TISMIR

Meseguer-Brocal, G., et al. (2020). Creating DALI, a Large Dataset of Synchronized Audio, Lyrics, and Notes. *Transactions of the International Society for Music Information Retrieval,* 3(1), pp. 55–67. DOI: https://doi.org/10.5334/tismir.30

DATASET

Creating DALI, a Large Dataset of Synchronized Audio, Lyrics, and Notes

Gabriel Meseguer-Brocal*, Alice Cohen-Hadria* and Geoffroy Peeters†

The DALI dataset is a large dataset of time-aligned symbolic vocal melody notations (notes) and lyrics at four levels of granularity. DALI contains 5358 songs in its first version and 7756 for the second one. In this article, we present the dataset, explain the developed tools to work the data and detail the approach used to build it. Our method is motivated by active learning and the teacher-student paradigm. We establish a loop whereby dataset creation and model learning interact, benefiting each other. We progressively improve our model using the collected data. At the same time, we correct and enhance the collected data every time we update the model. This process creates an improved DALI dataset after each iteration. Finally, we outline the errors still present in the dataset and propose solutions to global issues. We believe that DALI can encourage other researchers to explore the interaction between model learning and dataset creation, rather than regarding them as independent tasks.

Keywords: singing voice; lyrics alignment; dataset creation; teacher-student paradigm; deep learning

1. Introduction

The singing voice is one of the most important elements in popular music. It combines two primary music dimensions: melody and lyrics. Together, they tell stories and convey emotions enriching our listening experience. The singing voice defines the central melody around which songs are composed and adds a linguistic dimension that complements the abstraction of musical instruments. For instance, the topics of lyrics are highly related to music genres, and various lyric sections define musical structures: verses reveal stories and choruses sum up the emotional message.

Despite its importance, the singing voice is a lesser-studied topic in the Music Information Retrieval (MIR) community. It was introduced as a standalone topic only a few years ago (Goto, 2014; Mesaros, 2013). The lack of large and good quality datasets is one of the main issues when working on singing voice related tasks. It prevents the MIR community from training state-of-the-art machine learning (ML) algorithms and comparing their results.

There are two main paths for creating datasets: either doing so manually or reusing/adapting existing resources. Although the former produces precise labels, it is time-consuming, and the resulting datasets are often rather small. Since existing resources with large data do not meet MIR requirements, datasets that reuse/adapt them are usually noisy and biased. Our goal is to build a large and public dataset with audio, lyrics, and notes aligned in time, by adapting existing resources and using ML models to refine them.

1.1 Our proposal

We propose the Dataset of Aligned Lyric Information, DALI, with synchronized audio, lyrics, and notes (Meseguer-Brocal et al., 2018). DALI aims to serve as a reference dataset for singing voice research. Our approach is motivated by Active Learning (Settles, 2008) and the Teacher-Student paradigm (Hinton et al., 2014). We establish a loop where we incrementally adapt and refine imperfect karaoke annotations while simultaneously improving the ML models used. During this constant interaction both annotations and models are improved.

1.2 Paper organization

First, we review previous work related to our problem in Section 2. Then, we describe and define the DALI dataset itself and how to use it in Section 3. In Section 4, we outline the methodology used to create it. Our system acquires lyrics and notes aligned in time from the Web (Section 4.1); finds a set of audio candidates (Section 4.2); and selects the correct one and adapts the annotations to it (Section 4.3). To this end, we propose a similarity measurement. This procedure highly depends on the precision of the Singing Voice Detection (SVD) method. In Section 5, we thus progressively improve the SVD quality using the teacher-student paradigm. This produces an enhanced DALI. We study the quality of the data and

* Ircam Lab, CNRS, Sorbonne Universite, Ministére de la Culture, Paris, FR

† LTCI, Institut Polytechnique de Paris, Paris, FR

Corresponding author: Gabriel Meseguer-Brocal (gabriel.meseguerbrocal@ircam.fr)

present strategies to correct global issues in Section 6. Finally, we discuss our research (Section 7).

2. Related Work

2.1 Singing Voice Datasets

The lack of large and good quality datasets is a crucial factor that prevents the development of singing voice tasks. There are no widely used reference datasets as in other disciplines, e.g. MNIST (Le Cun et al., 1998), CIFAR (Krizhevsky, 2009), or ImageNet (Deng et al., 2009) for computer vision. This problem is common to other MIR tasks. Fortunately, many solutions have been proposed in the last years (Benzi et al., 2016; Fonseca et al., 2017; Donahue et al., 2018; Nieto et al., 2019; Maia et al., 2019; Yesiler et al., 2019).

There are some datasets with audio and lyrics aligned in time for commercial purposes (LyricFind, Musixmatch or Music-story). Yet, they are private, not accessible outside host applications, come without audio, have only aligned lines, and do not contain symbolic vocal melody annotations (notes).

Currently, researchers working on singing voice tasks use small datasets, designed following different methodologies (Fujihara and Goto, 2012). Large datasets remain private (Humphrey et al., 2017; Stoller et al., 2019) or are vocal-only captures of amateur singers recorded on mobile phones that involve complex preprossessing (Smith, 2013; Kruspe, 2016; Gupta et al., 2018). An overview of datasets of public lyrics aligned in time can be found in **Table 1**. Most of these datasets are created in the context of lyrics alignment i.e. assigning start and end times to every textual element.

Lyrics are inevitably language-dependent. Researchers have created datasets for different languages: English (Kan et al., 2008; Iskandar et al., 2006; Gupta et al., 2018), Chinese (Wong et al., 2007; Dzhambazov, 2017), Turkish (Dzhambazov, 2017), German (Müller et al., 2007) and Japanese (Fujihara et al., 2011). In theory, we can adapt models obtained in one language to others, but it has not been proven.

Most of the datasets contain polyphonic popular music. There also some with A Capella music (Kruspe, 2016). However, it is always difficult to migrate from monophonic methods to polyphonic (Mesaros and Virtanen, 2010). More recently, authors proposed to use Automatic Speech Recognition (ASR) on imperfect transcripts to align lyrics and A Capella singing signals in a semi-supervised way (Gupta et al., 2018). The authors also propose a method for

Table 1: Lyric alignment datasets comparison.

Dataset	Number of songs	Language	Audio type	Granularity
(Iskandar et al., 2006)	No training, 3 test songs	English	Polyphonic	Syllables
(Wong et al., 2007)	14 songs divided into 70 segments of 20s length	Cantonese	Polyphonic	Words
(Müller et al., 2007)	100 songs	English	Polyphonic	Words
(Kan et al., 2008)	20 songs	English	Polyphonic	Sections, Lines
(Mesaros and Virtanen, 2010)	Training: 49 fragments ~25 seconds for adapting a phonetic model Testing: 17 songs	English	Training: a capella Testing: vocals after source separation	Lines
(Hansen, 2012)	9 pop music songs	English	Both accompanied and a capella	Words, lines
(Mauch et al., 2012)	20 pop music songs	English	Polyphonic	Words
DAMP dataset, (Smith, 2013)	34k amateur versions of 301 songs	English	A capella	Not time-aligned lyrics, only textual lyrics
DAMPB dataset, (Kruspe, 2016)	A subset of DAMP with 20 performances of 301 songs	English	A capella	Words, Phonemes
(Dzhambazov, 2017)	70 fragments of 20 seconds	Chinese Turkish	Polyphonic	Phonemes
(Lee and Scott, 2017)	20 pop music songs	English	Polyphonic	Words
(Gupta et al., 2018)	A subset of DAMP with 35662 segments of 10s length	English	A capella	Lines
Jamendoaligned, (Ramona et al., 2008) (Stoller et al., 2019)	20 Creative commons songs	English	Polyphonic	Words
DALI v1 (Meseguer-Brocal et al., 2018)	5358 songs in full duration	Many	Polyphonic	Notes, words lines and paragraphs
DALI v2	7756 songs in full duration	Many	Polyphonic	Notes, words, phonemes, lines and paragraphs

transferring A Capella alignment to the polyphonic case (Gupta et al., 2019). Datasets do not always contain the full duration audio track but often a shorter version (Gupta et al., 2018; Dzhambazov, 2017; Mesaros and Virtanen, 2010; Wong et al., 2007). If the tracks are complete, respective datasets are typically small.

2.2 The Teacher-Student Paradigm

This paradigm has two agents: the teacher and the student. We train the teacher on well-known labeled ground truth datasets (often manually annotated). The teacher labels some unlabeled data, used to train the student(s). Thus, students indirectly acquire the desired knowledge by mimicking the teacher's "behavior". This paradigm was originally introduced as a model compression technique to transfer knowledge from larger architectures to smaller ones (Bucilua et al., 2006). Small models (the students) are trained on a larger dataset labeled by large models (the teachers). A general formalization of this knowledge distillation trains the student on both the teacher's labels and the training data (Hinton et al., 2014). In the context of Deep Learning, teachers also remove layers of the student's architecture, automating the compression process (Ashok et al., 2017).

The teacher-student paradigm is also used as a solution to overcome the problem of insufficient labeled data, for instance for speech recognition (Watanabe et al., 2017; Wu and Lerch, 2017) or multilingual models (Cui et al., 2017). Since manual labeling is time-consuming, teachers automatically label unlabeled data on larger datasets used for training the students. This paradigm is relatively undeveloped in MIR. Wu and Lerch (2017) provide one of the few examples, applying it to automatic drum transcription, in which the teacher labels the student dataset of drum recordings.

All of these works report that the students improve the performance of the teachers. Therefore, this learning paradigm meets our requirements.

3. Dataset description

The DALI dataset is a collection of songs described as a sequence of time-aligned lyrics, each one linked to its audio in full duration. Annotations define direct relationships between audio and text information at different hierarchical levels. This is very useful for a wide variety of tasks such as lyrics alignment and transcription, melody extraction, or structural analysis.

Time-aligned lyrics are described at four levels of granularity: notes, words, lines and paragraphs. We describe lyrics as a sequence of characters for all levels. The word level also contains the lyrics as sequence of phonemes. Lyrics at the note level correspond to the syllable (or group of syllables) sung, and the frequency defines the musical notes for the vocal melody. The different granularity levels are vertically connected; i.e., one level is associated with its upper and lower levels. For instance, we know the words of a line, or which paragraph a line belongs to. **Figure 1** illustrates an example with two levels of granularity: a line and its notes.

Finally, each song has extra metadata information such as artist name, song title, genres, language, album cover, link to video clip, or release date.

3.1 The Annotations

In DALI, songs are defined as:

$$S = \{A_{notes}, A_{words}, A_{lines}, A_{paragraphs}\} \quad (1)$$

where each granularity level g with K_g elements is a sequence of aligned segments:

$$A_g = (a_{k,g})_{k=1}^{K_g} \text{ where } a_{k,g} = (t_k^0, t_k^1, f_k, l_k, i_k)_g \quad (2)$$

with t_k^0 and t_k^1 being a text segment's start and end times (in seconds) with $t_k^0 < t_k^1$, f_k a tuple (f_{min}, f_{max}) with the frequency range (in Hz) covered by all the notes in the segment (at the note level $f_{min} = f_{max}$, a vocal note), l_k the actual lyric's information and $i_k = j$ the index that links an annotation $a_{k,g}$ with its corresponding upper granularity level annotation $a_{j,g+1}$. The text segment's events for a song are ordered and non-overlapping – that is, $t_k^1 \leq t_{k+1}^0 \, \forall k$. Note how the annotations define a unique connection in time between the musical and textual domains.

In this article, we present the second version (v2) of the DALI dataset with 7756 songs, augmenting the first version (v1) with 5358 songs (Meseguer-Brocal et al., 2018) by 2398 songs. DALI has a total of 344.9 (v1) and 488.1 (v2) hours of music, and 176.9 (v1) and 247.2 (v2) hours with vocals. There are also more than 3.6 (v1) and 8.7 (v2) million segments $a_{k,g}$ and the average per song is 679 (v1) and 710 (v2). There are, on average, 2.36 (2.71)

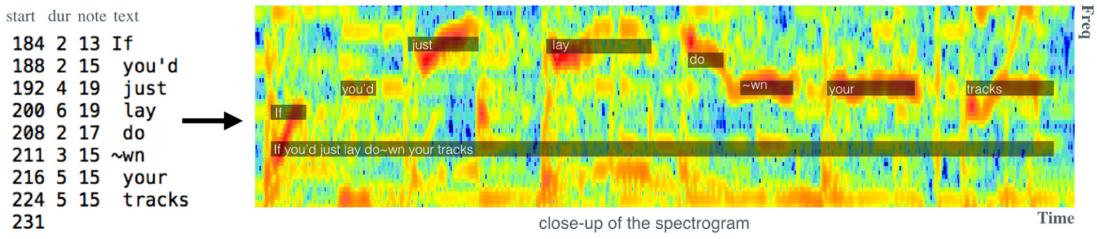

Figure 1: [Left] Our inputs are karaoke-user annotations presented as a tuple of `{time (start and duration), musical note, text}`. [Right] Our method automatically finds the corresponding full-audio track and globally aligns the vocal melody and the lyrics to it. The close-up of the spectrogram illustrates the alignment for a small excerpt at two levels of granularity: notes and lines.

songs per artist and 119s (115s) with vocals per song, in DALI v1 (v2), cf. **Table 3**. As seen in **Table 2**, DALI has many artists, genres, languages, and decades. Most of the songs are from popular genres like Pop or Rock, from the 2000s and English is the most predominant language.

3.2 Tools for Working with DALI

The richness of DALI renders the data complex. Hence, it would be difficult to use in a format such as `JSON` or `XML`. To overcome this limitation, we have developed a specific `Python` package with all the necessary tools to access the dataset. Users can find it at https://github.com/gabolsgabs/DALI and install it with pip.[1]

A song is represented as the Python class **`Annotations`**. Each instance has two attributes **`info`** and **`annotations`**. The attribute **`info`** contains the metadata, the link to the audio and the scores that indicate the quality of the annotations (Section 6.2).

The attribute **`annotations`** contains the aligned segments $a_{k,g}$. We can work in two modes, *`horizontal`* and *`vertical`*, and easily change from one to the other. The *`horizontal`* mode stores the granularity levels in isolation, providing access to all its segments $a_{k,g}$. The *`vertical`* mode connects levels vertically across the hierarchy. A segment at a given granularity contains all its deeper segments e.g. a line has links to all its words and notes, allowing the study of hierarchical relationships.

The package also includes additional tools for reading the dataset and automatically retrieving the audio from YouTube. We also provide tools for working with individual granularity levels, i.e., transforming the data into vectors or matrices with a given time resolution, manually correcting the alignment parameters, or re-computing the alignment process for different audio than the original one (Section 4.3). For a detailed explanation of how to use DALI, we refer to the tutorial: https://github.com/gabolsgabs/DALI.

Finally, using the correlation score that indicates the accuracy of the alignment (cf. Section 4.3), we propose to split DALI into 3 sets: train, validation, and test (**Table 4**). Other splits are possible depending on the task at hand (e.g. analyzing only English songs for lyrics alignment (Section 6.1)).

3.3 Distribution

Each DALI dataset version is a set of gzip files. Each file encloses an instance of the class **`Annotations`**. The different versions are distributed as open-source under an `Academic Free License (AFL)`[2] and can be found at https://zenodo.org/record/2577915 that provides a fingerprint (an `MD5` checksum file (Rivest, 1992)) that verifies their integrity. We describe them following the MIR corpora description (Peeters and Fort, 2012).

DALI is also part of mirdata (Bittner et al., 2019), a standard framework for MIR datasets that provides a fingerprint (also an `MD5` file) per **`Annotations`** instance and audio track that verifies their integrity.

3.4 Reproducibility

Our main problem is the restriction on sharing the audio tracks, which complicates the result comparison. This is common to other datasets with YouTube audio such as the Weimar Jazz Database (Balke et al., 2018). Using different audio may create misaligned annotations. We suggest three ways to overcome this issue:

1. to use our tools to retrieve the correct audio from YouTube. However, some links may be broken.
2. to use a different audio track and reproduce the alignment process (Section 4.3). We provide all the tools for this task and grant a model (second generation, Section 5.2) for computing the singing voice prediction vector needed.
3. to send us the feature extractor to be run on our audio. Users have to agree to distribute the new feature to other users (at `zenodo`) as we do with the f_0 representation computed in Section 6.2.

Table 2: DALI dataset general overview.

V	Songs	Artists	Genres	Languages	Decades
1.0	5358	2274	61	30	10
2.0	7756	2866	63	32	10

Table 4: Proposed split with respect to the time correlation values. NCC_t is defined at Section 4.3.

	Correlations	Tracks
Test	$NCC_t >= .94$	1.0: 167 2.0: 402
Validation	$.94 > NCC_t >= .925$	1.0: 423 2.0: 439
Train	$.925 > NCC_t >= .8$	1.0: 4768 2.0: 6915

Table 3: Statistics for the different DALI datasets. One song can have several genres.

V	Average songs per artist	Average duration per song	Full duration	Top 3 genres	Top 3 languages	Top 3 decades
1.0	2.36	Audio: 231.95s With vocals: 118.87s	Audio: 344.9hrs With vocals: 176.9hrs	Pop: 2662 Rock: 2079 Alternative: 869	ENG: 4018 GER: 434 FRA: 213	2000s: 2318 1990s: 1020 2010s: 668
2.0	2.71	Audio: 226.78s With vocals: 114.73s	Audio: 488.1hrs With vocals: 247.2hrs	Pop: 3726 Rock: 2794 Alternative: 1241	ENG: 5913 GER: 615 FRA: 304	2000s: 3248 1990s: 1409 2010s: 1153

4. Dataset creation

DALI stands as a solution to the absence of large reference datasets with lyrics and vocal notes aligned in time. These types of annotations are hard to obtain and very time-consuming to create. Our solution is to look outside the field of MIR. We turn our attention to karaoke video games where users sing along with the music. They win points comparing the sung melody with time and frequency aligned references. Therefore, large datasets of time-aligned melodic data and lyrics exist. Apart from commercial services, there are several active and large open-data communities. In those, non-expert users exchange text files with the reference annotations without any professional revision. We retrieved 13,339 of these files. However, they need to be adapted to the requirements of MIR applications.

4.1 From raw annotations to structured MIR data

Karaoke users create annotations and exchange them in text files that contain:

- the `song_title` and `artist_name`.
- a sequence of tuples {`time, musical-note, text`} with the actual annotations, **Figure 1**.
- the `offset_time` (start of the sequence) and the `frame rate` (annotation time-grid).

Table 5 defines the terms used in this article. We transform the raw annotation into useful data obtaining the time in seconds and the note's frequency (raw annotations express notes as intervals), and creating the four levels of granularity: notes (textual information underlying a note), words, lines and paragraphs. The first three levels are encoded in the retrieved files. We deduce the paragraph level as follows.

The paragraph level. Using the `song_title` and `artist_name`, we connect each raw annotation file to Wasabi, a semantic database of song metadata collected from various music databases (Meseguer-Brocal et al., 2017). Wasabi provides lyrics grouped in lines and paragraphs, in a text-only form. We create the paragraph level merging the two representations (melodic note-based annotations from karaoke annotations and text only from Wasabi) in a text to text alignment form. Let l^m be our existing raw lines and p^m the paragraph we want to obtain. Similarly, p^t represents the target paragraphs in Wasabi and l^t their lines. Our task is to progressively merge a set of l^m such that the new p^m is maximally similar to an existing p^t. This is not trivial since l^m and l^t differ in some regards. l^m tends to be shorter, some lines might be missing in one domain, and p^t can be rearranged/scrambled. An example of merging is shown in **Figure 2**.

The phoneme information. The phonetic information is computed only for the word level. We use the Grapheme-to-Phoneme (G2P)[3] system by CMU Sphinx[4] at Carnegie Mellon University. This model uses a `tensor2tensor` transformer architecture that relies on global dependencies between input and output. This information is only available for DALI version two.

The metadata. Wasabi provides extra multi-modal information such as cover images, links to video clips, metadata, biography, and expert comments.

4.2 Retrieving audio candidates

The annotations are now ready to be used. Nevertheless, they come without audio. The same `song_title` and `artist_name` may have many different versions (studio, radio, edit, live or remix) and each one can have a different lyrics alignment. Hence, we need to find the right version used by the karaoke users. Given a `song_title` and an `artist_name` Wasabi provides many possible versions. With this information, we query YouTube to recover a set of audio candidates. This is similar to other works that

Table 5: Overview of terms: definition of each term used in this article. NCC_t is defined at Section 4.3.

Term	Definition
`Notes`	time-aligned symbolic vocal melody annotations.
`Annotation`	basic alignment unit as a tuple of: time (start and duration in frames), musical note (with 0 = C3) and text.
`A file with annotations`	group of annotations that define the alignment of a particular song.
`Offset_time` (o)	the start of the annotations.
`Frame rate` (fr)	the reciprocal of the annotation grid size.
`Voice annotation sequence` ($vas(t) \in \{0,1\}$)	a vector that defines when the singing voice (SV) is active according to the karaoke-users' annotations.
`Predictions` ($\hat{p}(t) \in [0,1]$)	probability sequence indicating whether or not singing voice is active at any frame, provided by our singing voice detection.
`Labels`	label sequence of well-known ground truth datasets checked by the MIR community.
`Teacher`	SV detection (SVD) system used for selecting audio candidates and aligning the annotations to them.
`Student`	new SVD system trained on $vas(t)$ of the subset selected by the Teacher after $NCC(\hat{o}, \hat{fr}) \geq T_{corr}$.

used chroma features and diagonal matching to align jazz soli and audio candidates from YouTube (Balke et al., 2018).

To find the correct audio used by the karaoke-users we need to answer three questions:

1. Is the correct audio among the candidates?
2. If there are more than one correct audio, which one is the best?
3. Do annotations need to be adapted to the final audio to perfectly align with it?

Moreover, users are amateurs which may lead to errors. Depending on the amount of those, we may want to discard some files. This introduces a fourth question: are annotations good enough to be used?

4.3 Selecting the audio and adapting the annotation

To answer these questions, we measure the accuracy of aligned annotations to an audio candidate finding a common representation for both audio and text (Meseguer-Brocal et al., 2018).

We convert the audio into a singing-voice probability vector over time $\hat{p}(t)$, with $\hat{p}(t) \to 1$ when there is voice and $\hat{p}(t) \to 0$ otherwise. We call these $\hat{p}(t)$ predictions. The Singing Voice Detection (SVD) system computes $\hat{p}(t)$ from the audio signal represented as a sequence of patches of 80 log-mel bands over 115 time frames (0.014s per frame). Our system is based on the Deep Convolutional Neural Network (CNN) proposed by (Schlüter and Grill, 2015). **Figure 3** details the architecture of the network. The output of each patch corresponds to the $\hat{p}(t)$ of the central time frame.

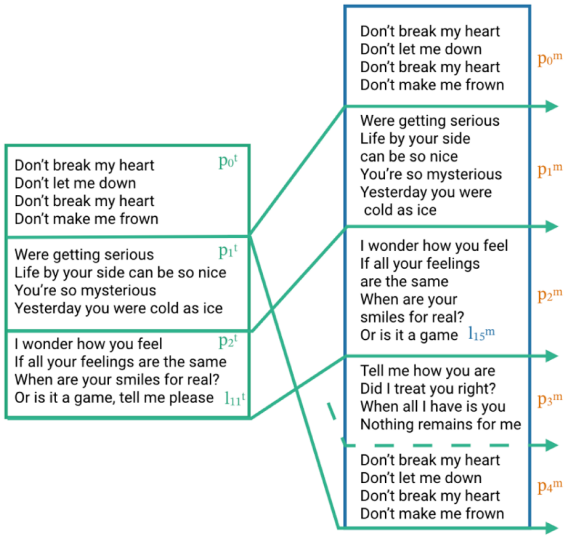

Figure 2: [Left part] Target lyrics lines and paragraphs as provided in WASABI. [Right part] The melody paragraphs p^m are created by merging the melody lines l^m into an existing target paragraph p^t. Note how line l^t_{11} in p^t_2 has no direct counterpart in l^m_* and verse p^m_3 does not appear in any p^t_*.

Likewise, we transform the lyrics annotations into *vas*(*t*), an voice annotation sequence with value 1 when there is vocal and 0 otherwise. Although we can construct *vas*(*t*) at any level of granularity, we work only with the note level because it is the finest.

Our hypothesis is that, with a highly accurate SVD system and exact annotations, both vectors *vas*(*t*) and $\hat{p}(t)$ should be identical. Consequently, it should be reasonably easy to use them to find the correct audio.

At this stage, audio and annotation are described as vectors over time $\hat{p}(t) \in [0,1]$ and $vas_{o,fr}(t) \in \{0,1\}$. We measure their similarity using the normalized cross-correlation (NCC). This similarity is not only important in recovering the right audio and finding the best alignment, but also in filtering imprecise annotations.

Since we are interested in global alignment we found this technique more precise than others such as Dynamic Time Warping (DTW). Indeed, DTW finds the minimal cost path for the alignment of two complete sequences. To do so, it can locally warp the annotations, this usually deforms them rather than correcting them. It is also costly to compute and its score is not directly normalized, which prevents us from selecting the right candidate.

The *vas*(*t*) depends on the parameters `offset_time` (*o*) and the `frame rate` (*fr*), $vas_{o,fr}(t)$. Hence, the alignment between $\hat{p}$ and *vas* depends on their correctness. While *o* defines the beginning of the annotations, *fr* controls the time grid size. When changing *fr*, the grid size is modified by a constant value that compresses or stretches the annotations as a whole respecting the global structure. Our *NCC* formula is as follows:

$$NCC(o, fr) = \frac{\Sigma_t \, vas_{o,fr}(t-o)\hat{p}(t)}{\sqrt{\Sigma_t \, vas_{o,fr}(t-o)^2}\sqrt{\Sigma_t \hat{p}(t)^2}}. \qquad (3)$$

For a particular *fr* value, *NCC*(*o*, *fr*) can be used to estimate the best $\hat{o}$ to align both sequences. We obtain the optimal $\hat{fr}$ using a brute force search in an interval α[5] of values around *fr*:

$$(\hat{fr}, \hat{o}) = \underset{fr \in [fr-\alpha, fr+\alpha], o}{\arg\max} \; NCC(o, fr). \qquad (4)$$

This automatically obtains the $\hat{o}$ and $\hat{fr}$ values that best align $\hat{p}(t)$ and $vas_{o,fr}(t)$ and yields a similarity value between 0 and 1. Given an annotation $vas_{o,fr}$, we compute $NCC(\hat{o}, \hat{fr})$ for all its audio candidates. Using it, we can find the best candidate (highest score) and establish if the final audio-annotation pair is good enough to keep.

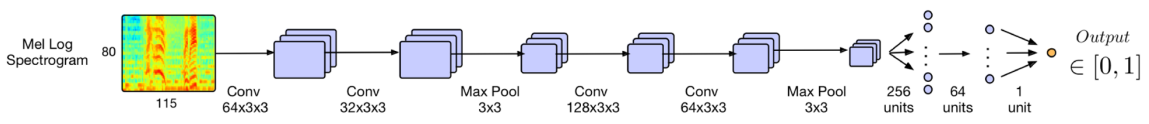

Figure 3: Architecture of our Singing Voice Detection (SVD) system using CNNs.

To this end, we set a threshold $NCC(\hat{o}, \hat{fr}) \geq T_{corr}$ to filter the audio-annotations, storing only the accurate pairs and discarding those for which the correct audio could not be found or those with insufficiently accurate annotations.

We establish the threshold (T_{corr} = 0.8) to ensure well-aligned audio-annotation pairs. This strategy is similar to active learning, where instead of labeling and using all possible data, we find ways of selecting the accurate data. The process is summarized in **Figure 4**.

5. Improving DALI

At this point and after manually examining the alignments, we noticed that the process strongly depends on $\hat{p}(t)$. The $\hat{p}(t)$ obtained with the baseline SVD systems is good enough to correctly identify the audio (although false negatives still exist), but not to align the annotations. Small variations lead to different alignments. Thus, we need to improve $\hat{p}(t)$. With improving $\hat{p}(t)$, we will find more suitable matches and align the annotations more precisely (more accurate $\hat{o}$ and $\hat{fr}$), which results in a better DALI.

There are two possibilities: to develop a novel SVD system or to train the existing architecture with better data. Since DALI is considerably larger (around 2000 songs) than similar datasets (around 100), we choose the latter. This idea re-uses all of the labeled data just created in the previous step to train a better SVD system. The resulting system can again be used to find more and better matches, improving and increasing the DALI dataset. This loop can be repeated iteratively. After our first iteration and using our best SVD system, we reach 5358 songs. We then perform a second iteration that defines the current version with 7756 songs. This process takes advantage of the data we just retrieved in a similar way to the teacher-student paradigm.

Note how our teachers do not label directly the input training data of the students but rather select the audio and align the annotations (Section 4.3). Similar to noisy label strategies in active learning, we use a threshold to separate good and bad data points. However, instead of doing this dynamically in training, we filter the data statically once an SVD model is trained.

This process is summarized in **Figure 5** and detailed by Meseguer-Brocal et al., (2018):

1- **Blue.** The retrieved karaoke annotation files are converted to an voice annotation sequence *vas*(*t*).
2- **Yellow.** We train an SVD (the teacher) either on ground-truth datasets or after the first iteration, on DALI annotations (green arrow in **Figure 5**, box 5).
3- **Red.** With the teacher's prediction $\hat{p}(t)$ and the *vas*(*t*) we compute the *NCC* to find the best audio candidate and alignment parameters $\hat{fr}, \hat{o}$ (Section 4.3).
4- **Purple.** We select the audio-annotation pairs with $NCC(\hat{o}, \hat{fr}) \geq T_{corr} = 0.8$. This defines a new training set (and DALI version).
5- **Green.** Using the new data we train a new SVD system, called the *student*. The new system is retrained from scratch, not adapting any previous system (no transfer learning).
6- **Yellow-Green.** The two systems, teacher and student, are compared on the ground-truth test sets.

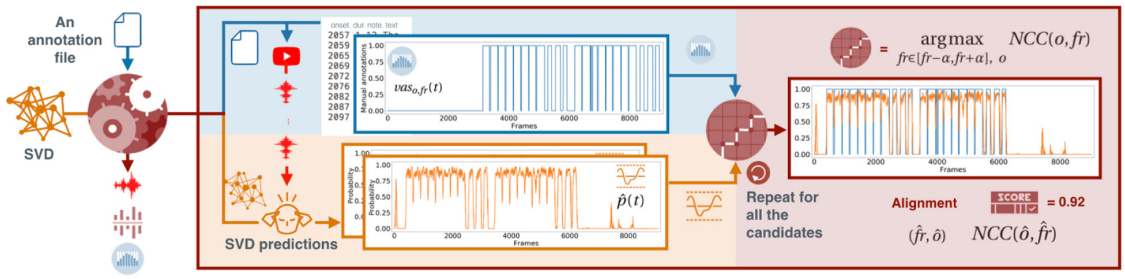

Figure 4: The input is an $vas_{o,fr}(t)$ (blue part – top left area) and a set of audio candidates retrieved from YouTube. The similarity estimation method uses an SVD model to convert each candidate in a $\hat{p}(t)$ (orange part – lower left area). We measure the similarity between the $vas_{o,fr}(t)$ and each $\hat{p}(t)$ using the cross-correlation method $\text{argmax}_{fr,o}\, NCC(o, fr)$ described in Section 4.3 (red part – right area). The output is the audio file with the highest $NCC(\hat{o}, \hat{fr})$ and the annotations aligned to it, according to the parameters $\hat{fr}$ and $\hat{o}$.

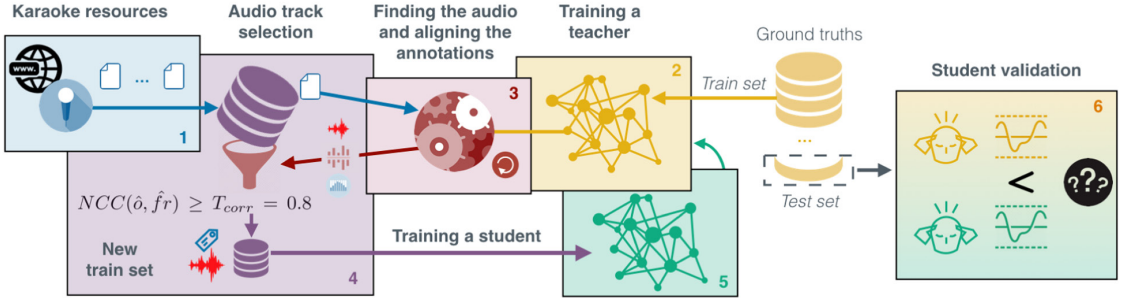

Figure 5: Creating the DALI dataset using the teacher-student paradigm.

This process incrementally adds more good audio-annotation pairs. We perform this three times as summarized in **Figure 6**. With this process, we are simultaneously improving the models and the dataset. Besides, this is also an indirect way of examining the quality of the retrieved annotations: a well-performing system trained with this data proves that the time alignment of the annotations is correct.

5.1 First generation

Datasets. Three ground-truth datasets are used to train the first teachers: Jamendo (Ramona et al., 2008), MedleyDB(Bittner et al., 2014) and a third that merges both, J+M. They are accurately labeled but small. Each dataset is split into a train, validation and test part using an artist filter.[6] We keep the test set of Jamendo and MedleyDB for evaluating the different SVD systems.

Teachers. We train three teachers using the training part of each ground-truth set. The teachers select the audio and align the annotations as described in Section 4.3. This creates three new datasets (DALI v0) with 2440, 2673 and 1596 audio-annotation pairs for the teacher: J+M, Jamendo and MedleyDB respectively.

Students. We train three different students. Among the possible target values: $\hat{p}$ given by the teacher (as is common in the teacher-student paradigm), *vas* after being aligned using *NCC*, or a combination of both; we use *vas*. We found this vector more accurate than $\hat{p}$. In our approach, the teacher 'filters' and 'corrects' the source of knowledge from which the student learns. Each student is trained with different data since each teacher may find different audio-annotation pairs or alignments (each one gets a different $\hat{p}$ which leads to different $\hat{fr}, \hat{o}$ values).

We hypothesize that if we have a more accurate $\hat{p}$, we can create a better DALI. In **Tables 6** and **7**, we observe how the students outperform the teachers in both the singing voice detection task and the alignment experiment. Thus, students compute better $\hat{p}$. Furthermore, we assume that if we use these SVD systems, we will retrieve better audio and have a more accurate alignment. For this reason, we use the student with highest results (based on J+M) to create DALI v1 with 5358 songs (Meseguer-Brocal et al., 2018).

5.2 Second generation

We now use as a teacher the best student of the first iteration, the student J+M. We repeat again the process described in Section 4.3. In this case, the teacher is not trained on any ground truth but on DALI v1, using as target the aligned *vas*(*t*). We split DALI v1 (5358 tracks) into 5253 for training, 100 for validation (the ones with higher NCC) and 105 for testing. We manually annotated the test set finding the optimal $\hat{fr}$ and $\hat{o}$. This is our ground-truth for future experiments.[7]

The new SVD (student of the second generation) obtains even better results in the singing voice detection task.

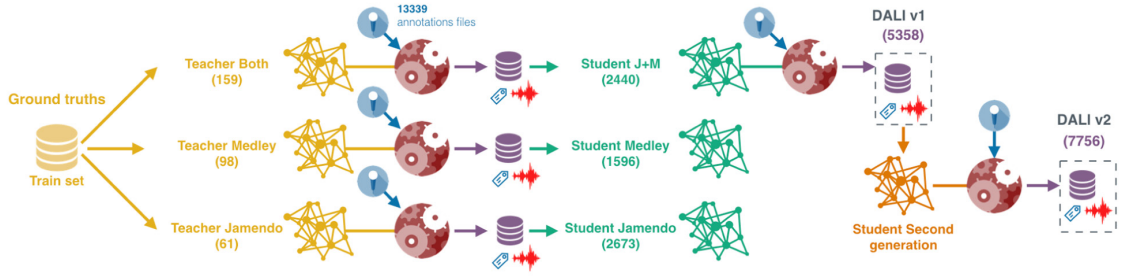

Figure 6: We create three SVD systems (teachers) using the ground truth datasets (Jamendo, MedleyDB and Both). The three systems generate three new datasets (DALI v0) used to train three new SVD systems (the first-generation students). Now, we use the best student, J+M, to define DALI v1 (Meseguer-Brocal et al., 2018). We train a second-generation student using DALI v1 and create DALI v2.

Table 6: Singing voice detection performance, measured as mean accuracy and standard deviation. Number of tracks is shown in parentheses. Nomenclature: T = Teacher, S = Student, J = Jamendo, M = MedleyDB, J+M = Jamendo + MedleyDB, 2G = second generation; in brackets we specify the name of the teacher used for training a student.

Test sets / SVD system	J_Test (16)	M_Test (36)	J_Test+Train (77)	M_Test+Train (98)
T_J_Train (61)	88.95% ± 5.71	83.27% ± 16.6	–	81.83% ± 16.8
S [T_J_Train] (2673)	87.08% ± 6.75	82.05% ± 15.3	87.87% ± 6.34	84.00% ± 13.9
T_M_Train (98)	76.61% ± 12.5	84.14% ± 17.4	76.32% ± 11.2	–
S [T_M_Train] (1596)	82.73% ± 10.6	79.89% ± 17.8	84.12% ± 9.00	82.03% ± 16.4
T_J+M_Train (159)	83.63% ± 7.13	83.24% ± 13.9	–	–
S [T_J+M_Train] (2440)	87.79% ± 8.82	85.87% ± 13.6	89.09% ± 6.21	86.78% ± 12.3
2G [S [T_J+M_Train]] (5253)	**93.37% ± 3.61**	**88.64% ± 13.0**	**92.70% ± 3.85**	**88.90% ± 11.7**

Table 7: Alignment performance for the teachers and students: mean offset deviation in seconds, mean frame rate deviation in frames, and pos is position in the classification.

	mean offset rank	pos	mean $offset_d$	mean fr rank	pos	mean fr_d
T_J_Train (61)	2.79 ± .48	4	0.082 ± 0.17	1.18 ± .41	4	0.51 ± 1.24
S [T_J_Train] (2673)	2.37 ± .19	3	0.046 ± 0.05	1.06 ± .23	3	0.25 ± 0.88
T_M_Train (98)	4.85 ± .50	7	0.716 ± 2.74	1.89 ± .72	7	2.65 ± 2.96
S [T_M_Train] (1596)	4.29 ± .37	6	0.164 ± 0.10	1.30 ± .48	5	0.88 ± 1.85
T_J+M_Train (159)	3.42 ± .58	5	0.370 ± 1.55	1.47 ± .68	6	1.29 ± 2.29
S [T_J+M_Train] (2440)	2.23 ± .07	2	0.043 ± 0.05	1.04 ± .19	2	0.25 ± 0.85
2G [S [T_J+M_Train]] (5253)	**1.82 ± .07**	**1**	**0.036 ± 0.06**	**1.01 ± .10**	**1**	**0.21 ± 0.83**

Hence, we assume that another repetition of the process will output a better DALI. This is indeed the DALI v2 with 7756 audio-annotation pairs.

This experiment proves that students work much better in the cross-dataset scenario (real generalization) when the train-set and test-set are from different datasets. It is important to note that the accuracy of the student networks is higher than that of the teachers, even if they have been trained on imperfect data.

6. Dataset analysis

6.1 On alignment

We hypothesize that a more accurate prediction $\hat{p}$ leads to a better DALI. To prove this, we measure the precision of the $\hat{fr}$ and $\hat{o}$ of each SVD system. To this end, we define a ground-truth dataset by manually annotating 105 songs of DALI v1, i.e. finding the $\hat{fr}$ and $\hat{o}$ values that give the best global alignment. We measure then how far the estimated $\hat{fr}$ and $\hat{o}$ are from the manually annotated ones. We name these deviations $offset_d$ and fr_d.

Results are indicated in **Table 7**. We estimate the average $offset_d$, fr_d and the mean rank. For instance, four systems a, b, c and d with offset deviations 0.057, 0.049, 0.057 and 0.063 seconds are ranked as: b = 1st, a = 2nd, c = 2nd and d = 3rd, respectively. The mean rank per model is the average of all individual ranks per song.

Baseline SVD. The main motivation to improve $\hat{p}$ is that the alignment observed with the baseline SVD systems is not sufficiently precise. This experiment quantifies this judgment. The T_M_Train and T_J+M_Train are ranked last in finding both the correct offset and frame rate. Their values are considerably different to the ground-truth and produce unacceptable alignments. Remarkably, the T_J_Train is much better and its results are comparable to the student networks.

First students. Each student exceeds its teacher with consistently higher rank and lower deviations. S[T_J+M_Train] is the best student. This is surprising because it is not trained with particularly well-aligned data (its teacher T_J+M_Train is placed 5th and 6th for *offset* and *fr*). Yet it scores almost as well as the S[T_J_Train], which was trained with better aligned data (its teacher is the best teacher). We presume an error tolerance in the singing voice detection task, which is not critical below an unknown value, but crucial above it: S[T_M_Train] (trained with the most misaligned data) is worse than the other students.

Second iteration. The Second Generation has the lowest deviations and the best results, being placed first for both rankings. However, the increase is moderate. We presume we are reaching the limit of the alignment precision that we can achieve with the *NCC*.

These results, together with those in **Table 6** prove that DALI is improving at each iteration.

6.2 Quality of the annotations

We can only guarantee that each new DALI version has better audio-annotation pairs with a more accurate global alignment. But there is still the recurring question: how good are the annotations?. After manually analyzing the errors we can group them into two types.

Global errors. They affect the song as a whole and are the least frequent issues. We define two groups.

- **Time:** The most common global time errors are those with misaligned sections despite the audio-annotation pair having a high *NCC* and good *o* and *fr* values. In these cases, each section has a different offset. Moreover, there are songs with one or more missed sections. This is likely to occur when several vocalists sing at the same time or when a chorus is repeated at the end but not its lyrics.
- **Frequency:** Raw annotations store the notes as interval differences with respect to an unknown reference. Most songs use C4. But, some use a different reference.

To solve the global frequency errors, we perform a correlation in frequency between the note level $(a_{k,notes})_{k=0}^{K_{notes}} = (t_k^0, t_k^1, f_k, l_k, i_k)_{notes} \rightarrow (t_k^0, t_k^1, f_k)_{notes}$ and the extracted fundamental note frequency (f_0) (Doras et al., 2019). The f_0 is a matrix over time where each frame stores the note likelihoods obtained directly from the original audio. We compress the original f_0 to 6 octaves, 1 bin per semitone and a time resolution of 0.058s. Similar to the process done in Section 4.3, we transform the annotations ($a_{k,\ notes}$) into a matrix. Unlike the previous correlation, we measure correlation along the frequency axis and not the time axis. We then simply transpose all

the f_k in the $a_{k,\ notes}$ by the same value. The transposition values covers all the frequency range in the f_0. We find the frequency scaling factor that maximizes the energy between the annotated frequencies $a_{k,\ notes}$ and the estimated f_0. This defines the correct global frequency position of the annotations.

Moreover, we calculate the new "flexible" versions of the melody metrics Overall Accuracy, Raw Pitch Accuracy, Raw Chroma Accuracy, Voicing Recall, Voicing False Alarm (Bittner and Bosch, 2019). These metrics add new extra knowledge for understanding the quality of the annotations. Together with the NCC, they can guide us to know which annotations are good and which ones are of lesser quality. We can assume that a high Raw Pitch Accuracy suggests a good alignment. However and since the f_0 has errors, low metrics do not necessarily indicate bad alignment.

Local errors. These errors (**Figure 7**) occur because users are non-professionals. They cover local segment alignments, text misspellings and single note errors:

- **Time:** Errors in the positions of the start or end t_k^0 and t_k^1. Notes are placed in the wrong position in time or have the wrong duration.
- **Frequency:** Errors in f_k. These errors are quite arbitrary and include octave, semitone and other harmonic interval errors such as major third, perfect fourth or perfect fifth.
- **Text:** Misspellings in l_k. Moreover, there are also errors in the phoneme level due to the automatic process employed.
- **Missing:** Occasional missing notes during humming, "oh"s or similar parts.

To quantify such local issues, it would be necessary to manually review the annotations one by one. This is demanding and time-consuming. Indeed, this is what we aim to avoid. Although solving these errors remains for future work, we can try to infer where they occurred using the time evolution of the f_0 correlation. This vector can filter the annotations, selecting only those segments with high correlation. The f_0 representation and the correlation vector are jointly accessible with the annotations.

7. Discussion and further work

In this article, we improved the work of Meseguer-Brocal et al. (2018) and presented the second version of the DALI dataset: a large and rich multimodal dataset that contains 7756 songs with their time-aligned vocal melody notes and lyrics at four levels of granularity. We defined and detailed DALI, highlighting the variety of artists, genres, epochs, and languages. We created the tools for working with DALI as well as diverse solutions to get the matched audio.

We explained our methodology that builds a loop where dataset creation and model learning interact in a way that benefits each other. Our approach is motivated by active learning and the teacher-student paradigm. The time-aligned lyrics and notes come from karaoke resources where non-expert users manually annotated lyrics as a sequence of time-aligned notes with their associated textual information. From the textual information, we derived different levels of granularity. We then linked each annotation to the correct audio and globally aligned the annotations with it. To improve the alignment, we iteratively improved the SVD using the teacher-student paradigm. Through experiments, we showed that the students outperform the teachers notably in two tasks: alignment and singing voice detection. We showed that in our context it is better to have imperfect but large datasets rather than small and perfect ones.

Finally, we analyzed the quality of the annotations and proposed a solution to the global alignment problems. Indirectly and during the SVD improvement step, we confirmed that the current time alignment of the annotations is good enough to create a state-of-the-art SVD system. We also presented a way of measuring data quality via f_0 correlations.

However, there is room for improvement. In this iteration of the DALI dataset, our goal was to find the matching audio candidate and globally align the annotations to it. But we still have false positives. Once we have a good global alignment we can face local issues. We plan to apply source separation techniques to extract the vocals from the mixture and use local warping techniques such as DTW, Viterbi decoding or Beam Search Decoding. However, these techniques do not guarantee that the new annotation version is better than the original one, requiring a manual verification every time a new alignment is created. It is essential then to face the core question: how can we automatically evaluate the quality of the annotations? We plan to address this question, which should help with both reducing the number of false positives in the global alignment and finding the appropriate local corrections. Having such a way to measure quality will also help with filtering false-negatives to get closer to the 13,339 annotations retrieved from the Web.

DALI represents a great challenge. It has a large number of imperfect annotations that have the potential to make our field move forward. But we need to solve its issues which requires ways to automatically identify and quantify

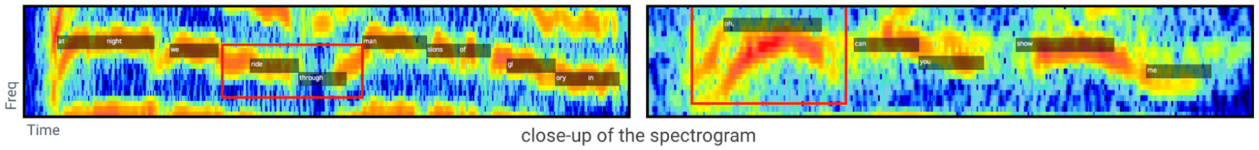

Figure 7: Local errors still present in DALI. [Left] Misalignments in time due to an imperfect annotation. [Right] An individual mis-annotated note. These problems remain for future versions of the dataset.

such imperfect annotations. Otherwise, correcting them is costly and time-consuming. Besides, the deep learning models we use also possess imperfections that are hard to quantify. This means that to create DALI, we have to deal with two sources of imperfect information. This puzzle is also common in other machine learning domains. Our solution creates a loop in which machine learning models are employed to filter and enhance imperfect data, which is then used to improve those same models. We prove that this loop benefits both the model creation and the data curation. We believe that DALI can be an inspiration to our community to not treat model learning and dataset creation as independent tasks, but rather as complementary processes.

Notes

[1] https://pypi.org/project/DALI-dataset/.
[2] https://opensource.org/licenses/AFL-3.0.
[3] https://github.com/cmusphinx/g2p-seq2seq.
[4] https://cmusphinx.github.io/wiki/.
[5] we use $\alpha = fr * 0.05$.
[6] No artist who appears in the training set can appear in the test set.
[7] Note that this split is different from that proposed in Section 3 because of the nature of the experiments carried out in Section 6.1.

Acknowledgement

This research has received funding from the French National Research Agency under the contract ANR-16-CE23-0017-01 (WASABI project).

Competing Interests

The authors have no competing interests to declare.

References

Ashok, A., Rhinehart, N., Beainy, F., & **Kitani, K. M.** (2017). N2N learning: Network to network compression via policy gradient reinforcement learning. CoRR, abs/1709.06030.

Balke, S., Dittmar, C., Abeßer, J., Frieler, K., Pfleiderer, M., & **Müller, M.** (2018). Bridging the gap: Enriching YouTube videos with jazz music annotations. Frontiers in Digital Humanities, 5:1. DOI: https://doi.org/10.3389/fdigh.2018.00001

Benzi, K., Defferrard, M., Vandergheynst, P., & **Bresson, X.** (2016). FMA: A dataset for music analysis. CoRR, abs/1612.01840.

Bittner, R., & **Bosch, J. J.** (2019). Generalized metrics for single-f0 estimation evaluation. In *Proceedings of 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands.

Bittner, R., Fuentes, M., Rubinstein, D., Jansson, A., Choi, K., & **Kell, T.** (2019). mirdata: Software for reproducible usage of datasets. In *Proceedings of 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands.

Bittner, R., Salamon, J., Tierney, M., Mauch, M., Cannam, C., & **Bello, J.** (2014). MedleyDB: A multitrack dataset for annotation-intensive MIR research. In *Proceedings of 15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan.

Bucilua, C., Caruana, R., & **Niculescu-Mizil, A.** (2006). Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '06*, page 535–541, New York, NY, USA. Association for Computing Machinery. DOI: https://doi.org/10.1145/1150402.1150464

Cui, J., Kingsbury, B., Ramabhadran, B., Saon, G., Sercu, T., Audhkhasi, K., Sethy, A., Nussbaum-Thom, M., & **Rosenberg, A.** (2017). Knowledge distillation across ensembles of multilingual models for low-resource languages. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. DOI: https://doi.org/10.1109/ICASSP.2017.7953073

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & **Fei-Fei, L.** (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. DOI: https://doi.org/10.1109/CVPR.2009.5206848

Donahue, C., Henry Mao, H., & **McAuley, J.** (2018). The NES Music Database: A multi-instrumental dataset with expressive performance attributes. In *Proceedings of 19th International Society for Music Information Retrieval Conference*, Paris, France.

Doras, G., Esling, P., & **Peeters, G.** (2019). On the use of u-net for dominant melody estimation in polyphonic music. In *2019 International Workshop on Multilayer Music Representation and Processing (MMRP)*, pages 66–70. DOI: https://doi.org/10.1109/MMRP.2019.00020

Dzhambazov, G. (2017). Knowledge-based Probabilistic Modeling for Tracking Lyrics in Music Audio Signals. PhD thesis, Universitat Pompeu Fabra.

Fonseca, E., Pons, J., Favory, X., Font, F., Bogdanov, D., Ferraro, A., Oramas, S., Porter, A., & **Serra, X.** (2017). Freesound datasets: A platform for the creation of open audio datasets. In *Proceedings of 18th International Society for Music Information Retrieval Conference*, Suzhou, China.

Fujihara, H., & **Goto, M.** (2012). Lyrics-to-audio alignment and its application. In *Multimodal Music Processing, volume 3 of Dagstuhl Follow-Ups*, pages 23–36. Dagstuhl, Germany.

Fujihara, H., Goto, M., Ogata, J., & **Okuno, H. G.** (2011). LyricSynchronizer: Automatic synchronization system between musical audio signals and lyrics. *IEEE Journal of Selected Topics in Signal Processing*, *5*(6), 1252–1261. DOI: https://doi.org/10.1109/JSTSP.2011.2159577

Goto, M. (2014). Singing information processing. In *12th International Conference on Signal Processing*, pages 2431–2438. DOI: https://doi.org/10.1109/ICOSP.2014.7015431

Gupta, C., Tong, R., Li, H., & **Wang, Y.** (2018). Semi-supervised lyrics and solo-singing alignment. In *Proceedings of 19th International Society for Music Information Retrieval Conference*.

Gupta, C., Yılmaz, E., & **Li, H.** (2019). Acoustic modeling for automatic lyrics-to-audio alignment. arXiv preprint

arXiv:1906.10369. DOI: https://doi.org/10.21437/Interspeech.2019-1520

Hansen, J. K. (2012). Recognition of phonemes in acappella recordings using temporal patterns and mel frequency cepstral coefficients. In *Proceedings of 9th Sound and Music Computing Conference*, Copenhagen, Denmark.

Hinton, G., Vinyals, O., & **Dean, J.** (2014). Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Humphrey, E. J., Montecchio, N., Bittner, R., Jansson, A., & **Jehan, T.** (2017). Mining labelled data from web-scale collections for vocal activity detection in music. In *Proceedings of 18th International Society for Music Information Retrieval Conference*, Suzhou, China.

Iskandar, D., Wang, Y., Kan, M.-Y., & **Li, H.** (2006). Syllabic level automatic synchronization of music signals and text lyrics. In *Proceedings of the 14th ACM International Conference on Multimedia, MM '06*, pages 659–662, New York, NY, USA. ACM. DOI: https://doi.org/10.1145/1180639.1180777

Kan, M.-Y., Wang, Y., Iskandar, D., Nwe, T. L., & **Shenoy, A.** (2008). LyricAlly: Automatic synchronization of textual lyrics to acoustic music signals. *IEEE Transactions on Audio, Speech, and Language Processing, 16*(2), 338–349. DOI: https://doi.org/10.1109/TASL.2007.911559

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, Department of Computer Science.

Kruspe, A. M. (2016). Bootstrapping a system for phoneme recognition and keyword spotting in unaccompanied singing. In *Proceedings of 17th International Society for Music Information Retrieval Conference*, pages 358–364, New York City, United States.

Le Cun, Y., Bottou, L., Bengio, Y., & **Haffner, P.** (1998). Gradient based learning applied to document recognition. *Proceedings of IEEE*, *86*(11), 2278–2324. DOI: https://doi.org/10.1109/5.726791

Lee, S. W., & **Scott, J.** (2017). Word level lyrics-audio synchronization using separated vocals. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. DOI: https://doi.org/10.1109/ICASSP.2017.7952235

Maia, L., Fuentes, M., Biscainho, L., Rocamora, M., & **Essid, S.** (2019). SAMBASET: A dataset of historical samba de enredo recordings for computational music analysis. In *Proceedings of 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands.

Mauch, M., Fujihara, H., & **Goto, M.** (2012). Integrating additional chord information into HMM-based lyrics-to-audio alignment. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*, 200–210. DOI: https://doi.org/10.1109/TASL.2011.2159595

Mesaros, A. (2013). Singing voice identification and lyrics transcription for music information retrieval. In *7th Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pages 1–10. DOI: https://doi.org/10.1109/SpeD.2013.6682644

Mesaros, A., & **Virtanen, T.** (2010). Automatic recognition of lyrics in singing. *EURASIP Journal on Audio, Speech, and Music Processing, 2010*, 1–11. DOI: https://doi.org/10.1155/2010/546047

Meseguer-Brocal, G., Cohen-Hadria, A., & **Peeters, G.** (2018). Dali: a large dataset of synchronised audio, lyrics and notes, automatically created using teacher-student machine learning paradigm. In *Proceedings of 19th International Society for Music Information Retrieval Conference*, Paris, France.

Meseguer-Brocal, G., Peeters, G., Pellerin, G., Buffa, M., Cabrio, E., Faron Zucker, C., Giboin, A., Mirbel, I., Hennequin, R., Moussallam, M., Piccoli, F., & **Fillon, T.** (2017). WASABI: A two million song database project with audio and cultural metadata plus WebAudio enhanced client applications. In *Web Audio Conference*, London, U.K.

Müller, M., Kurth, F., Damm, D., Fremerey, C., & **Clausen, M.** (2007). Lyrics-based audio retrieval and multimodal navigation in music collections. In Kovács, L., Fuhr, N., & Meghini, C., editors, *Research and Advanced Technology for Digital Libraries*, pages 112–123. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-540-74851-9_10

Nieto, O., McCallum, M., Davies, M., Robertson, A., Stark, A., & **Egozy, E.** (2019). The Harmonix set: Beats, downbeats, and functional segment annotations of Western popular music. In *Proceedings of 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands.

Peeters, G., & **Fort, K.** (2012). Towards a (better) definition of annotated MIR corpora. In *Proceedings of 13th International Society for Music Information Retrieval Conference*, Porto, Portugal.

Ramona, M., Richard, G., & **David, B.** (2008). Vocal detection in music with support vector machines. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. DOI: https://doi.org/10.1109/ICASSP.2008.4518002

Rivest, R. (1992). The MD5 message-digest algorithm. RFC 1321, Internet Engineering Task Force Network Working Group. DOI: https://doi.org/10.17487/rfc1321

Schlüter, J., & **Grill, T.** (2015). Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks. In *Proceedings of 16th International Society for Music Information Retrieval Conference*, Malaga, Spain.

Settles, B. (2008). Curious Machines: Active Learning with Structured Instances. PhD thesis, Stanford University, Music Department.

Smith, J. (2013). Correlation Analyses of Encoded Music Performance. PhD thesis, Stanford University, Music Department.

Stoller, D., Durand, S., & **Ewert, S.** (2019). End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5275–5279. DOI: https://doi.org/10.1109/ICASSP.2019.8683470

Watanabe, S., Hori, T., Le Roux, J., & **Hershey, J.** (2017). Student-teacher network learning with enhanced features. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5275–5279. DOI: https://doi.org/10.1109/ICASSP.2017.7953163

Wong, C. H., Szeto, W. M., & **Wong, K. H.** (2007). Automatic lyrics alignment for Cantonese popular music. *Multimedia Systems, 12*(4), 307–323. DOI: https://doi.org/10.1007/s00530-006-0055-8

Wu, C., & **Lerch, A.** (2017). Automatic drum transcription using the student-teacher learning paradigm with unlabeled music data. In *Proceedings of 18th International Society for Music Information Retrieval Conference*, Suzhou, China.

Yesiler, F., Tralie, C., Correya, A., Furtado Silva, D., Tovstogan, P., Gomez, E., & **Serra, X.** (2019). Da-TACOS: A dataset for cover song identification and understanding. In *Proceedings of 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands.

How to cite this article: Meseguer-Brocal, G., Cohen-Hadria, A., & Peeters, G. (2020). Creating DALI, a Large Dataset of Synchronized Audio, Lyrics, and Notes. *Transactions of the International Society for Music Information Retrieval,* 3(1), pp. 55–67. DOI: https://doi.org/10.5334/tismir.30

Submitted: 24 January 2019 **Accepted:** 09 April 2020 **Published:** 11 June 2020

]u[*Transactions of the International Society for Music Information Retrieval* is a peer-reviewed open access journal published by Ubiquity Press.

OPEN ACCESS