MULTILINGUAL LYRICS-TO-AUDIO ALIGNMENT

Andrea Vaglio[12] **Romain Hennequin**[1] **Manuel Moussallam**[2]
Gaël Richard[2] **Florence d'Alché-Buc**[2]
[1] Deezer R&D
[2] LTCI, Télécom Paris, Institut Polytechnique de Paris
research@deezer.com

ABSTRACT

Lyrics-to-audio alignment methods have recently reported impressive results, opening the door to practical applications such as karaoke and within song navigation. However, most studies focus on a single language - usually English - for which annotated data are abundant. The question of their ability to generalize to other languages, especially in low (or even zero) training resource scenarios has been so far left unexplored. In this paper, we address the lyrics-to-audio alignment task in a generalized multilingual setup. More precisely, this investigation presents the first (to the best of our knowledge) attempt to create a language-independent lyrics-to-audio alignment system. Building on a *Recurrent Neural Network* (RNN) model trained with a *Connectionist Temporal Classification* (CTC) algorithm, we study the relevance of different intermediate representations, either character or phoneme, along with several strategies to design a training set. The evaluation is conducted on multiple languages with a varying amount of data available, from plenty to zero. Results show that learning from diverse data and using a universal phoneme set as an intermediate representation yield the best generalization performances.

1. INTRODUCTION

Lyrics-to-audio alignment aims at synchronizing lyrics text units such as paragraphs, lines or words to the timed position of their appearance in the audio signal. Tools dedicated to this task have many practical applications: they can be applied to generate new annotated data to train more robust singing voice recognizers [1]; or be used as building blocks in specific applications such as karaoke [2], navigation within songs [3] or explicit lyrics removal [4]. Lyrics alignment methods are typically inspired from text-to-speech methods. Although text-to-speech alignment is a mature [5] and widely studied task [6], lyrics-to-audio alignment remains a challenging problem with specific limitations. First, the musical accompaniment acts as a loud background "noise", potentially highly correlated with the signal of interest since vocalists usually sing in harmony and rhythm with instruments. A singing voice separation algorithm pre-processing step is often used to partially overcome this problem [7]. Second, singing voice exhibits more variety than speech with potentially large phonemes pronunciation variations between songs and extended tessitura. Recent studies have proposed efficient alignment methods for singing voice [8,9], but only for the English language, for which annotated data is abundant. The question of their ability to generalize to other languages, especially in low (or even zero) training resource scenarios, has not been properly addressed.

Arguably a monolingual evaluation is unrepresentative of the variety of music recordings available in large scale collections. Commercial streaming services commonly serve content in hundreds of languages and a non-negligible number of popular songs even have multilingual lyrics [10]. However, annotated data on this type of content are scarce. A source of inspiration comes from the related field of multilingual speech recognition [11]. Transfer learning methods [12] have been shown to improve performance on language with few to zero training data. However, this improvement on low-resource languages can sometimes be detrimental to performances on languages with more resources [11].

The goal of this paper is to evaluate and extend state of the art lyrics-to-audio alignment methods to a language-independent setup. First, we review the fitness of these systems to the multilingual framework. Then, we focus on one architecture and study two key features likely to allow generalization to several languages: 1) the intermediate representation space (character versus phoneme) and 2) the design of the training dataset. Evaluation is performed on multiple languages with various amounts of data available, from plenty to zero. The paper is organised as follows: related works are presented in Section 2. We then describe the proposed method in Section 3. The experimental setup and results are described respectively in Section 4 and Section 5. Finally, conclusions are drawn in Section 6 and future works are discussed.

2. RELATED WORKS

Singing voice alignment methods are typically inspired from text-to-speech alignment systems. Classically, an acoustic model is trained and used to force text to audio alignment using a Viterbi algorithm [5]. These models are usually trained using alignment annotations, at the frame level, between audio and text. However, the availability of such annotations is very limited for polyphonic music where they are traditionally generated by employing an intermediate model [1], leading to suboptimal performances [8]. More generally, the development of such approaches for singing voice was slowed down by the lack of publicly available annotated dataset at word or even line level. Some models were trained on speech and adapted to singing using speaker adaptation technique and a small singing dataset. For instance, in [13], a monophone *Hidden Markov Model* (HMM) is trained on speech and adapted on a small corpus of manually annotated *a cappella* songs with *Maximum Likelihood Linear Regression* (MLLR). The alignment is then performed on polyphonic songs after extracting the singing voice with a melody transcription and a sinusoidal modeling technique. Other models were trained with "low quality" automatic annotations generated with forced alignment using an *Automatic Speech Recognition* (ASR) system. In [1], a speech recognizer is used to generate a large amount of singing annotations by aligning a large corpus of *a cappella* singing to their corresponding lyrics. Annotations are then used to train a new acoustic model. This new model is used to align 19 vocal tracks from English language pop songs: the phoneme sequence is estimated for each track and its Levenshtein distance to the ground truth sequence from the lyrics is computed to find the alignment path. To help alignment, multiple approaches tried extending speech recognizers with external information such as chords [14], score [15] or note onsets [16].

The recent release of the DALI dataset [17] has led to significant progress in lyrics-to-audio alignment. This dataset is the first publicly annotated singing voice dataset available. It contains 5358 audio tracks with time-aligned lyrics at paragraph, line and word levels. These annotations are created from manual annotations and are considered to be very good. It is composed of varied western genres (*e.g.* rock, rap and electronic) in several languages. Novel singing voice separation algorithms displayed impressive results [18] and were also shown to improve significantly lyrics-to-audio alignment systems performances [7]. State-of-the-art approaches for lyrics alignment were compared in the MIREX 2019 challenge [1] . Two submitted systems showed particularly strong performances. The first one was SDE2, described in [8]. It is based on an end-to-end audio-to-character architecture, more precisely a wave-U-net. A preprocessing step of singing voice separation is performed, during training and inference, using a U-net convolutional network. The acoustic model is trained on a private English dataset of 40000 songs using a CTC algorithm. The second one was GYL1, described in [9], which obtained the best results on the challenge. It is based on a *Time Delay Neural Network* (TDNN) which is trained on the English subpart of the DALI dataset. It uses an extended lexicon to cope with long vowels duration in singing and genre labelling information (phoneme units are annotated with genres) but does not rely on a preprocessing step of singing voice separation.

Although it achieved the best performances in the MIREX challenge, GYL1 can not be straightforwardly used in a multilingual setup: it is composed of multiple parts, some of them, such as the pronunciation dictionary and the language model, being specific to English. To be able to use it on a new language, it would require to modify, or retrain, these parts. In comparison, SDE2 is based on an end-to-end acoustic model, trained with CTC algorithm, that directly outputs characters. It is more suitable to perform multilingual lyrics-to-audio alignment as it can be theoretically applied to any languages being based on the same script (writing system) as the training language.

Employing characters may not be optimal for multilingual lyrics-to-audio alignment: [8] suggest that using phoneme as an intermediate representation could be more relevant for aligning song in other languages. They argue that, for phoneme based systems, only the pronunciation dictionary has to be replaced for a new language, while a character based system is limited by the set of characters that the acoustic model outputs. For instance, SDE2 can only be used to align songs in Latin script languages. The output of the acoustic model could be extended with characters from scripts of new languages, as in [19], but it would require retraining the acoustic model each time a new script is added in the language pool. Using phoneme as an intermediate representation, any language can be theoretically aligned for any trained model if a pronunciation dictionary is available. In this work, we study a system inspired from [8] using either a character or a phoneme intermediate representation.

3. PROPOSED METHOD

A general overview of the proposed system is described in Figure 1. It is composed of three parts: a singing voice separation model, an acoustic model and a lyrics-to-audio alignment procedure. It takes as input a song x, its corresponding lyrics y and output the synchronized lyrics $\hat{y}$. Vocals are extracted from the song using a singing voice separation module. The acoustic model processes features extracted from the isolated vocal signal. The acoustic model consists in an RNN trained with a CTC algorithm. The set of outputs of the acoustic model is either characters of the Latin alphabet or phonemes of an universal phoneme set. Lyrics-to-audio alignment is performed on outputs of the acoustic model by a CTC-based alignment decoding function.

3.1 Acoustic model

The acoustic model of our system is a RNN trained with a CTC algorithm. CTC-based acoustic models were successfully used for multilingual speech recognition [19,20]. The RNN part is composed of bi-directional *Long Short-*

[1] `https://www.music-ir.org/mirex/wiki/2019:Automatic_Lyrics-to-Audio_Alignment`

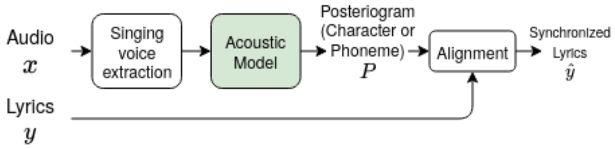

Figure 1. Overview of the lyrics-to-audio alignment system. Our study focuses on the training of the acoustic model (section 3.1) and the design of intermediate posteriogram representation spaces (section 3.2). The alignment block is described in section 3.3

Term Memory (LSTM) layers. Authors in [21] argue that such models can give reliable alignments given that outputs at each frame depend on the entire input sequence. In contrast, uni-directional CTC acoustic model suffers from alignment delay [22].

CTC makes it possible to directly train RNN models using weakly aligned annotations, e.g. at word or line level. To do that, the CTC algorithm introduces a new symbol called "blank" (noted ϵ) which represent a non-emission token. The total probability of the output label sequence is then marginalized over all possible alignment for a given input. In our case, the output label sequence is a sequence of character or phoneme. Since the objective function is differentiable, the network can be trained with standard back-propagation through time. The CTC algorithm is more extensively described in [23].

3.2 Character vs Phoneme

We consider two different intermediate representations for our architecture. The first one is a character set, here the Latin alphabet. This representation does not need any kind of expert linguistic knowledge as the acoustic model directly outputs characters probability. However, such a representation is not suitable to perform alignments of songs in a language with a different script. To process those, the acoustic model would need to be retrained with new data on the given script. Moreover, even for languages sharing the same script, a character-based representation is sub-optimal for transferring knowledge between languages, as characters pronunciation can significantly differ from one language to another. Our approach relies on the following remarks: all languages share some common phonemes and phonemes are considered to be language independent [24], i.e. to be pronounced the same way across languages. Therefore, using a universal phoneme set as an intermediate representation makes it possible to leverage similarity between sounds across languages. The idea is to use consistent phonemes across languages used for training and that most phonemes from an unseen language appear in the languages used for training.

It can be achieved using *international phonetic alphabet* (IPA) symbols. The IPA is a set of phonetic notations which is a standardized representation of sounds of all spoken language. IPA Pronunciations of words from all languages can be obtained using *Grapheme-To-Phoneme* (GTP) tools. Such tools are available for most common languages. The universal phoneme set is created by concatenating and merging the union of phoneme sets of all languages based on their IPA symbols.

3.3 Lyrics to audio alignment

In order to align a song to its corresponding lyrics y, the audio is sliced into segments of 5 seconds with a step size of 2.5 seconds. A posteriogram is generated by the trained acoustic model for each segment. To obtain the final posteriogram, all segments posteriograms are concatenated, cropped to half of their duration centered in their middle. We obtain a posteriogram $P \in [0,1]^{|C| \times T}$, C being the set of symbols supported by our acoustic model, either characters of phonemes, and T the number of temporal frames of the song. This matrix provides an estimation of the posterior probabilities of each symbol through time. Alignment annotations are then predicted, using the generated posteriogram P and lyrics y, with a CTC-based alignment function inspired from the CTC-based decoding function presented in [25] and is akin to a Viterbi forced alignment [26]. Viterbi forced alignment is a simpler version of Viterbi decoding where the possible paths are limited to the lyrics symbol sequence. To allow the use of ϵ during decoding, y is extended to z by adding a ϵ at the beginning, end, and between every unit. A decoding network of size $|z| \times T$ is then constructed from z. The goal of the decoding function is to find the path in the network that give the most probable alignment $\hat{y}$ of y given the posteriogram P. More precisely:

$$\hat{y} = \underset{B(\hat{y})=y}{\arg\max} \prod_{j=1}^{T} P(\hat{y}_j, j) \qquad (1)$$

where B is an operator that removes blanks and repetitions from a sequence $\hat{y}$. To do that, network's node $\alpha_{s,j}$ is defined as the probability of the best alignment of the subsequence $z_{1:s}$ after j frames. $\alpha_{s,j}$ scores can be calculated efficiently using a forward-backward algorithm, by merging together paths that reach the same node. $\alpha_{s,j}$ is then computed recursively from the values of α in the previous frame. Only transitions between blank and non-blank characters, and between pairs of distinct non-blank characters are allowed. ϵ at the beginning and end of the sequence being optional, there are two valid starting nodes and two final nodes. The coefficients α are initialized such as:

$$\alpha_{s,1} = P(z_s, 1) \text{ for } s \in \{1,2\} \text{ and } \alpha_{s,1} = 0, \forall s > 2 \quad (2)$$

Recursion is given by:

$$\begin{aligned} \alpha_{s,j} &= \max_{\tau \in \{0,1\}} (\alpha_{s-\tau,j-1}) P(z_s, j), \text{if } z_s \in \{\epsilon, z_{s-2}\} \\ \zeta_{s,j} &= \underset{\tau \in \{0,1\}}{\arg\max} (\alpha_{s-\tau,j-1}) \\ \alpha_{s,j} &= \max_{\tau \in \{0,1,2\}} (\alpha_{s-\tau,j-1}) P(z_s, j), \qquad \text{otherwise} \\ \zeta_{s,j} &= \underset{\tau \in \{0,1,2\}}{\arg\max} (\alpha_{s-\tau,j-1}) \end{aligned} \qquad (3)$$

Then, the probability of the best alignment is given by:

$$P(\hat{y}) = \max_{\tau \in \{0,1\}} (\alpha_{|z|-\tau,T}) \tag{4}$$

Alignment $\hat{y}$ can finally be computed with an inverse recursion. The initial unit is initialized such as:

$$\hat{y}_T = |z| - \arg\max_{\tau \in \{0,1\}} (\alpha_{|z|-\tau,T}) \tag{5}$$

Inverse recursion is given by:

$$\hat{y}_{j-1} = \hat{y}_j - \zeta_{\hat{y}_j,j} \tag{6}$$

Calculations are performed in log-space using the log-sum-exp trick [27] to avoid numerical instabilities. As some phonemes from target languages can be unseen in the training languages, the acoustic model will be unable to predict them, resulting in all alignment having a probability of zero. To get rid of this problem, a small amount of uniformly distributed noise is added to all entries of the posteriogram, as suggested in [8].

4. EXPERIMENTAL SETUP

4.1 Dataset

For this study, we consider several language subsets of the DALI dataset. They are described in Table 1. Experiments are conducted using 5 source languages for the initial multilingual system development. These source languages are: English, German, French, Spanish and Italian. English is considered as a high-resource language. The 4 others languages are considered as low-resource languages in this study. The split between train, validation and test datasets for the first five languages is an artist aware split [28]. We also consider 4 additional target zero-resource languages: Portuguese, Polish, Finnish and Dutch. Data from these languages are only used for evaluation. The split of the different language data, i.e. dali ids belonging to each dataset, are made publicly available at `https://github.com/deezer/MultilingualLyricsToAudioAlignment`. One dataset, that we named *5lang*, is created for multilingual training. The training and validation sets of this dataset are generated by simply concatenating respectively the training and validation sets of the 5 source languages. This dataset is largely unbalanced, English data dominating the corpus. Balancing the dataset with oversampling was tested without modification on performances of the multilingual model on low-resource and zero-resource languages. Similar results were also found for speech [29]. Worse, it significantly degrades results for the English language. These results were expected as the quantity of English data being far superior in comparison to other languages in the multilingual dataset, diminishing their importance could only degrade results for the multilingual model when tested on English dataset. Results of multilingual models trained with balanced dataset are displayed in supplementary materials.

Language	# Phonemes	Train (h)	Test (h)
English (en)	44 (5)	192.7	31.5
German (de)	44 (1)	17.4	2.3
French (fr)	42 (0)	8.9	0.9
Spanish (es)	35 (3)	8.4	1.1
Italian (it)	33 (0)	8.5	1.2
Portuguese (pt)	37 (0)	X	1.8
Polish (pl)	31 (2)	X	4.2
Finnish (fi)	25 (0)	X	3.1
Dutch (nl)	41 (2)	X	3.1

Table 1. Description of DALI language subset datasets and corresponding phoneme dictionary sizes. In parenthesis are displayed the number of phonemes only occurring in the given language and its equivalent ISO 639.1 code

The procedure to generate training samples and corresponding labels for the acoustic model is similar to the one described in [25]. To recall, Spleeter [18] is used to isolate vocals from each song. Training samples are then computed by segmenting extracted vocals. The character sequence associated with a segment is created from word level annotations of DALI by concatenating all words whose start position is within the segment. An instrumental token is generated if no words are present in the segment. For phoneme models, the phoneme sequence associated with a segment is generated from his corresponding character sequence using Phonemizer[2]. Phonemizer includes GPT tools for most common languages. It decomposes each word into a sequence of IPA symbol. To create the phoneme dictionary of one given language, we collect all IPA phonemes present in the corresponding dataset. For simplicity, we did not consider IPA symbols others than vowels and consonants. Sizes of dictionaries of phoneme of each language are given in Table 1. After concatenating and merging all dictionaries, we obtain a universal phoneme set of 62 phonemes. The language sharing factor [24] for the nine languages we used in this study is 5.35. It means that, on average, one unit of the universal phoneme set is shared by 5 to 6 languages of the language pool which supports the fact that IPA phonemes are rather consistent across languages that we consider in this study.

4.2 Parameters of acoustic models

We use the same architecture for all acoustic models. Several sets of regularisation and architecture's size parameters were tested without a clear impact on performances. Parameters of architecture are similar to those used in [25]. The model has 3 layers of bidirectional LSTM and a dense layer. It takes as input mel-scale log filterbanks coefficients and energy plus deltas and double-deltas. The acoustic model output is the probabilities of characters or IPA phonemes. In the first case, the set of outputs is the concatenation of the Latin alphabet, the apostrophe, the instrumental token, the space token and the CTC blank symbol

[2] `https://github.com/bootphon/phonemizer`

ϵ. A set of size 30 is obtained. In the second case, it is constituted of the universal phoneme set, plus the instrumental token, the space token and the CTC blank symbol ϵ. A set of size 65 is obtained. Parameters of training are the same as those used in [25].

4.3 Evaluation

To evaluate our system, we use the *Average Absolute Error* (AAE) [13]. For its calculation, the absolute difference between the actual start of the word timestamp and its estimation for each word is calculated. The final error score for a song is obtained by averaging over all word-level errors. A known issue of this metric is its perceptive dependence on tempo. In fact, one absolute error will not be perceived the same if the tempo is fast or slow. The *Percentage of correct onsets* (PCO) [14] was proposed to mitigate this effect. It is computed as the percentage of start of the word timestamps whose estimation are below a certain distance from the ground truth. This metric considers that errors bellow a certain threshold fall within human listeners perceptive tolerance. We use 0.3 seconds as the tolerance window. Both metrics are classic metrics of MIREX lyrics-to-audio alignment challenge. They are computed using the same evaluation script as the one used for the challenge [30] [3].

5. RESULTS AND DISCUSSION

5.1 State of the art comparison

To validate our implementation, We first compare our system with two state-of-the-art ones. Results are collected from the 2019 MIREX lyrics-to-audio alignment challenge. For this comparison, we use characters as intermediate representation space and only English for training. We use three standard evaluation datasets for lyrics-to-audio task. Hansen [31] and Mauch [14] are constituted of respectively 9 and 20 English pop music songs. Jamendo [8] is made of 20 English music songs of several western genres. All three datasets are annotated with start-of-word timestamps. Results are summarized in Table 2.

Our system performances are close to those of GYL1, with no significant differences for PCO metric on the three evaluation datasets. Although we use an architecture somewhat similar to SDE2 (i.e. a CTC based approach with a pre-step of singing voice separation), we report significantly better performances. It is worth noting that GYL1 and our system both use the English part of DALI as training dataset, while SDE2 uses a private dataset of unknown quality. We can postulate that the DALI dataset annotation quality is higher, which would explain the better performances reached by our implementation despite using a much smaller train set than SDE2.

Dataset	System	Mean AAE (s)	Mean PCO (%)
Hansen	SDE2 [8]	0.39 (0.12)	88 (3)
	GYL1 [9]	**0.10 (0.03)**	**97 (1)**
	Ours	0.18 (0.05)	95 (2)
Mauch	SDE2 [8]	0.26 (0.04)	87 (2)
	GYL1 [9]	**0.19 (0.03)**	**91 (2)**
	Ours	0.22 (0.03)	**91 (1)**
Jamendo	SDE2 [8]	0.38 (0.11)	87 (3)
	GYL1 [9]	**0.22 (0.06)**	**94 (2)**
	Ours	0.37 (0.05)	92 (2)

Table 2. Comparison between our character based architecture trained with the English part of DALI and state-of-the-art systems on standard lyrics-to-audio alignment evaluation datasets. Mean AAE is better if smaller, mean PCO is better if larger. Standard errors over tested songs are given in parenthesis

5.2 Multilingual generalization

Results of multilingual generalization experiments are displayed in Figure 2. Precise numerical values are reported in supplementary materials. Several conclusions can be drawn:

- Using a multilingual training set helps For both character and phoneme based architectures, the model exhibiting the best multilingual generalization is trained with multilingual dataset. In fact, this model significantly outperforms the ones trained on English on low-resource and zero-resource languages without degrading performances on English. With phoneme as intermediate representation, it even improves results on English. On low-resource languages, multilingual trained model obtains results on par with models trained only on the target language (e.g. French trained model on French dataset). It is worth noticing that the multilingual training dataset is only marginally larger than the English one. Performances differences are to be attributed to the additional information the model was able to extract from the diversity of languages seen during training.

- Use phonemes over characters as an intermediate representation has better performances Performances of phoneme based architectures are almost always better than those of their character based counterparts in all our experimental setups. The gap is bigger for models trained on the multilingual dataset than for those trained on monolingual ones. The only models that are not improved are the ones trained and tested on the same languages. Such results show that the use of phoneme as an intermediate representation enables transfer knowledge between language better than character representation.

- Training on multilingual data and a phoneme internal representation yields the best results in all considered cases Training the acoustic model on multilingual data and use a universal phoneme set is a relevant way for improving the generalization capacity of the considered lyrics-to-audio alignment architecture even to zero-

[3] `https://github.com/georgid/AlignmentEvaluation`

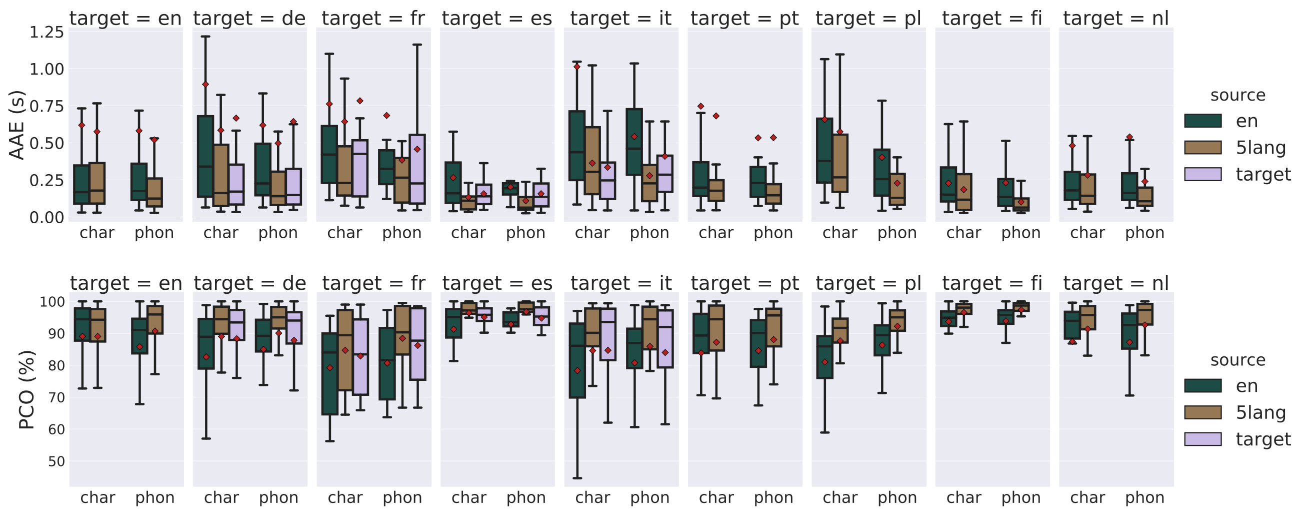

Figure 2. Lyrics-to-audio evaluation on DALI language subset datasets for phoneme and character based architectures. Several training set design strategies are considered. Languages are given by their ISO 639.1 code. Here "source" refers to data language used to train the given model and "target" refers to data language used to evaluate the trained model. When source is equal to target, architectures are trained and tested on the same language. Mean AAE is better if smaller, mean PCO is better if larger. Mean values are displayed using squares

resource scenarios.

6. CONCLUSION

In this paper, we investigated extending state-of-the-art methods in the multilingual context. Focusing on one architecture that seemed fit for generalization, we demonstrated that design choices regarding the training dataset and the acoustic representation space are salient factors. We have shown that using many languages to train the acoustic model and a universal phoneme set improves the multilingual generalization of such architecture. In this work, we have built a dataset using the language distribution found in DALI, which resulted in a largely unbalanced dataset. For comparison, we also conducted experiments with a balanced dataset, in which all 5 languages were equally present. The performance was similar, except for English, when it was significantly degraded. This raises the issue of how to design training sets in a setting where several high-resource languages are available. Although there are no publicly available datasets exhibiting such characteristics, future work should investigate this case. Existing works on multilingual speech processing [11] point towards increasing model complexity to circumvent this. Also, only a small set of languages were considered in this study. Additional experiments on a wider, more diverse set of songs remain to be conducted. Finally, future works should consider the specific case of songs with multilingual lyrics. This problem, known as code-switching, has been studied for speech [21] but never for music. Such a phenomenon is however not uncommon in popular music [10], thus it should be addressed too.

7. REFERENCES

[1] A. M. Kruspe, “Application of Automatic Speech Recognition Technologies to Singing Doctoral Thesis,” Ph.D. dissertation, University Fraunhofer, apr 2018. [Online]. Available: http://ieeexplore.ieee.org/document/7178348/

[2] A. Mesaros, “Singing Voice Recognition for Music Information Retrieval,” Ph.D. dissertation, Tampere university of technology, 2012. [Online]. Available: https://dspace.cc.tut.fi/dpub/handle/123456789/21404

[3] H. Fujihara, M. Goto, J. Ogata, and H. G. Okuno, “Lyric synchronizer: Automatic synchronization system between musical audio signals and lyrics,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1252–1261, 2011.

[4] A. Kruspe, “Automatic B**** Detection,” in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, no. September, 2016, pp. 3–4.

[5] C. W. Wightman and D. T. Talkin, “The aligner: Text-to-speech alignment using markov models,” in *Progress in speech synthesis*. Springer, 1997, pp. 313–323.

[6] A. Haubold and J. R. Kender, “Alignment of speech to highly imperfect text transcriptions,” in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2007, pp. 224–227.

[7] B. Sharma, C. Gupta, H. Li, and Y. Wang, “Automatic lyrics-to-audio alignment on polyphonic music using singing-adapted acoustic models,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 396–400.

[8] D. Stoller, S. Durand, and S. Ewert, "End-to-end Lyrics Alignment for Polyphonic Music Using an Audio-to-Character Recognition Model," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019. [Online]. Available: http://arxiv.org/abs/1902.06797

[9] C. Gupta, E. Yılmaz, and H. Li, "Automatic lyrics transcription in polyphonic music: Does background music help?" *arXiv preprint arXiv:1909.10200*, 2019.

[10] E. E. Davies and A. Bentahila, "Translation and code switching in the lyrics of bilingual popular songs," *The Translator*, vol. 14, no. 2, pp. 247–272, 2008.

[11] S. Watanabe, T. Hori, and J. Hershey, "Language independent end-to-end architecture for joint language and speech recognition," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2017.

[12] J. Cho, M. K. Baskar, R. Li, M. Wiesner, S. H. Mallidi, N. Yalta, M. Karafiat, S. Watanabe, and T. Hori, "Multilingual Sequence-to-Sequence Speech Recognition: Architecture, Transfer Learning, and Language Modeling," in *Proc. IEEE Spoken Language Technology Workshop (SLT)*, no. 1, 2019, pp. 521–527.

[13] A. Mesaros and T. Virtanen, "Automatic alignment of music audio and lyrics," in *Proc. Int. Conference on Digital Audio Effects (DAFx)*, 2008, pp. 1–4. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.212.6683{&}rep=rep1{&}type=pdf

[14] M. Mauch, H. Fujihara, and M. Goto, "Integrating Additional Chord Information Into HMM-Based Lyrics-to-Audio Alignment," in *Proc. IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, jan 2012, pp. 200–210. [Online]. Available: http://ieeexplore.ieee.org/document/5876304/

[15] G. Dzhambazov and X. Serra, "Modeling of phoneme durations for alignment between polyphonic audio and lyrics," in *Proc. of the 12th International Conference in Sound and Music Computing (SMC)*, 2015, pp. 281–286.

[16] G. Dzhambazov and A. Srinivasamurthy, "On the Use of Note Onsets for Improved Lyrics-To-Audio Alignment in Turkish Makam Music," in *Proc. 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 716–722.

[17] G. Meseguer-brocal and A. Cohen-hadria, "Dali : a Large Dataset of Synchronized Audio , Lyrics and Notes , Automatically Created Using Teacher-Student Machine Learning Paradigm," in *Proc. International Society on Music Information Retrieval Conference (ISMIR)*, 2018.

[18] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: A fast and state-of-the art music source separation tool with pre-trained models," in *Proc. Late-Breaking/Demo of International Society of Music Information Retrieval Conference (ISMIR)*, November 2019, deezer Research.

[19] S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, and K. Rao, "Multilingual speech recognition with a single end-to-end model," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4904–4908.

[20] M. Müller, S. Stüker, and A. Waibel, "Language adaptive multilingual ctc speech recognition," in *Proc. International Conference on Speech and Computer (SPECOM)*. Springer, 2017, pp. 473–482.

[21] K. Li, J. Li, G. Ye, R. Zhao, and Y. Gong, "Towards code-switching asr for end-to-end ctc models," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6076–6080.

[22] H. Sak, F. de Chaumont Quitry, T. Sainath, K. Rao *et al.*, "Acoustic modelling with cd-ctc-smbr lstm rnns," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 604–609.

[23] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ACM International Conference Proceeding Series*, vol. 148, 2006, pp. 369–376.

[24] T. Schultz and A. Waibel, "Language-independent and language-adaptive acoustic modeling for speech recognition," *Speech Communication*, vol. 35, no. 1-2, pp. 31–51, 2001.

[25] A. Vaglio, R. Hennequin, M. Moussallam, G. Richard, and F. d'Alché-Buc, "Audio-based detection of explicit content in music," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 526–530.

[26] D. G. Forney, "The viterbi algorithm," *Proc. of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[27] A. Hannun, "Sequence modeling with ctc," *Distill*, vol. 2, no. 11, p. e8, 2017.

[28] A. Flexer, "A closer look on artist filters for musical genre classification," in *Proc. of the 8th International Conference on Music Information Retrieval (ISMIR)*, no. 122, 2007, pp. 16–17.

[29] T. Alumäe, S. Tsakalidis, and R. Schwartz, "Improved multilingual training of stacked neural network acoustic models for low resource languages," in *Proc. Interspeech*, 09 2016, pp. 3883–3887.

[30] G. Dzhambazov, "Knowledge-Based Probabilistic Modeling For Tracking Lyrics In Music Audio Signals," Ph.D. dissertation, Universitat Pompeu Fabra Barcelona, 2017. [Online]. Available: http://www.tdx.cat/bitstream/handle/10803/404681/tgd.pdf?sequence=1{%}0Ahttp://mtg.upf.edu/node/3751

[31] J. K. Hansen, "Recognition of Phonemes in A-cappella Recordings using Temporal Patterns and Mel Frequency Cepstral Coefficients," in *Proc. 9th Sound and Music Computing Conference (SMC)*, 2012, pp. 494–499.