# THE WORDS REMAIN THE SAME: COVER DETECTION WITH LYRICS TRANSCRIPTION

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

## ABSTRACT

Cover detection has gained sustained interest in the scientific community and has recently made significant progress both in terms of scalability and accuracy. However, most approaches are based on the estimation of harmonic and melodic features and neglect lyrics information although it is an important invariant across covers. In this work, we propose a novel approach leveraging lyrics without requiring access to full texts though the use of lyrics recognition on audio. Our approach relies on the fusion of a singing voice recognition framework and a more classic tonal-based cover detection method. To the best of our knowledge, this is the first time that lyrics estimation from audio has been explicitly used for cover detection. Furthermore, we exploit efficient string matching and an approximated nearest neighbors search algorithm which lead to a scalable system which is able to operate on very large databases. Extensive experiments on the largest publicly available cover detection dataset demonstrate the validity of using lyrics information for this task.

## 1. INTRODUCTION

Cover detection, also known as version identification, aims at detecting whether two recordings are of the same underlying musical work. A cover can be played by the same artist as the original song, or by another artist, and can be quite similar or vastly different. Generally, it is assumed, as in [1], that tonal progression features (chord, melody, and harmony) are mostly preserved between covers of the same work. Inversely, musical attributes such as key, timbre, tempo, and structure significantly vary across covers [1]. Variations of these features between covers were extensively studied in [2]. Cover detection systems are then built to be insensitive to these variations and exploit tonal progression features. The task has been frequently studied as a *query and answer* [1] one, i.e. given an input query, the system outputs a ranked list of possible covers from a music collection. True covers are to be ranked as highly as possible while other songs should be ranked low. This list is usually obtained by computing pairwise similarities between the query and each song of a pre-defined dataset [1]. If earlier cover detection systems were shown to be highly efficient on small datasets (1000 songs or less) [3], performances quickly dropped on larger ones [2, 4]. Recent works have made significant advances in scalability and accuracy, for larger datasets, taking inspiration from metric learning [5] and knowledge distillation [6].

Almost none of the existing approaches explicitly consider the textual information provided by the lyrics. To the best of our knowledge, it is only used in [4], in which lyrics are assumed to be available for a significant part of the dataset. In this paper, the authors use metadata and lyrics alongside audio to perform cover detection. The textual similarity of lyrics and song titles is computed using a plain Bag-of-word *Term Frequency–Inverse Document Frequency* (TFIDF). The authors show that results obtained with lyrics are on par with those given by audio-based features on a large-scale dataset. Moreover, the best results are obtained when combining all of the features. However, each feature is only used in a separate part of a multi-layer database pruning method; the information carried by each modality is thus not optimally combined. One limitation of this work is that it assumes that the lyrics of most songs are available. Considering the task of query by singing, which may be regarded as a related task to the cover detection one, authors in [7] employed lyrics and melody recognition to recognize a singing query and match it against a collection of songs. They employed a basic bigram *Hidden Markov model* (HMM) model that is trained on speech and adapted to singing voice. However, this approach also presupposes that the lyrics of songs are available. Lyrics from the considered dataset are, in fact, utilized to inform singing voice recognition.

While this assumption arguably does not hold for large musical collections, one could turn to *Singing Voice Recognition* (SVR) frameworks to retrieve a noisy estimate of the lyrics. We thus propose a novel cover detection approach leveraging lyrics information extracted from audio. It is based on the fusion of a SVR framework and a more classic tonal-based cover detection system. Based on our review of the literature, this is the first time that an estimation of lyrics transcripts from audio has been explicitly leveraged

**Attribution:** A. Vaglio, R. Hennequin, M. Moussallam and G. Richard, "The words remain the same: cover detection with lyrics transcription", in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.,* Online, 2021.

to perform cover detection. Our assumption, based on the results of [4], is that lyrics are often preserved between covers in popular western music. For the first modality of our fused system, we thus propose using transcription methods to obtain estimates of these lyrics for all songs. The cover song here is framed as a noisy text-matching task. We expect a lyrics-recognition based system to be particularly relevant for pairs of covers displaying hugely different tonal features while using the same lyrics. An example of such cases is the cover of *Summertime* by *Janis Joplin* where the harmony and melody are considerably different from the original score, but the lyrics remain quite similar. Nevertheless, it is clear that a pure lyrics-based system is inadequate for instrumental music (e.g. without a singing voice). Therefore, we use a tonal-based system such as the second modality of our fused system. An instrumental detector is applied on the output of the lyrics recognition framework to inform the fusion strategy. We provide extensive empirical evidence that both modalities are indeed complementary. Extra attention is placed on the scalability of our proposed approach using *Approximated Nearest Neighbors* (ANN) methods.

## 2. RELATED WORKS

Classically, cover song detection systems use tonal features, which are thought to be the least altered between a song and its covers. Chroma [8] and derived features such as *Harmonic Pitch Class Profil* (HPCP) [3] and CremaPCP [5] are among most effective examples. Before computing the similarity between two songs, multiple preprocessing steps can be applied to obtain features that are invariant to the key [9], the tempo [10], or the structure of the song [5]. After extracting the features to be compared for both songs, a cross similarity matrix [11], or a cross recurrent plot [9], is then generally computed. A similarity score is then computed using dynamic programming like *Dynamic Time Warping* (DTW) [12] and recurrence quantification analysis [3]. For a given query, this score is calculated for all tracks in a pre-defined dataset and thus yields the desired sorted list. These methods achieve satisfactory results for small datasets of up to a thousand songs [3], but are computationally costly for larger datasets.

To address this issue, some authors have attempted to reduce the size of the input representation to obtain a low-dimensional fixed size representation for each track. The similarity comparison thus boils down to a basic distance metric such as Euclidean distance or cosine similarity [5] that are much faster than dynamic programming algorithms of quadratic complexity. Early approaches of this type include using fingerprinting in the form of Chroma landmarks [10] and $2D$ Fourier transform of Chroma vectors [13], both obtaining low performances. More recent approaches using metric learning, triplet loss, and distillation methods show greater improvement [5, 6] in terms of computation speed and retrieval performances. Database pruning was also used to decrease the overall complexity in [4, 14]. A first fast global candidate selection using text and metadata was performed, followed by a more complex similarity function to re-rank the subset. Most of these approaches, which are based on low-dimensional embeddings and simple distance functions, are then simply exploited into existing scalable nearest-neighbors methods. For example, the authors in [10, 15] use index-based matching on extracted audio fingerprinting. Scalable nearest-neighbors methods are more broadly discussed in Section 3.6.

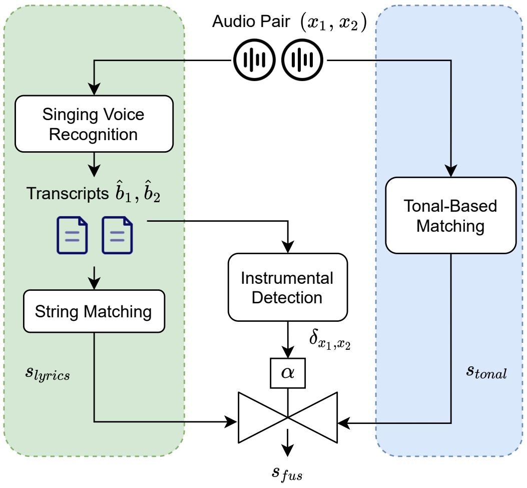

**Figure 1**. Audio from a pair of tracks is processed in parallel by two branches computing lyrics and tonal-based similarities respectively. The fusion mechanism is informed by an *instrumental* detection on the transcripts

## 3. PROPOSED APPROACH

A general overview of our approach is described in Figure 1. It is composed of a lyrics-recognition based cover detection system and a classic tonal-based system. The first branch is constituted of a lyrics recognition framework and a string matching function. It takes two songs $x_1$ and $x_2$ as input and outputs the respective estimated lyrics $\hat{b_1}$ and $\hat{b_2}$. A similarity estimation $s_{lyrics}$ is then obtained using these transcriptions. The second branch of our approach, the classic tonal-based system, also takes these two songs as input and outputs a similarity estimation $s_{tonal}$. They are then fused using a fusion function to obtain a new similarity estimation $s_{fus}$. Extra input is added to the fusion function $\alpha$ to weigh the participation of both modalities during the fusion. The value of this input depends on the instrumental detector taking as input both transcripts and outputting the probability that at least one of the tracks is purely instrumental. This avoids using the lyrics-based recognition system during the fusion in the absence of lyrics. To obtain the desired sorted list, for a given query, a similarity is then computed for the considered system between the query and each track of the dataset. Finally, a fast approximate index search technique is used on our system to make it scalable. In our work, we rely on the ANN approach where the similarity is only computed between the query and the nearest neighbors returned by the method.

### 3.1 Lyrics recognition

We choose a state-of-the-art framework [16] that obtained the best results in the *Music Information Retrieval Evaluation eXchange* (MIREX) 2020 lyrics transcription challenge [1]. It uses an acoustic model composed of several layers of *Time Delay Neural Network* (TDNN) that are trained using the English tracks of the DALI dataset [17]. Background music is directly modeled as an output of the acoustic model as such that it does not use any preprocessing step of *Singing Voice Separation* (SVS). Moreover, phoneme units are annotated with genre labelling information. An extended lexicon is also employed to handle long-vowel duration. Finally, a 3-gram word language model with interpolated Kneser-Ney smoothing is trained on the English portion of DALI lyrics. The complete framework extracts *Mel-Frequency Cepstral Coefficients* (MFCC) of dimension $40$ from the input audio and outputs transcribed English words. As this model cannot output non-English words, extra care on the results of non-English tracks will be considered later in this paper. The acoustic model and lexicon are collected from the code implementation of the authors [2]. We compute the language model with the kenLM toolkit [18]. The vocabulary of the language model is restricted to the $6000$ most frequent words, thus reducing overfitting. Obtained transcription results are on par with those in MIREX with a *Word Error rate* (WER) of $62\%$ on the Jamendo dataset [19].

### 3.2 String matching

To allow for a swift computation of the similarity between pairs of estimated transcripts, each string is transformed to a vector using a TFIDF based on a 3-gram at character level with IDF values computed from the DALI dataset. The complexity of this type of algorithm is $O(m + n)$ with $m$ and $n$, which are the respective length of each transcript. The similarity is then simply given using a cosine similarity, which is independent of the length of each transcript. Using a word level 3-gram was not shown to improve performances on a cover song tuning set described in Section 4. We also considered the Levenshtein distance for the string matching, but it did not yield significant gains in performances while inducing a quadratic complexity.

### 3.3 Detecting instrumentals

Looking at various transcripts given by our SVR framework, we notice that, for most instrumental tracks, the transcript obtained is composed of either a very few number of words, or highly repeated ones such as onomatopoeia. Therefore, we consider a track as instrumental if the respective transcription is composed of less than $l$ different words with $l$ tuned on the cover song tuning set. The module outputs $\delta_{x_1,x_2} = 1$ if both tracks are not detected as instrumentals, and 0 otherwise. For some rare cases where the SVR framework is truly performing poorly, it is also only outputting a few words. The instrumental detector then helps with additionally filtering some marginal cases where the lyrics transcription fails completely. We chose to keep this very simple as it performed sufficiently well for our purposes and allowed for improvements in future works.

### 3.4 Tonal-based cover detection

The tonal-based cover detection method selected is described in [6]. This system, called Re-MOVE [6], is an updated version of MOVE [5] and obtains the second most accurate benchmark on the Da-Tacos dataset [2]. Compared to the best one reported [20], it has the advantage of being publicly available [3]. The system is trained using the training part of Da-Tacos, as described in Section 4.1, and early stopping is performed using its validation component. For a given track, the system takes CremaPCP extracted from the audio as input and outputs a corresponding compact embedding. The CremaPCP feature is an intermediate representation of a chord estimation model. It is considered an efficient way to capture the tonal information of music and is shown to outperform more classic HPCP features for cover detection [2]. The similarity between the query and each track of the dataset is then the cosine similarity of their respective embeddings.

The Re-MOVE system uses a latent space reconfiguration technique on top of MOVE in order to reduce the embedding dimension (and then reduce memory requirements and retrieval time) while maintaining high detection performances. This technique reconfigures a pre-trained learned distance metric into a more compact embedding space with the same learned semantic relation.

### 3.5 Fusion

It has been shown in multiple domains that the fusion of different modalities can yield better performances than those obtained with each single modality [21]. For cover detection, fusing modalities, features or similarities matrix have already shown to improve results [22,23], notably using rank aggregation methods [24]. The fusion function chosen here is a weighted sum. It is more precisely described by:

$$s_{fus} = \begin{cases} \alpha s_{lyrics} + (1-\alpha) s_{tonal} & \text{if } \delta_{x_1,x_2} = 1 \\ s_{tonal} & \text{otherwise} \end{cases} \tag{1}$$

$\alpha$ is a simple scalar defined as an hyperparameter to tune. As the distributions of both similarities are very different, calibration before fusion was also tested. However, no improvement was shown on the cover song tuning set. Other fusion functions, such as linear regression or max function, did not lead to improvements in our simulations.

### 3.6 Scalability

Pairwise comparisons between a given query and all tracks in a dataset are linearly dependent on the size of the dataset

[1] https://www.music-ir.org/mirex/wiki/2020:MIREX2020_Results

[2] https://github.com/chitralekha18/AutoLyrixAlign

[3] https://github.com/furkanyesiler/re-move

without optimization, which cannot be considered scalable. In fact, a linear complexity for the queries involves a quadratic complexity for retrieving all musical works in the dataset, which can quickly become prohibitive for large collections. To achieve better scalability properties, most cover detection studies use ANN methods such as *Locality Sensitive Hashing* (LSH) [25, 26]. The idea behind ANN is that for a given query $x$ and a database $D$ the method outputs an approximation of the $k$ nearest neighbors of the query in the database with the complexity being sublinear in the size of the database. For a given query, in contrast with classic *K-Nearest Neighbors* (KNN), ANN methods are only browsing a subset of the complete search graph. All these methods are based on an index table allowing fast queries by outputting a "good" guess of the $k$ nearest neighbors of a given query, making it possible to recover the most highly classified covers in the ranked list obtained with all candidate points. The recall is used to quantify the quality of an ANN method by averaging percentages obtained, for various queries, of true k-nearest-neighbors from $k$ points returned by the method. In our case, we use the Hierarchical Navigable Small World Graph (HNSW) state-of-the-art ANN method; an extensive study of it is given in [27]. This algorithm gives logarithmic complexity for a query in terms of the size of the dataset. This method is directly applied on Re-MOVE and TFIDF embeddings, outputting for a given query $k$ nearest neighbors for each of them. Both sets of points are then concatenated and merged, obtaining a maximum of $2k$ points to consider for the fusion. Pairwise similarities between the query and these points are then generated using the Re-MOVE system and our lyrics-recognition pipeline.

# 4. EXPERIMENTAL EVALUATION

## 4.1 Dataset

Da-Tacos [2, 6] is the largest publicly available dataset for cover detection; the training set is composed of 83904 songs in 14999 cliques and the validation set of 14000 songs in 3500 cliques. A clique is defined as a cover group gathering multiple recordings of the same underlying "piece". The Da-Tacos benchmark test subset is a 15000 tracks dataset composed of 1000 cliques with 13 songs each and 2000 noise songs (i.e. that are in a single-song clique) that are not queried. To avoid overfitting, no clique overlaps with any set of Da-Tacos. Instrumentals represent around 20% of the dataset which motivates our choice of using an instrumental detection process. Currently, only a set of precomputed audio features are publicly available for the benchmarking subset test dataset. The dataset is mainly composed of English tracks and popular western music with a few non-English cliques. All hyperparameter tuning made during this paper is carried out on a subpart of the Da-Tacos validation that we choose to refer to as a *Da-Tacos tuning* set. We verified that no clique of this subset overlaped with any clique present in the dataset used to train the tonal-based cover detection system, i.e. the Da-Tacos training set. Also, a clique is discarded if it possesses one track present in the dataset used to train the SVR framework, i.e. DALI dataset. Detection of overlapping tracks and cliques is made using metadata, *i.e.* titles and artists names. *Da-Tacos tuning* is notably used to choose the string matching algorithm and the fusion function. We recover audio of 12862 tracks from the test dataset. 1849 are in single-song cliques and thus are not queried and only used as noise songs. We make sure no clique of this dataset overlaps with cliques in the Da-Tacos train and validation and that cliques possessing tracks existing in the DALI dataset are discarded. It will be simply referred to as *Da-Tacos test* for the rest of the document.

## 4.2 Fusion parameters

A track is classified as instrumental if the number of different words of its transcript is less than $l = 8$. This number is adjusted using *Da-Tacos tuning* as the value that maximizes the recall for the highest $F1$ score. Emphasis is put on the recall in order to avoid taking into account the lyrics-recognition based similarity for an instrumental track that has been misclassified as non-instrumental. An $\alpha$ value of 0.6 for fusing both system is tuned on *Da-Tacos tuning*.

## 4.3 Parameters of ANN

We use the HNSW implementation of the NMSLIB similarity search library [28]. For each query, we return the $k = 100$ nearest neighbors. This choice is derived from [4], which shows performance does not evolve significantly after the top-100 pruning. We use an approximated cosine similarity function to retrieve 100 candidates for each branch which results in, at most, 200 items for the fused model after concatenating and merging both sets.

## 4.4 Evaluation

The empirical evaluation of the cover detection task performances is given using the *Mean Average Precision* (MAP) [4]. For a query, *Average Precision* (AP) is quantifying the number of actual covers that are highly ranked. The AP score increases when actual covers are detected in the top ranks. The MAP is then simply obtained by averaging on the AP of all queries. As the MAP is not properly defined for systems that do not score every track (such as ANN), we report MAP@100 for these cases considering only the top-100 ranked item of each query. In any case, the MAP does not significantly evolve after the top-100 pruning as explained in the previous section.

# 5. RESULTS AND DISCUSSION

## 5.1 Lyrics-recognition based system results

### 5.1.1 Instrumental detection

Among the 12862 tracks in the test set, 3269 are detected as instrumentals. We compared this with the "Instrumental" tag available in the Da-tacos for all tracks. We obtain a

[4] Computed using the Metrics toolkit from https://github.com/benhamner/Metrics

| Query | System | MAP (%) |
|---|---|---|
| Da-Tacos-voice | Lyrics | **66.4 (0.4)** |
| | Tonal | 54.0 (0.4) |
| Da-Tacos-instr | Lyrics | 0.45 (0.06) |
| | Tonal | **47.8 (0.7)** |

**Table 1**. Results of lyrics-recognition based and tonal-based cover detection system on *Da-Tacos-voice*. *Da-Tacos-instr* is the subset of the *Da-Tacos test* restricted to instrumental tracks. Standard errors are given in parenthesis

| Dataset | System | MAP (%) |
|---|---|---|
| Da-Tacos test | Fused | **62.7 (0.3)** |
| | Fused-wo-inst | 50.2 (0.3) |
| | Tonal | 50.6 (0.3) |
| Da-Tacos-voice | Fused | **80.4 (0.3)** |

**Table 2**. Results of fused, with and without instrumental detection, tonal and lyrics-recognition based cover detection system on various datasets

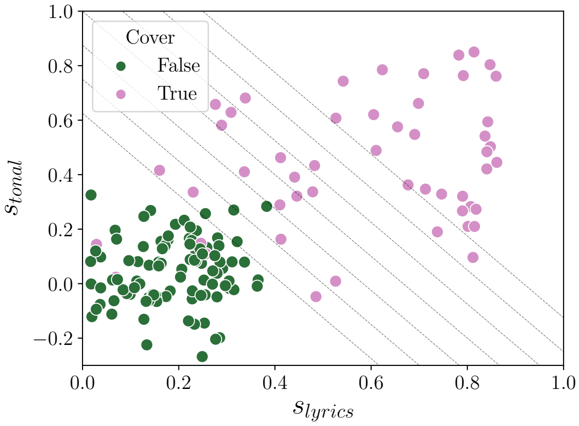

**Figure 2**. Similarities of sampled pairs of tracks from the *Da-Tacos-voice*. Here, each point is a pair of tracks. Each color indicates a same-clique belonging status. Some level curves of $s_{fus}$ are also displayed

precision of 82.86% for the instrumental detection, a recall of 96.68% and a F1 score of 89.24%. A closer look at misclassified tracks showed that there is some annotation noise in the Da-Tacos annotations which could artificially lower the previous metrics. As simple as it is, the instrumental detection performance seems suitable for our application. After filtering detected instrumentals, we obtain a subset of 9593 tracks that we label *Da-Tacos-voice*. 1582 tracks are in single-song cliques. 8011 tracks are then queried.

### *5.1.2 Lyrics-based cover detection*

We first evaluate our lyrics-recognition based system on the *Da-Tacos-voice*. Our results, displayed in Table 1, show that it is generally performing better than the tonal-based one in terms of MAP. They validate the assumption that lyrics can be considered as a strong invariant between covers. It also proves that the most recent state-of-the-art singing voice recognition framework produces transcriptions of sufficiently good quality to perform the cover song as a noisy text matching task. Looking empirically at results coming from both systems, most improvements of the lyrics-recognition system over the tonal-based system come, as expected, from covers with highly different tonal-content and lyrics being roughly the same.

We also query the tracks detected as instrumental and not from single-song cliques. Results are also displayed in Table 1. As expected, performance of the lyrics-recognition based system is almost close to zero. For the tonal-based system, results seem to degrade when compared to non-instrumentals tracks. This suggests that the system has either learned characteristics of the melody carried by the singing voice or implicitly estimated some of the lyrics information to perform cover detection.

### *5.1.3 The case of non-English tracks*

As stated in Section 3.1, our lyrics recognition framework cannot output non-English words, therefore non-English tracks may produce unexpected results. In order to assess the impact of this issue, we predicted a language label for every track of the *Da-Tacos-voice* using a language classifier [29] taking track metadata as input. Results show that the dataset is largely composed of English with more than 92.4% of the tracks being detected as English. Looking at tracks outside single-song cliques detected as non-English, half of them are false positives. We query non-English tracks of the resulting 44 cliques on the *Da-Tacos-voice*, representing 299 tracks. It is interesting to report that almost all cliques are homogeneous in terms of language. We obtain a MAP of 28% (2). Results show that even if performances deteriorate for these cases, our system is often able to correctly classify these tracks. It can be explained as the chosen singing voice framework is transcribing something similar from one cover to another even for non-English lyrics. Considering the small quantity of non-English tracks and results on these tracks, we consider that this issue has a limited impact on performance in our evaluation setup.

## 5.2 Fused system results

Results for the fused system on the full *Da-Tacos test* and its *Da-Tacos-voice* subset are given in Table 2. The fused system significantly outperforms the results of the tonal-based one alone showing the validity of our assumption of both systems being highly complementary. The use of the instrumental detection module to inform the fusion strategy is empirically validated, with a major drop of performances occurring when it is not considered. The gain in performance comes essentially for increased accuracy on the *Da-Tacos-voice* subset, where information from both

| System | SVR | MAP (%) |
|---|---|---|
| Lyrics | CTC | 40.3 (0.7) |
| | Our | 79.0 (0.6) |
| | Lyrics-informed | **89.7 (0.4)** |
| Fused | CTC | 71.1 (0.6) |
| | Our | 88.5 (0.4) |
| | Lyrics-informed | **93.6 (0.4)** |

**Table 3**. Performances of lyrics-recognition and fused based cover detection system on *Da-Tacos-lyrics* with various SVR framework. Lyrics-informed framework are informed by lyrics at test time

branches is available and the MAP reaches around 80%. To highlight this complementarity, similarities for sampled pairs of tracks from the *Da-Tacos-voice* are displayed in Figure 2. While the majority of same-clique pair lyrics and tonal similarities are significantly higher than non-matching pairs, there are multiple cases where one modality seems more indicative than the other. Level curves of $s_{fus}$ are also displayed illustrating most pairs being linearly separable in the combined modality plane.

### 5.3 ANN results

We first evaluate the impact of pruning results to the first 100 candidates by computing the MAP@100 of the fused system on the *Da-Tacos test* dataset. A small decrease is observed with a MAP@100 of 62.4% (0.3). After applying an ANN to our fused system, results remain the same with a MAP@100 of 62.4% (0.3) . This result can be explained as the recall of the HNSW for both a tonal-based and lyrics-recognition system being more than 99.5%. Thus, the scalability of our system is assured while maintaining the cover detection performances.

### 5.4 Impact of the SVR framework

A detailed analysis of failing samples of the lyrics-recognition based system shows that the main cause for failure is the low quality of the transcriptions. To further investigate this impact, we introduce two baselines by changing the SVR framework part of our system. In the first, an alternative *Connectionist Temporal Classification* (CTC) based SVR framework is used. The acoustic model of this framework is described in [30]. It consists of several *Bidirectional Long Short-Term Memory* (BiLSTM) layers, is trained on a multilingual subpart of the DALI dataset with a CTC algorithm and relies on a pre-processing step of singing voice separation. The language model used is the same as the one described in Section 3.1. Decoding is performed, after tuning the language model weight and insertion penalty value using a validation dataset, with a CTC beam search decoding tooklit[5] . The transcription of the results obtained on Jamendo dataset [19] are significantly lower than our current singing voice recognition framework with a WER of 84.4%. We thus expect this CTC-baseline to obtain results far below our system for cover detection tasks.

In the second baseline, we simulate an "ideal" SVR framework outputting an exact transcription. It can be considered as an oracle system, yielding an upper bound for performances of lyrics-recognition based systems. To compare these three systems, we retrieve the lyrics text information for part of the *Da-Tacos test*. The subset obtained is labeled *Da-Tacos-lyrics* and is composed of 3467 tracks for which we found matching lyrics. Considering that this subset only contains non-instrumental tracks, we discard the instrumental detector for this section. Again, tracks from single-song cliques are not queried and are used as noise songs.

The results obtained on *Da-Tacos-lyrics* are given in Table 3. These results confirm the intuition that the lyrics-recognition system's strength for covering detection task directly depends on the quality of the lyrics transcription. Ranking performances on *Da-Tacos-lyrics* for these systems are conserved after fusing them with the tonal-based branch. In comparison to the oracle system, our fused system shows excellent results even if there is still some room for improvement. With the transcription performances of our SVR framework being as low as 62% WER, it certainly indicates that a perfect transcription is not needed for the cover detection task. Interestingly, even an oracle system informed by the true lyrics benefits from being fused with a tonal-based one. This, once again, demonstrates both branches are acutely complementary to address the cover song detection problem. Future works will extend our system to take into account cases where lyrics are available for a part of the dataset.

## 6. CONCLUSION

Using only audio, we have proposed a framework that explicitly leverages two types of similarities, tonal and lyrics based, and reach high accuracy levels while remaining simple and scalable. With that said, work on more diverse data still remains to be done, notably on non-English tracks where performances seem to be limited.

Future work will include replacing the current monolingual lyrics recognition with a multilingual framework. A multilingual similarity, capable of detecting the similarity of two texts based on their semantic content, independently of their language, will also be defined and evaluated. More generally, the Da-Tacos dataset is quite biased towards popular western music. Additional experimentation on a wider range of genres, notably none western music, and cover types (e.g. karaoke, renditions, etc.) remains to be conducted. Finally, we will explore more elaborate fusion schemes, specifically, a mid-level fusion which can be further optimized and possibly lead to improved performance.

[5] `https://github.com/parlance/ctcdecode`

## 7. REFERENCES

[1] J. Serra, E. Gómez, and P. Herrera, "Audio cover song identification and similarity: background, approaches, evaluation, and beyond," in *Advances in Music Information Retrieval*. Springer, 2010, pp. 307–332.

[2] F. Yesiler, C. Tralie, A. A. Correya, D. F. Silva, P. Tovstogan, E. Gómez Gutiérrez, and X. Serra, "Da-tacos: A dataset for cover song identification and understanding," in *Int. Soc. for Music Information Retrieval (ISMIR)*, 2019.

[3] J. Serra, X. Serra, and R. G. Andrzejak, "Cross recurrence quantification for cover song identification," *New Journal of Physics*, vol. 11, no. 9, 2009.

[4] A. A. Correya, R. Hennequin, and M. Arcos, "Large-scale cover song detection in digital music libraries using metadata, lyrics and audio features," *arXiv:1808.10351*, 2018.

[5] F. Yesiler, J. Serra, and E. Gómez, "Accurate and scalable version identification using musically-motivated embeddings," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 21–25.

[6] F. Yesiler, J. Serra, and E. Gomez, "Less is more: Faster and better music version identification with embedding distillation," in *Int. Soc. for Music Information Retrieval (ISMIR)*, 2020.

[7] C.-C. Wang and J.-S. R. Jang, "Improving query-by-singing/humming by combining melody and lyric information," *IEEE/ACM Trans. on Audio, Speech, and Language Processing (TASLP)*, vol. 23, no. 4, pp. 798–806, 2015.

[8] D. P. Ellis and G. E. Poliner, "Identifying cover songs' with chroma features and dynamic programming beat tracking," in *IEEE Int. Conf. on Acoustics, Speech and Signal (ICASSP)*, vol. 4, 2007.

[9] J. Serra, E. Gómez, and P. Herrera, "Transposing chroma representations to a common key," in *IEEE CS Conf. on The Use of Symbols to Represent Music and Multimedia Objects*, 2008, pp. 45–48.

[10] T. Bertin-Mahieux and D. P. Ellis, "Large-scale cover song recognition using hashed chroma landmarks," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2011, pp. 117–120.

[11] J. Lee, S. Chang, S. K. Choe, and K. Lee, "Cover song identification using song-to-song cross-similarity matrix with convolutional neural network," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 396–400.

[12] J. Serra, E. Gómez, P. Herrera, and X. Serra, "Chroma binary similarity and local alignment applied to cover song identification," *IEEE Trans. on Audio, Speech, and Language Processing (TASLP)*, vol. 16, no. 6, pp. 1138–1151, 2008.

[13] D. P. Ellis and B.-M. Thierry, "Large-scale cover song recognition using the 2d fourier transform magnitude," in *Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2012.

[14] J. B. Smith, M. Hamasaki, and M. Goto, "Classifying derivative works with search, text, audio and video features," in *IEEE Int. Conf. on Multimedia and Expo (ICME)*, 2017, pp. 1422–1427.

[15] T. J. Tsai, T. Prätzlich, and M. Müller, "Known artist live song id: A hashprint approach," in *Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2016, pp. 427–433.

[16] C. Gupta, E. Yılmaz, and H. Li, "Automatic lyrics transcription in polyphonic music: Does background music help?" in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[17] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, "Creating dali, a large dataset of synchronized audio, lyrics, and notes," *Trans. of the Int. Soc. for Music Information Retrieval (TISMIR)*, vol. 3, no. 1, 2020.

[18] K. Heafield, "KenLM: Faster and smaller language model queries," in *Workshop on Statistical Machine Translation (SMT)*, 2011.

[19] D. Stoller, S. Durand, and S. Ewert, "End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 181–185.

[20] X. Du, Z. Yu, B. Zhu, X. Chen, and Z. Ma, "Bytecover: Cover song identification via multi-loss training," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[21] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[22] N. Chen, W. Li, and H. Xiao, "Fusing similarity functions for cover song identification," *Multimedia Tools and Applications*, vol. 77, no. 2, pp. 2629–2652, 2018.

[23] C. J. Tralie, "Early mfcc and hpcp fusion for robust cover song identification," in *Int. Soc. for Music Information Retrieval (ISMIR)*, 2017.

[24] J. Osmalsky, J.-J. Embrechts, P. Foster, and S. Dixon, "Combining features for cover song identification," in *Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2015.

[25] M. Khadkevich and M. Omologo, "Large-scale cover song identification using chord profiles," in *Int. Soc. for Music Information Retrieval (ISMIR)*, vol. 13, 2013, pp. 233–238.

[26] P. Grosche and M. Müller, "Toward characteristic audio shingles for efficient cross-version music retrieval," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 473–476.

[27] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 4, pp. 824–836, 2018.

[28] L. Boytsov and B. Naidan, "Engineering efficient and effective non-metric space library," in *Similarity Search and Applications (SISAP)*, 2013.

[29] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv:1607.01759*, 2016.

[30] A. Vaglio, R. Hennequin, M. Moussallam, G. Richard, and F. D'alché-Buc, "Multilingual lyrics-to-audio alignment," in *Int. Soc. for Music Information Retrieval Conference (ISMIR)*, 2020.