E}_S$ and $\mathcal{E}_C$ denote the spectrogram and side information encodings[1], respectively.

[1]The mechanism accepts any length for the side information since $\mathcal{E}_S$ and $\mathcal{E}_C$ are shaped (T, E) and (T or M, E) for E encoding units.

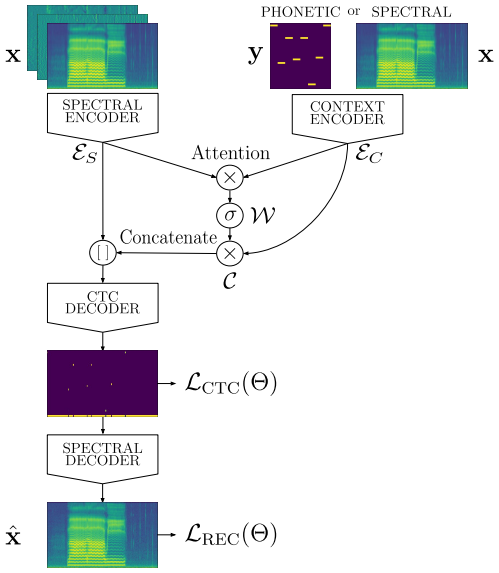

Figure 1: *Overview of our proposed system. Mel-spectrogram and side information (phonetic/spectral) are encoded, involved in an attention mechanism and then decoded, resulting in CTC posteriogram that is expected to allow spectral reconstruction.*

Attention weights $\mathcal{W}$ are computed thanks to a learnable dense layer $\boldsymbol{w}$, and are turned into a context vector $\mathcal{C}$ as below:

$$\mathcal{W} = \sigma\left(\mathcal{E}_S(\boldsymbol{w}\mathcal{E}_C)^\mathcal{T}\right) = \frac{\exp\left(\mathcal{E}_S(\boldsymbol{w}\mathcal{E}_C)^\mathcal{T}\right)}{\sum_{m=0}^{M-1}\exp\left(\mathcal{E}_S(\boldsymbol{w}\mathcal{E}_C)^\mathcal{T}\right)} \tag{3}$$

$$\mathcal{C} = \mathcal{W}\mathcal{E}_C. \tag{4}$$

All in all, the concatenation $[\mathcal{C}, \mathcal{E}_S]$ is sent to a CTC decoder, composed of two Bi-LSTM followed by a dense layer with `softmax` activation, leading to probabilities over $\mathcal{A} \cup \{\varepsilon\}$. This is summarized on **Fig. 1** presenting our system overview.

2.3. Temporal coherence reinforcement

Although a CTC-based network is trained to predict per-frame phoneme probabilities, it is worth noticing that the CTC loss is originally a transcription loss in the sense that it only makes sure that the sequence decoded from probabilities $\hat{\mathbf{y}}$ is close, if not equal, to the correct one $\mathbf{y}$. As a result, the time locations in the phonetic posteriogram do not contribute to the loss value whereas they should, as they directly have an impact on the alignment quality. To cope with this, and help the model predict phonemes at their accurate position, we add a supplementary constraint that consists in reconstructing the input spectrogram from the CTC predictions, seen as a compressed representation of input data. To this aim, a spectral decoder composed of two Bi-LSTM followed by a dense layer with `tanh` activation is used. See also on **Fig. 1**. The inputs to this decoder are the outputs of the dense layer prior to CTC `softmax` activation, without blank. We further argue that computing the spectral estimate $\hat{\mathbf{x}}$ and training the network to minimize the L2 loss

$$\mathcal{L}_{\text{REC}}(\Theta) = ||\hat{\mathbf{x}} - \mathbf{x}||_2 \tag{5}$$

will reinforce temporal coherence in the CTC posteriogram.

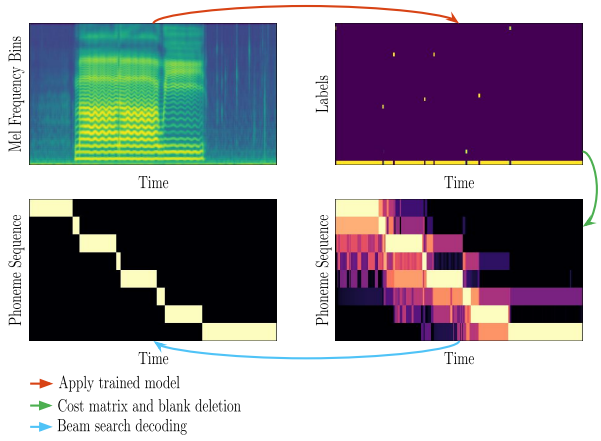

Figure 2: *Alignment procedure summary: the trained model outputs labels (phonemes+blank) probabilities that are used to create an accumulative cost matrix over time. Alignment can be retrieved with beam search decoding. Example is singing voice.*

2.4. Final alignment retrieval

Once the model is trained to predict relevant, well-localized probabilities, we can use it to retrieve the phonetic alignment. First, an accumulative score matrix, denoted α, is computed. To align the sequence $\mathbf{y}$ of length M along T frames, we define the padded sequence $\tilde{\mathbf{y}}$, which is required to deal with the blank label intrinsically linked to the CTC. It reads

$$\tilde{\mathbf{y}} = \{\varepsilon, y(0), \varepsilon, \dots, \varepsilon, y(m), \varepsilon, \dots, \varepsilon, y(M-1), \varepsilon\} \quad (6)$$

and constraints α to be $[2M+1, T]$-shaped. The element $\alpha[n,t]$ represents the score accumulated by the sub-sequence $\tilde{\mathbf{y}}_{0:n}$ at time t. The only allowed transitions are between two non-blank labels or between a blank and a non-blank label [21], so α can be computed efficiently with a dynamic programming technique resembling Viterbi's algorithm. The recursion rule is given by

$$\alpha[n,t] = \left(\sum_{p=0}^{2} \alpha[n-p, t-1] \left(1 - \delta^{p=2}_{\tilde{y}(n)=\varepsilon}\right)\right) \mathbb{P}[\tilde{y}(n), t] \quad (7)$$

where $\mathbb{P}[\tilde{y}(n), t]$ is the emission probability for the token $\tilde{y}(n)$ at time t and $\delta^{p=2}_{\tilde{y}(n)=\varepsilon}$ is the Kronecker delta returning here 1 when both $\tilde{y}(n) = \varepsilon$ and $p = 2$, and 0 otherwise.

The next step is to get rid of the blank labels to retrieve a $[M, T]$-shaped matrix α. To do so, a simple rule is applied to distribute all scores related to blanks ε between surrounding phonemes based on prior odds. For any k such that $\tilde{y}(k) = \varepsilon$, and for all time frame t, the blank cost $\alpha[k,t]$ is attributed to previous label cost $\alpha[k-1,t]$ if $\mathbb{P}[\tilde{y}(k-1),t] > \mathbb{P}[\tilde{y}(k+1),t]$, otherwise it is attributed to next label cost $\alpha[k+1,t]$. The kth blank can finally be removed from α after this subdivision.

Lastly, we retrieve the best path within our score matrix α thanks to beam search decoding [22]. We ensure to go through all of the sequence $\mathbf{y}$ in the right order. This gives us the target alignment. These various steps are illustrated on **Fig. 2**. Plus, we encourage reading [21] for more CTC algorithmic details.

3. Experiments

We compare our proposal to the state-of-the-art, attention-based alignment strategy for speech [14], which also provides data for the classical MFA algorithm from [7]. The experimental setup is exposed in this section.

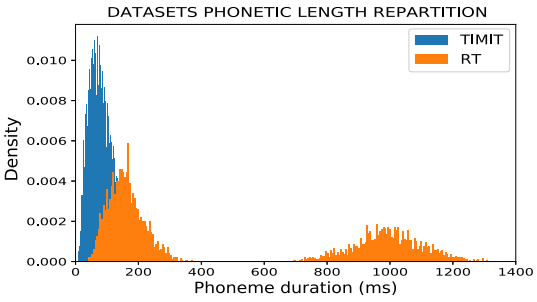

Figure 3: *Datasets analysis for phonetic duration in speech and singing. Silences are not counted. Pauses excluded, TIMIT and RT feature 187783 and 4768 phonetic utterances, respectively.*

3.1. Datasets

To study these aligners and reflect the diversity of voiced phonemes [3], we consider speech and singing datasets. For speaking voice, the 5-hour, multi-speaker TIMIT dataset is used [23]. For singing voice, we use the 1.5-hour French Chanter RT dataset whose construction is detailed in [24]. These are split in train, test, and validation sets[2]. Reference phonetic alignments are accessible for evaluations. The phonetic alphabet $\mathcal{A}$ size is 36 for RT (French) and 45 for TIMIT (English), pause included.

To quantify our temporal precision, we report on **Fig. 3** the repartition of phoneme duration. It shows that most (98%) of the spoken phonemes and half (47%) of the sung ones are shorter than 200ms. Phonemes in singing are longer and have higher variance than in speech. The RT singer was asked to sing slowly and hold long vowels to facilitate the development of a singing synthesis system [24], hence the bi-modal distribution.

3.2. Implementation details

3.2.1. Input pipeline

All tracks are converted to mono signals, resampled to 16kHz and cut into excerpts of at least 5s without truncating the last phoneme. To feed the neural network, log-scaled magnitude mel-spectrograms are derived from the Short Term Fourier Transform (STFT) by means of a 128-D mel-filterbank. Each of them is scaled to the normalized amplitude range $[-1, 1]$. For STFT computation, we use Hanning window with size 1024, FFT size 1024 and hop length 256, which corresponds to frames of 16ms (at 16kHz) as in the baseline [14]. We also concatenate its derivatives in time (delta $\Delta\mathbf{x}$ and delta-delta $\Delta\Delta\mathbf{x}$ features) along depth axis to let convolutional layers exploit temporal transitions more easily for phonetic boundaries estimation.

3.2.2. Architecture design

All Bi-LSTM layers have 512 units. To extract spectral features, we use convolutional blocks made up of a 2-D convolutional layer, preceded by batch normalization, a pooling operation that reduces only the feature axis by half, and a 25% dropout. Each block uses convolutions with 3×3 kernel size, `ReLU` activation, padding mode `same` and 16×2^b filters with $b \in \{0, 1\}$.

For the baseline, we have carefully followed instructions from original paper [14] for an reimplementation in Tensorflow. For the sake of fair comparison, and in opposition to their paper, we do not corrupt audio signals with music accompaniment to work only with clean data for training, validation and test.

[2]Splits (%) are 73.4/13.3/13.3 for TIMIT and 68.8/15.6/15.6 for RT.

Table 1: *Comparative quantitative results for alignment.* BSLN: *our baseline reimplementation;* V1 *and* MFA *are copied from [14];* OURS: *our proposal with phonetic* (PHATT)*, spectral* (SPATT) *or without* (NOATT) *attention. The metrics* MAE *and* MED *(see text) are in expressed in ms.*

	SPEAKING VOICE		SINGING VOICE	
BSLN	22.5	12.2	47.4	30.8
V1 from [14]	22.5	12.9	–	–
MFA from [14]	16.3	15.7	–	–
OURS − PHATT				
$\lambda = 1$	67.1	58.4	161.1	82.2
$\lambda = 1\text{e}^{-1}$	**16.3**	**11.8**	**29.8**	**19.5**
$\lambda = 1\text{e}^{-2}$	34.2	24.6	50.8	32.7
$\lambda = 1\text{e}^{-3}$	59.6	39.1	119.1	53.8
OURS − SPATT				
$\lambda = 1$	73.4	55.2	134.8	75.0
$\lambda = 1\text{e}^{-1}$	20.6	13.1	35.8	25.7
$\lambda = 1\text{e}^{-2}$	40.1	27.5	49.5	36.8
$\lambda = 1\text{e}^{-3}$	66.5	45.9	123.8	55.3
OURS − NOATT				
$\lambda = 1\text{e}^{-1}$	44.1	27.3	66.7	45.4
	MAE	MED	MAE	MED

3.3. Training procedure

We use batch size 16 and 500 epochs each composed of 128 training steps. Mel-spectrograms and phoneme sequences are padded with zeros. Training minimizes the loss function $\mathcal{L}(\Theta) = \mathcal{L}_{\text{CTC}}(\Theta) + \lambda\mathcal{L}_{\text{REC}}(\Theta)$ using the ADAM optimizer with learning rate 1e^{-4}. The impact of hyperparameter λ is discussed below. Training stops early after 101 consecutive epochs without validation loss improvement. The learning rate is halfed after 50 stagnant epochs. All codes are written in Python/Tensorflow 2.4 based on the CTC-Model from [25]. Training a model on a GeForce GTX 1080 Ti GPU takes 4hours.

4. Results and further works

4.1. Results

The performances of the models are evaluated with respect to the two main assessment metrics for alignment task [26], namely *Mean Average Error* (MAE), which is the average time imprecision in predictions; and *MEDian average error* (MED), which is the median time imprecision in predictions. They are expressed in milliseconds (ms). The quantitative evaluations for these metrics on chosen datasets can be found on Table 1.

As stated, the hyperparameter λ is of key importance: it acts as a trade-off between the two losses from Eq. (2) & (5). Should λ be too small (or zero), the CTC loss is not sufficiently constrained and alignment quality deteriorates. Should λ be too large, the reconstruction cost dominates the CTC loss in which case the model struggles to converge and degenerates into some simple spectral Auto-Encoder that does not perform alignment. It is worth noting that $\lambda = 0.1$ works best for all evaluations, including both databases and types of side information. This leads us to believe that this setting is not specific to contexts or data, and therefore is a good default value. These results prove the usefulness of the reconstruction loss coupled with the CTC loss for data alignment. For next discussions, we fix $\lambda = 0.1$.

4.2. Discussions

Evaluations in Table 1 show that our proposal allows significant performance improvement when compared to the baseline, and outperforms it for speech and singing for both MAE and MED. This means that the alignments produced by our system are more precise and less prone to cause severe outliers. This holds true for the two contexts of side information we have studied, spectral (model SPATT) and phonetic (model PHATT). The importance of the attention mechanism is noteworthy: trained without attention, the model NOATT in Table 1 performs poorly.

One can also note that our reimplementation of the baseline is coherent with the results from original paper, and highlights to some extent our competitiveness to the MFA for speech.

The best outcomes are obtained when training is enriched with phonetic transcripts (model PHATT). Significant gains in performances on MAE (30.6% on speech, 33.8% on singing) and MED (4.8% on speech, 25.5% on singing) are measured.

The model SPATT might not be as powerful as PHATT; but has a major benefit to mention. For long audio alignment (*e.g.,* entire songs), phonetic transcripts might be available as a whole, but not for small excerpts (5−10s). It would not be possible to rely on PHATT, at least not without a pre-segmentation, since short-term phonetic contexts would not be provided. For such cases, SPATT remains an interesting, robust alternative.

Another observation is that aligning singing voice is more challenging than speech. Independently of the aligner, MAE and MED drastically increase when considering sung phonemes instead of spoken ones. A reason is the much larger variance of phonetic length as exposed on **Fig. 3**. The network has to learn to recognize short and long time ranges for phonetic utterance, which is not the case for speech. It is also worth noticing that RT has been sung with a single pitch point.

4.3. Perspectives

The current study has dealt with clean data and relatively short audio excerpts. Further work will investigate alignment of singing voice in musical performances with DALI [27]. This requires to deal with the problems of background music and longer audio. Long audio (*e.g.,* audio books synchronization [28]) is particularly challenging [29, 30] because the memory consumption prevents from loading the complete audio and text so that a pre-segmentation of audio with text must be required.

5. Conclusion

In this paper, we have presented a new deep learning-based phoneme-to-audio alignment system designed to predict per-frame labels probabilities. To this aim, we have trained deep recurrent networks to learn from audio data and phonetic or spectral side information through a context-oriented attention mechanism. Not only did we take advantage of the CTC loss to ensure that output odds were consistent with the phonetic sequence to align, we also put forward a strategy to reinforce temporal coherence in CTC outputs. Based on a spectral reconstruction constraint, this additional cost guaranteed well-localized predictions in time. After evaluations, we have shown that our system was suitable for both speaking and singing voice. Our model PHATT exploiting phonetic transcripts has significantly outperformed the baseline reference in terms of mean and median errors, with an average progress of 23.7%. We have also proposed an alternative, fully-spectral aligner, model SPATT, usable when only global sequences are available.

6. Acknowledgement

This work has been funded by the French National Research Agency (ANR) project **ARS** (ANR-19-CE38-0001-01).

7. References

[1] A. Cont, "A coupled duration-focused architecture for real-time music-to-score alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence, Institute of Electrical and Electronics Engineers*, pp. 974–987, 2010.

[2] H. Fujihara and M. Goto, "Lyrics-to-audio alignment and its application," *Dagstuhl Follow-Ups*, vol. 3, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.

[3] Z. Duan, H. Fang, B. Li, K. C. Sim, and Y. Wang, "The nus sung and spoken lyrics corpus: A quantitative comparison of singing and speech," *IEE APSIPA*, pp. 1–9, 2013.

[4] J.-P. Hosom, "Speaker-independent phoneme alignment using transition-dependent states," *Speech Communication 51(4)*, pp. 352–368, 2009.

[5] F. Shield, "Maus goes iterative," in *Proceedings of the IV Internation conference on language resources end evaluation, Lisbon, Portugal*, 2004, pp. 1015–1018.

[6] K. Gorman, J. Howell, and M. Wagner, "Prosodylab-aligner: A tool for forced alignment of laboratory speech," *Canadian Acoustics 39.3*, pp. 192–193, 2011.

[7] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi," *Interspeech*, pp. 498–502, 2017.

[8] I. Rosenfelder, J. Fruehwald, K. Evanini, S. Seyfarth, K. Gorman, H. Prichard, and J. Yuan, "Fave (forced alignment and vowel extraction) 1.1.3," web resource, http://dx.doi.org/10.5281/zenodo.9846, 2017, accessed 22 September 2020.

[9] M. C. Kelley and T. B. V., "A comparison of input types to a deep neural network-based forced aligner," *Interspeech*, 2018.

[10] D. Backstrom, M. C. Kelley, and T. B. V., "Forced-alignment of the sung acoustic signal using deep neural nets," *Canadian Acoustics*, vol. 47 No. 3, 2019.

[11] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," *ACM/IEEE Supercomputing Conference (SC)*, vol. 148, pp. 369–376, 2014.

[12] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," *International Conference on Machine Learning (ICML)*, 2014.

[13] D. Stoller, S. Durand, and S. Ewert, "End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[14] K. Schulze-Forster, C. S. J. Doire, G. Richard, and R. Badeau, "Joint phoneme alignment and text-informed speech separation on highly corrupted speech," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[15] S. Aso, T. Saitou, M. Goto, K. Itoyama, T. Takahashi, K. Komatani, T. Ogata, and H. G. Okuno, "Speakbysinging: converting singing voices to speaking voices while retaining voice timbre," in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx-10)*, September 6–10, 2010.

[16] M.-T. Luong, H. Pham, and C. Manning, "Effective approaches to attention-based neural machine translation," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

[17] T. Bluche, H. Ney, J. Louradour, and C. Kermorvant, "Framewise and ctc training of neural networks for handwriting recognition," *13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 81–85, 2015.

[18] Y. Zhang, M. Pezeshki, B. P., S. Zhang, C. Laurent, Y. Bengio, and A. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," *Interspeech*, 2016.

[19] A. Vaglio, R. Hennequin, M. Moussallam, G. Richard, and F. d'Alché Buc, "Audio-based detection of explicit content in music," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[20] A. Vaglio, R. Hennequin, M. Moussallam, G. Richard, and F. d'Alché Buc, "Multilingual lyrics-to-audio alignment," *International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[21] A. Hannun, "Sequence modeling with ctc," *Distill*, 2017, https://distill.pub/2017/ctc.

[22] M. Freitag and Y. Al-Onaizan, "Beam search strategies for neural machine translation," *Proceedings of the First Workshop on Neural Machine Translation*, pp. 56—-60, 2017.

[23] V. Zue, S. Seneff, and J. Glass, "Speech database development at mit: Timit and beyond," *Speech communitation 9(4)*, pp. 351–356, 1990.

[24] L. Ardaillon, "Synthesis and expressive transformation of singing voice," Ph.D. dissertation, chapter 3, EDITE; UPMC - Paris VI Sorbonne Universités, 2017.

[25] Y. Soullard, C. Ruffino, and T. Paquet, "Ctcmodel: a keras model for connectionist temporal classification," research report, Université de Rouen Normandie, 2019, hal-02420358.

[26] A. Cont, D. Schwarz, N. Schnell, and C. Raphael, "Evaluation of real-time audio-to-score alignment," in *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, September 23–27, 2007.

[27] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, "Dali: a large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm," *IEEE International Society for Music Information Retrieval (ISMIR)*, 2018.

[28] X. Anguera, N. Perez, A. Urruela, and N. Oliver, "Automatic synchronization of electronic and audio books via tts alignment and silence filtering," *IEEE International Conference on Multimedia and Expo (ICME*, pp. 1–6, 2011.

[29] A. Katsamanis, M. Black, P. G. Georgiou, L. Goldstein, and S. Narayanan, "Sailalign: Robust long speech-text alignment," *Proceedings of Workshop on New Tools and Methods for Very-Large Scale Phonetics Research*, 2011.

[30] G. Bordel, M. Penagarikano, L. J. Rodríguez-Fuentes, A. Álvarez, and A. Varona, "Probabilistic kernels for improved text-to-speech alignment in long audio tracks," *IEEE Signal Processing Letters*, vol. 23, no. 1, p. 126–129, 2015.