AND WHAT IF TWO MUSICAL VERSIONS DON'T SHARE MELODY, HARMONY, RHYTHM, OR LYRICS ?

Mathilde Abrassart
Ircam Amplify
abrassart@ircam.fr

Guillaume Doras
Ircam, Sorbonne Université, CNRS, STMS Lab
doras@ircam.fr

ABSTRACT

Version identification (VI) has seen substantial progress over the past few years. On the one hand, the introduction of the metric learning paradigm has favored the emergence of scalable yet accurate VI systems. On the other hand, using features focusing on specific aspects of musical pieces, such as melody, harmony, or lyrics, yielded interpretable and promising performances. In this work, we build upon these recent advances and propose a metric learning-based system systematically leveraging four dimensions commonly admitted to convey musical similarity between versions: melodic line, harmonic structure, rhythmic patterns, and lyrics. We describe our deliberately simple model architecture, and we show in particular that an approximated representation of the lyrics is an efficient proxy to discriminate between versions and non-versions. We then describe how these features complement each other and yield new state-of-the-art performances on two publicly available datasets. We finally suggest that a VI system using a combination of melodic, harmonic, rhythmic and lyrics features could theoretically reach the optimal performances obtainable on these datasets.

1. INTRODUCTION

The version identification (VI) problem has received much attention over the last two decades. Pioneering works showed promising accuracy on small audio datasets, but remained difficult to scale to larger modern audio corpora. The recent introduction of data-driven approaches based on neural networks led to significant progress towards accurate yet scalable VI systems [1].

Different paradigms are currently active: one approach considers VI as a classification task, and intends to classify versions into the same class [2], while another approach formulates VI as a metric learning problem, and intends to minimize (resp. maximize) a distance between versions (resp. non-versions) [3, 4]. Recent works have also proposed a combination of both [5]. Metric learning or classification approaches seem to yield similar performances, as

it has been observed for other MIR applications [6]. These systems also differ according to a perhaps more important aspect: their input feature. Some use a generic audio representation, such as the Constant-Q transform [7], and rely on the expressivity of the network to disentangle relevant musical features. Others use specialized features, such as the melodic line, the harmonic structure and/or the lyrics, and rely on input data to discriminate between versions and non-versions [3, 4, 8, 9].

In this work, we pursue in the direction of a metric-based approach using specialized features. We build upon the system described in [8], conserving its principle and architecture, and explore the use of new input features. The reason is motivated by three practical considerations: firstly, the metric learning approach yields a very compact representation of audio (its embedding), which can be conveniently stored, indexed and queried in very large databases. Secondly, the embedding space and the musical similarity measure that is obtained for each different specialized features is meaningful from a musical perspective, and can be reused for other purposes, e.g. playlist generation. Thirdly, the use of specialized features requires smaller models that are faster to train and less energy consuming than larger architectures [10].

It can reasonably be assumed that different versions of the same musical work share at least one of these four features: the melodic line, the harmonic structure, the rhythmic patterns and sometimes the lyrics. The role of melody and harmony in version similarity has been thoroughly investigated [8]. The role of the lyrics has also been studied recently, albeit not from a metric learning perspective [9].

In this work, we present a systematic study of the contribution of these four features to version similarity, and describe a metric learning-based system combining all of them. We show that this combination provides new state-of-the-art performances on two publicly available datasets. We also show that an oracle using these feature embeddings nearly achieves the maximum theoretical performances on these datasets, suggesting that design of future VI systems reaching these performances may be possible.

The rest of this paper is organized as follows: we briefly review the previous studies inspiring our current work (Section 2). We then describe how we extracted rhythmic and lyrics features, and our metric learning-based architecture (Section 3). We present our experiments, discuss our results (Section 4), and illustrate them with some examples (Section 5). We conclude this paper with our future work.

2. RELATED WORK

In this section, we present a brief overview of the main concepts that inspired our present work.

2.1 Metric learning

Learning a similarity metric that generalizes to unseen examples is a common objective in machine learning. The goal is to learn how to map the data of interest into a compact representation (its embedding), and to minimize (resp. maximize) the distance between the embeddings of similar (resp. dissimilar) examples. Various MIR applications rely on a concept of musical similarity, e.g. music classification [11], music recommendation [12], or VI systems [1], among many others. Musical similarity between two tracks is typically evaluated first deriving an intermediate feature representation from the audio waveform and then computing a distance between feature pairs.

In the past few years, metric learning has proven its efficiency to build scalable yet accurate VI systems. These modern architectures typically rely on a CNN-based model trained with a triplet loss [13] to embed the musical information contained in the input feature into a single vector embedding that can be rapidly compared via Euclidean distance computation. However, these different systems have made different choices regarding their input features: for instance, Doras et al. [3] used a melodic line representation, Yesiler et al. [4] used a harmonic structure representation, and Du et al. [5] used a more generic CQT. The choice between specialized or generic features seems to have a non-negligible impact on the required size of the models (the former uses a 5-layers CNN, while the latter uses a ResNet50).

In this work, we choose the first alternative, and we propose to explore other specialized features beside melody and harmony, in particular the rhythmic and lyrics features.

2.2 Rhythm patterns detection

Musical similarity based on rhythmic patterns has long been investigated in MIR research, e.g. for audio retrieval [14] or music classification [15]. With the purpose of analysis of musical style and recognition of musical genres, Pampalk et al. introduced the fluctuation patterns (FP), representing rhythmic patterns in different frequency bands, and their evolution over time [16]. The basic assumption is that similar songs exhibit similar characteristic rhythmic patterns, and that comparing FP between tracks shall give enough information about their similarity or dissimilarity from a genre or mood perspective. We propose here to extend this idea to VI context, and to use the fluctuation patterns to discriminate versions from non-versions.

In practice, the FP is obtained computing the audio Mel spectrogram, summing up the high Mel bands to highlight the low frequencies and performing a second STFT in each mel band along the time axis. This results in a 3-dimensional matrix with axes corresponding to the the Mel bands, the frequency modulation and the time. The frequency modulation axis therefore represents the periodicity of the loudness in the corresponding Mel band: for example, a drum kick playing at 120 bpm will be represented here with a frequency modulation at 2 Hz in the low frequency bands. Finally, a perceptual filter is applied on each frequency modulation band. This filter is supposed to highlight the frequency modulations most perceived by the human ear ; for example, a frequency modulation at 4 Hz gives a more intense feeling of fluctuation strength. The 3-dimensional matrix is then averaged along the frequency axis, resulting in a representation of the variations of the frequency modulation over time.

One of the FP limitation is the use of a linear scale to represent periodicities. This was addressed by Pohle et al., who used a log scale to represent frequency modulations [17]. The advantage is that the same onset structure played at different tempi will have all its activations shifted by the same amount along the frequency modulation axis. Another way to achieve this tempo invariance is to compute periodicities with a Constant-Q transform (CQT) instead of a STFT [18].

2.3 Lyrics recognition

In automatic speech recognition (ASR), the traditional approach relies on a language model and an acoustic model, typically implemented as a Hidden Markov Model, possibly coupled with a neural network [19]. An alternative approach consists in implementing both language and acoustic models as a single neural network, trained in an end-to-end fashion with a Connectionist Temporal Classification (CTC) loss [20]. CTC-based models outputs the probability distribution of symbols at each time frame, which can be decoded into the most likely sequence of symbols via classical beam search. This approach has become very popular since fully convolutional end-to-end architectures have achieved performances comparable to those of the hybrid architectures [21].

Although singing voice has many obvious differences with speech, automatic lyrics recognition (ALR) or alignment (ALA) systems are usually directly inspired by ASR applications. Moreover, the recent introduction of large lyrics annotated audio datasets, such as Dali [22], has fostered the development of new ALR systems, whether they are based on the traditional [23] or end-to-end [24] architectures. While the former seems to yield better results [25], the latter has the advantage of its simplicity, both at training and inference time.

However, very few attempts have been made to use lyrics to assess audio track similarity. It was proposed to improve a query-by-humming system [26], but to the best of our knowlege, only Vaglio et al. proposed the use of lyrics for version identification [9]. They used an existing ALR system to extract lyrics from the audio, and estimate track similarity via a string matching algorithm. However, the comparison cost of such algorithm quickly becomes prohibitive when querying large modern corpora, and limits the scalability of this approach.
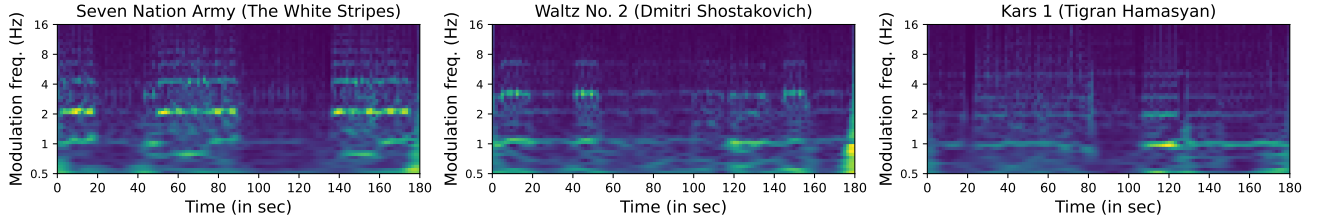
Figure 1: Examples of the Constant Q Fluctuation Patterns (CQFP).

3. PROPOSED METHOD

In this section, we describe and motivate our design choices. We first present how we extracted our rhythmic and lyrics features, and publicly release our datasets [1] . We then present our metric learning-based VI model.

3.1 Rhythmic features

Our assumption is that the FP representation described in Section 2.2 displays both the local rhythmic patterns along the frequency modulation axis, as well the global rhythmic evolution of the piece over time. We therefore propose to use this representation as our rhythmic feature for version identification. However, we introduce a CQT to compute the periodicities, and to achieve tempo invariance along the frequency modulation dimension. We choose as minimum frequency for the CQT 0.5 Hz (i.e. 30 bpm which we assume would be the tempo of the slowest tracks), and to cover up to 5 octaves i.e. 16Hz (960 bpm) with 10 bins per octave to get all the rhythmic subtleties. We kept the same other parameters as in the original implementation [16].

As an illustration, Figure 1 represents the CQ-FP obtained for different tracks with characteristic time signature. The first example has a 4/4 time signature, and clearly displays a rhythmic pattern present around 2 Hz (~120 bpm), as well as another periodicity around 4 Hz, which corresponds to a clear binary rhythm. The second example has a 3/4 time signature, and the rhythm of a waltz appears clearly: one beat at 1 Hz (60 bpm) and another at about 3 Hz. Finally, the third example has a 5/4 time signature (irregular), and we find this characteristic by observing bands at 1, 2 and 3 Hz. It can also be observed on each example that our rhythmic feature also represents the global structure of the piece over time, which is probably another relevant aspect in a VI context. Finally, as the modulation frequency dimension has a constant Q-factor, a change in tempo would not change the spacing between activations.

3.2 Lyrics features

We argue that accurate lyrics recognition is not required for version identification, and that identifying only a few common words, or even a few common character sequences, between tracks shall be sufficient to determine whether they are versions or not. We thus implemented a deliberately simple fully convolutional ALR system inspired by recent ASR system [21, 27].

---
[1] https://ircam-anasynth.github.io/papers/2022/abrassart

Model It is an 8-layers 2D-CNN with 3x3 kernels. Max-pooling 2x2 is applied on the first 2 layers to decrease time and frequency dimensions, and max-pooling 1x2 is applied on the next 6 layers only for frequency dimension. The first layer has 64 filters, doubled at each layer up to 512. A dropout with rate 0.3 is applied.

Inputs/outputs We use no pre-processing on the audio (no data augmentation, no voice separation). As classically done in ASR, we use 40 band Mel-spectrogram with 10ms timeframes as input feature. The model outputs a posteriorgram corresponding to the log-probabilities of the 'a...z' letters, the space symbol and the CTC blank symbol, i.e. 28 bins in total. An output example is shown Figure 2.
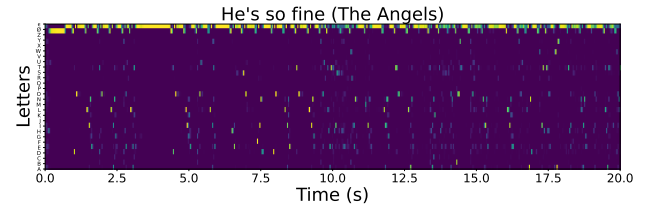
Figure 2: A posteriorgram obtained for 20 sec. of audio.

Training and inference The model is trained on the Dali dataset, which provide 12k+ polyphonic audio with lyrics annotations at the word level. We used audio chunks of 10 seconds, and used a CTC loss with Adam optimizer to train the model to align between audio and text. We evaluated its performances on a distinct Dali test subset using a greedy beam search decoding with no language model, and achieved a modest Character Error Rate (CER) of 0.496.

3.3 Convolutional architecture

The same simple architecture (with different configurations) has shown its ability to capture relevant melodic or harmonic similarity between versions [3,4]. It consists in a plain 5-layers Convolutional Neural Network (CNN), encoding each input feature using time and frequency max-pooling, while increasing the number of filters at each layer. The CNN output feeds a gated temporal attention mechanism [28], which was found to help the model to focus on relevant portions of the input features. We refer the reader to our previous work [8] for implementation details.

In this work, we keep the exact same generic architecture, and propose two new configurations to process also the rhythmic and lyrics features. We summarize the configuration used for each of the four features in Table 1.

Features	Melodic [3]						Harmonic [4]						Rhythmic						Lyrics					
Layers	n	k	ks	p	ps	d	n	k	ks	p	ps	d	n	k	ks	p	ps	d	n	k	ks	p	ps	d
1	64	3x3	1x1	2x2	2x2	0.0	256	180x12	1x1	1x12	1x1	0.0	64	3x20	1x1	1x2	1x2	0.4	64	10	1	5	2	0.3
2	128	3x3	1x1	2x2	2x2	0.1	256	5x1	1x1	1x1	1x1	0.0	128	3x3	1x1	1x2	1x2	0.3	128	10	1	5	2	0.2
3	256	3x3	1x1	2x2	2x2	0.1	256	5x1	1x1	1x1	1x1*	0.0	256	3x3	1x1	1x2	1x2	0.2	256	10	1	5	2	0.1
4	512	3x3	1x1	2x2	2x2	0.2	512	5x1	1x1	1x1	1x1	0.0	512	3x3	1x1	1x2	1x2	0.1	512	10	1	5	2	0.1
5	1024	3x3	1x1	2x2	2x2	0.3	512	5x1	1x1	1x1	1x1*	0.0	1024	3x3	1x1	1x2	1x2	0.0	1024	10	1	5	2	0.0

Table 1: Configuration of the 5-layers CNN used for the features. n : number of filters, k : kernel size, ks : kernel stride, p : pool size, ps : pool stride, d : dropout. 1st dimension is time, 2d dimension is frequency. Convolutions are "same" for Me, Rh and Ly, and "valid" for Ha. *Ha uses dilation rate of 20 and 13 along time dimension on 3rd and 5th layers respectively.

Rhythmic features Given the tempo invariance on the periodicity dimension explained in Section 3.1, the first layer has a 20 bins kernel on this axis to capture all patterns within 2 octaves (for instance quarter, eight and sixteenth notes). All other layers have 3x3 kernel with max-pooling of size 2 on the periodicity axis.

Lyrics features We conducted various experiments to find the best kernel size to apply to the lyrics. It appeared that a rather short receptive field of 10 bins yield the best results. We kept it for all layers, with a mean-pooling of size 5.

Finally, a dense layer applied after the temporal attention block outputs a 512 bins embeddings, L2-normalized so that each track becomes a point on the surface of the unit hypersphere, bounding the distance between 2 points within the [0,2] interval.

3.4 Embedding concatenation

In this work, we investigated only a late embedding fusion scheme, which simply consists in concatenating each feature embedding, and to L2-normalize the concatenated result. It is straightforward to show that the distance between a pair of normalized concatenation of n feature embeddings is the quadratic mean of the n distances between each feature embedding pair. All feature combination scores in Section 4 have been obtained using this method.

The practical advantage of this simple concatenated embedding is twofold: it remains easy to store and to query in an large index, and the lookup can be done for some specific feature combinations only (zero masking the unwanted feature embeddings).

4. EXPERIMENT AND RESULTS

In this section, we present our experimental setup and results. For brevity, we will denote melodic, harmonic, rhythmic and lyrics features by their abbreviations Me, Ha, Rh, and Ly, respectively.

4.1 Experimental setup

We use the exact same protocol as in our previous work [8].
Training We trained our four models on the publicly available dataset SHS$_{5+}$ [1] , which contains ~62k covers of ~7.5k works. We used the provided features for Me and Ha, and extracted Rh and Ly from the audio, as described in sections 3.1 and 3.2. We used a semi-hard triplet loss, using an Adam optimizer, an initial learning rate of $1e^{-4}$ and a batch size of 64. All other details are replicated from [8].

Test We tested our four models on SHS$_{4-}$ [1] , containing ~50k covers of ~20k works. We also retrained models on SHS$_{5+/4-}$ and tested on Da-Tacos [2] , containing 13k covers of 1k works and 2k confusing tracks [29]. As some samples overlap between Da-Tacos, SHS$_{4-}$ and Dali, we made sure that none of these samples were used for scoring the models, as done in [5].

For each feature, we used the corresponding trained model to compute each track embedding, and computed their pairwise distance matrix. For feature combinations, distance matrix is computed using the quadratic mean of each feature distance matrix, as described in Section 3.4.

4.2 Results

We summarize in Table 2 the performances obtained on SHS$_{4-}$ and Da-Tacos by our models, reporting the metrics classically used for VI: Mean Average Precison (MAP), mean number of correct answers in the first 10 (MT@10) and mean rank of first correct answer (MR1).

Train set	SHS$_{5+}$			Pruned SHS$_{5+/4-}$		
Test set	Pruned SHS$_{4-}$			Pruned Da-Tacos		
Input feature	MAP	MT@10	MR1	MAP	MT@10	MR1
Me	0.427	0.822	1131	0.363	4.064	97
Ha	0.538	1.003	982	**0.488**	**5.256**	**63**
Rh	0.099	0.231	2921	0.055	0.689	244
Ly	**0.672**	**1.190**	**968**	0.393	4.596	199
Me+Ha	0.693	1.256	453	**0.626**	**6.668**	**32**
Me+Ha+Ly	**0.800**	**1.396**	291	0.602	6.480	33
Me+Ha+Rh	0.688	1.250	413	0.557	5.994	33
Me+Ha+Rh+Ly	0.785	1.378	**286**	0.560	6.054	33
Me+Ha (O)	0.879	1.521	97	0.837	8.709	4
Me+Ha+Rh (O)	0.939	1.607	21	0.905	9.303	**1**
Me+Ha+Ly (O)	**0.963**	**1.637**	14	0.918	9.398	**1**
Me+Ha+Rh+Ly (O)	**0.978**	**1.658**	**4**	**0.951**	**9.657**	**1**

Table 2: Performance metrics obtained on SHS$_{4-}$ and Da-Tacos for input features and their combinations. O=oracle

We first examine the performances of each single feature. Me and Ha results are in line with our previous work [8].
Rhythmic features The performances obtained using Rh are clearly lower, which suggests that our rhythm feature is not providing relevant VI information, or that the rhythm patterns themselves are not specific enough to discriminate between versions and non-versions. As a consequence, adding Rh degrades our Me+Ha baseline. This is not entirely surprising, as many non-versions might exhibit similar rhythmic patterns (we will however see in Section 4.3 and Section 5 that Rh can be relevant in some cases).

Lyrics features In contrast, the use of lyrics clearly outperforms the other features on SHS_{4-}. This confirms that versions often share the same lyrics, and that even a very inaccurate ALR system can be beneficial to VI. The combination Me+Ha+Ly improves by more than 10% the Me+Ha performances, which will be explained in Section 5.1.

On Da-Tacos, we observe much less clear-cut results. As already noticed by Vaglio et al. [30], Da-Tacos has about 20% of instrumental songs. A closer look to our results shows that these songs typically produce false positives. Following Vaglio et al., we pruned the instrumental songs from Da-Tacos, and recomputed the results for Ly, obtaining MAP=0.674, MT@10=5.931 and MR1=59, which is consistent with the values obtained for SHS_{4-}.

4.3 Oracle results

We also present in Table 2 the results obtained by an oracle. This oracle only considers the best performing feature to compute each pairwise distance: for each pair of tracks, it only uses the feature embeddings yielding the lowest (resp. highest) distance for versions (resp. non-versions).

Interestingly, it appears that the performances of an oracle using three or four features approaches the theoretical optimal values on our two datasets. This is particularly clear with the Me+Ha+Ly and Me+Ha+Rh+Ly combinations: the MAP tends to 1.0, i.e. most versions have been ranked in the first answers for each query. The MR1 also tends to 1, i.e. first answer is correct for most queries. The MT@10 also tends to the theoretical optimal values (in SHS_{4-}, each track has 2, 3 or 4 versions, and its optimal MT@10=1.695, while in Da-Tacos, each track has 0 or 12 versions, and its optimal MT@10=10).

The Figure 3 displays the contribution of each feature to the oracle score, i.e. which feature is the most relevant to determine if two tracks are versions ("Positive pairs") or non-versions ("Negative pairs").
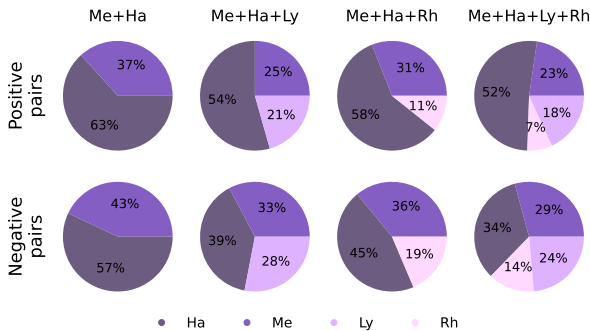
Figure 3: Most relevant feature proportions to identify positive and negative pairs on SHS_{4-}.

It appears that Ha is usually the most efficient feature to discriminate between versions or non-versions. However, both Me and Ly contribute importantly to identification (e.g. resp. 25% and 21% of the positive pairs, 33% and 28% of the negative pairs in the Me+Ha+Ly combination). Finally, and even though Rh alone yields poor results, its contribution is not negligible when combined with other features (e.g. 14% of the negative pairs in the Me+Ha+Rh+Ly combination).

4.4 Comparison with state-of-the-art

Performances obtained by recent VI systems are summarized in Table 3 (second column indicates the embedding size used by each system).

Our system using Me+Ha+Ly improves the state-of-the-art on SHS_{4-} and on Da-Tacos (without instrumentals).

| Test set		SHS₄₋			Da-Tacos		
Model	Emb.	MAP	MT@10	MR1	MAP	MT@10	MR1
Doras et al. [8]	512	0.660	1.080	657	0.635	6.744	30
Vaglio et al. [30]	n/a	n/a	n/a	n/a	0.804*	n/a	n/a
Du et al. [5]	1536	n/a	n/a	n/a	**0.791**	n/a	**19.2**
Me+Ha+Ly (ours)	1536	**0.800**	**1.396**	**291**	**0.818***	7.205*	16*
					0.602	6.480	33

Table 3: Sota comparison on SHS_{4-} and Da-Tacos. *results obtained on Da-Tacos-Vocals (w/o instrumental tracks).

5. QUALITATIVE ANALYSIS

In this section, we illustrate qualitatively the previous quantitative results. We encourage readers to listen to the audio samples available on the paper companion website [1] as part of reading the paper.

5.1 Ly vs. Me+Ha examples

We intend here to illustrate how Ly improves the Me+Ha system. Figure 4 plots the distance obtained for Me+Ha and Ly between randomly sampled positive (green) and negative (red) pairs.
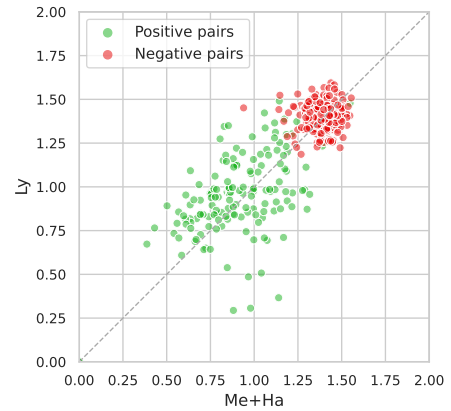

Figure 4: Pairwise distances for Me+Ha vs. Ly (500 pairs from SHS_{4-}).

This plot confirms that Me+Ha and Ly are complementary, as already seen in Section 4.3. The dots situated away from the diagonal correspond to track pairs scored correctly by Me+Ha and incorrectly by Ly (or vice-versa). As certain features yield better results for certain songs, their combination will statistically improve the results, as we will now illustrate with some contrasted examples.

Ly > Me+Ha There are many versions whose musical style, melody and harmony differ greatly from the original, and where only the lyrics can help to identify them. This is illustrated on Figure 5(a), which shows that the Jimmy Noone's and John Fogerty's versions of "You Rascal You" are very different musically while the lyrics exhibit enough similarity to be correctly identified.
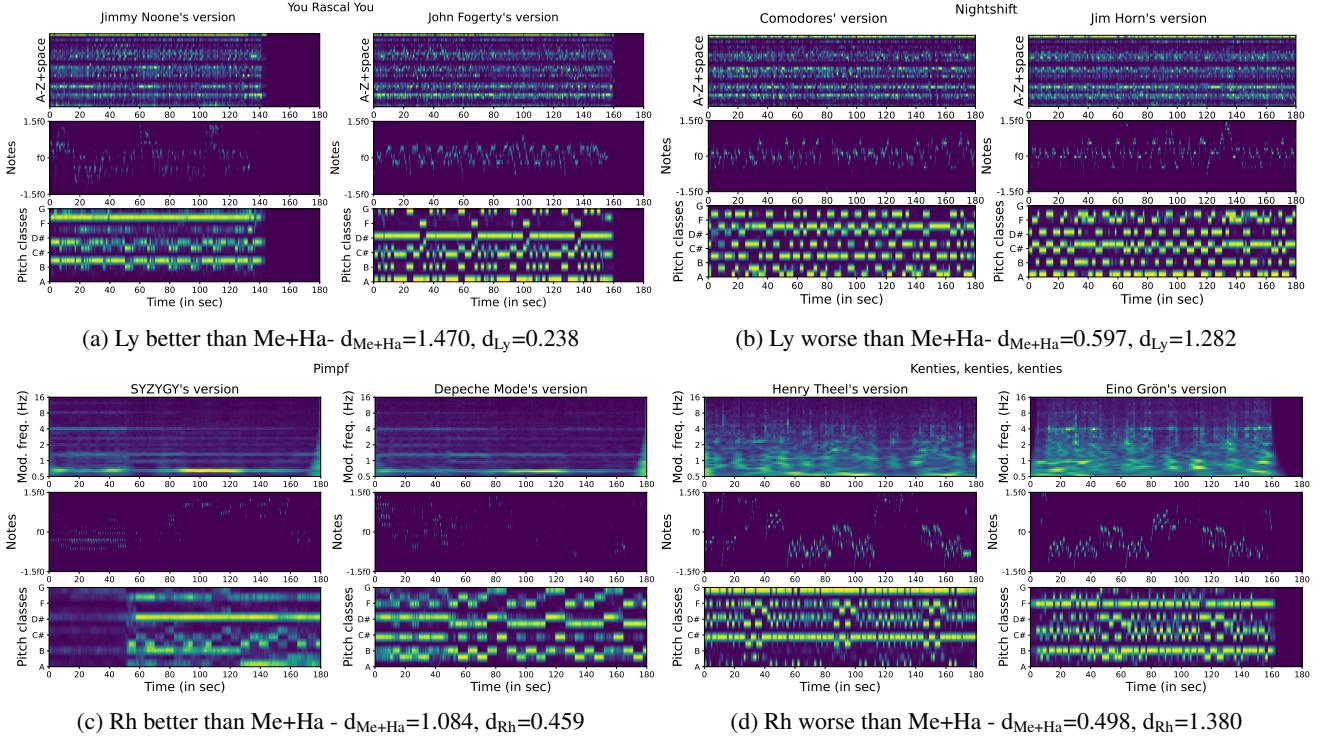
Figure 5: Examples for rhythm and lyrics : Rh > Me+Ha 5(c), Rh < Me+Ha 5(d), Ly > Me+Ha 5(a) and Ly < Me+Ha 5(b)

It also appears that our approximated ALR system is efficient for different languages. For instance, the versions of Asta Kask and of The Hep Stars of the song "I natt jag drömde", are very different in melody and harmony (d_{Me+Ha}=1.448) while the lyrics remain similar (d_{Ly}=0.342), despite the fact that the lyrics are in Swedish.
Ly < Me+Ha As already mentioned, using Ly will yield wrong results in presence of instrumental version (i.e. no lyrics). In the example shown in Figure 5(b), the version of "Nightshift" by the Commodores has lyrics while the one of Jim Horn's does not. We noticed that the system sometimes considers lead instruments as voices. However, this Ly false negative is correctly caught by the Me+Ha.

5.2 Rh vs. Me+Ha examples

Although less obvious than for Ly, combining Rh and Me+Ha also appears to be complementary in some cases.
Rh > Me+Ha Even though Rh yields poor performances in general, there are cases where it is the only feature available to identify versions. This is illustrated on Figure 5(c), which shows the Rh, Me and Ha features for two versions of "Pimpf". In this song, the melody is almost non-existent, and the harmony is very different between both versions. Only a few bass notes in the middle are salient enough to identify the song, and this short bassline appears similarly on the two FP features.

Rh might also be a good discriminating feature in other cases, e.g. for live concert versions. One version of "Mama's Little Baby" is recorded in studio while the other is a concert filmed from the audience. The Me+Ha distance between these versions is high (d_{Me+Ha}=1.329) because of the bad live recording quality. But, the drums are distinguishable enough to find similarity (d_{Rh}=0.714).

Rh < Me+Ha But Rh often yields wrong results. This is illustrated on Figure 5(d), which shows the features of two versions of "Kenties kenties kenties". Although melody and harmony are similar, the rhythm is very different, and the use of Rh produces a false negative.

6. CONCLUSION

It was shown previously that VI systems combining melody and harmony yields promising performances. In this paper, we proposed to consider also rhythmic and lyrics features to improve these results further. We showed that an existing rhythmic feature commonly used for genre classification is only helpful in a few cases, such as live version identification. But we also showed that an approximate lyrics representation can improve the performances of existing melody and harmony-based systems. We explained these results by the fact that detecting correctly only a few character sequences appears to be enough to distinguish versions and non-versions. We showed that our system combining these features establishes new state-of-the-art on two public datasets. More importantly, we indicated that these feature combinations provide enough information to approach the theoretical optimal performances obtainable on these datasets.

In our future work, we will investigate a more elaborated fusion scheme in order to train our model to behave as an oracle: our objective is to teach the system how to choose between available features to pick only the most relevant one for each pair of tracks. This might answer the question of whether the concept of musical version can be reduced to its melodic, harmonic, rhythmic, and lyrics dimensions.

7. ACKNOWLEGMENTS

8. REFERENCES

[1] F. Yesiler, G. Doras, R. M. Bittner, C. J. Tralie, and J. Serrà, "Audio-based musical version identification: Elements and challenges," *IEEE Signal Processing Magazine*, vol. 38, no. 6, 2021.

[2] Z. Yu, X. Xu, X. Chen, and D. Yang, "Learning a representation for cover song identification using convolutional neural network," *Proceedings of ICASSP (International Conference on Acoustics, Speech and Signal Processing)*, 2020.

[3] G. Doras and G. Peeters, "Cover detection using dominant melody embeddings," in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2019.

[4] F. Yesiler, J. Serrà, and E. Gómez, "Accurate and scalable version identification using musically-motivated embeddings," *Proceedings of ICASSP (International Conference on Acoustics, Speech and Signal Processing)*, 2020.

[5] X. Du, K. Chen, Z. Wang, B. Zhu, and Z. Ma, "Bytecover2: Towards dimensionality reduction of latent embedding for efficient cover song identification," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 616–620.

[6] J. Lee, N. J. Bryan, J. Salamon, Z. Jin, and J. Nam, "Disentangled multidimensional metric learning for music similarity," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020.

[7] Z. Yu, X. Xu, X. Chen, and D. Yang, "Temporal pyramid pooling convolutional neural network for cover song identification," *Proceedings of the International Joint Conference on Artificial Intelligence*, 2019.

[8] G. Doras, F. Yesiler, J. Serrà, E. Gomez, and G. Peeters, "Combining musical features for cover detection," in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2020.

[9] A. Vaglio, R. Hennequin, M. Moussallam, and G. Richard, "The words remain the same - cover detection with lyrics transcription," in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2021.

[10] C. Douwes, P. Esling, and J.-P. Briot, "Energy consumption of deep generative audio models," *arXiv preprint arXiv:2107.02621*, 2021.

[11] M. Slaney, K. Weinberger, and W. White, "Learning a metric for music similarity," in *International Symposium on Music Information Retrieval (ISMIR)*, vol. 148, 2008.

[12] B. McFee, L. Barrington, and G. Lanckriet, "Learning content similarity for music recommendation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 8, pp. 2207–2218, 2012.

[13] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of IEEE CVPR (Conference on Computer Vision and Pattern Recognition)*, 2015.

[14] J. Foote, M. Cooper, and U. Nam, "Audio retrieval by rhythmic similarity." in *ISMIR*. Citeseer, 2002.

[15] S. Dixon, F. Gouyon, G. Widmer *et al.*, "Towards characterisation of music via rhythmic patterns." in *ISMIR*, 2004.

[16] E. Pampalk, "Computational models of music similarity and their application in music information retrieval," Ph.D. dissertation, 2006.

[17] T. Pohle, D. Schnitzer, M. Schedl, P. Knees, and G. Widmer, "On rhythm and general music similarity." in *ISMIR*, 2009, pp. 525–530.

[18] H. Foroughmand and G. Peeters, "Deep-rhythm for tempo estimation and rhythm pattern recognition," in *International Society for Music Information Retrieval (ISMIR)*, 2019.

[19] E. Trentin and M. Gori, "A survey of hybrid ann/hmm models for automatic speech recognition," *Neurocomputing*, vol. 37, no. 1-4, pp. 91–126, 2001.

[20] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[21] R. Collobert, C. Puhrsch, and G. Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:1609.03193*, 2016.

[22] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, "Dali: a large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm." in *19th International Society for Music Information Retrieval Conference*, ISMIR, Ed., 2018.

[23] C. Gupta, E. Yılmaz, and H. Li, "Acoustic modeling for automatic lyrics-to-audio alignment," *arXiv preprint arXiv:1906.10369*, 2019.

[24] D. Stoller, S. Durand, and S. Ewert, "End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model," *arXiv preprint arXiv:1902.06797*, 2019.

[25] C. Gupta, E. Yılmaz, and H. Li, "Automatic lyrics alignment and transcription in polyphonic music: Does background music help?" in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 496–500.

[26] X. Wang and Y. Wang, "Improving content-based and hybrid music recommendation using deep learning," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 627–636.

[27] N. Zeghidour, Q. Xu, V. Liptchinsky, N. Usunier, G. Synnaeve, and R. Collobert, "Fully convolutional speech recognition," *arXiv preprint arXiv:1812.06864*, 2018.

[28] J. Serrà, S. Pascual, and A. Karatzoglou, "Towards a universal neural network encoder for time series." in *CCIA*, 2018.

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

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