AUDIO BASED DISAMBIGUATION OF MUSIC GENRE TAGS

Romain Hennequin, Jimena Royo-Letelier, Manuel Moussallam
Deezer R&D, Paris
research@deezer.com

ABSTRACT

In this paper, we propose to infer music genre embeddings from audio datasets carrying semantic information about genres. We show that such embeddings can be used for disambiguating genre tags (identification of different labels for the same genre, tag translation from a tag system to another, inference of hierarchical taxonomies on these genre tags). These embeddings are built by training a deep convolutional neural network genre classifier with large audio datasets annotated with a flat tag system. We show empirically that they makes it possible to retrieve the original taxonomy of a tag system, spot duplicates tags and translate tags from a tag system to another.

1. INTRODUCTION

Large genre annotated databases have been made available lately: the Google Audio Set (GAS) [11], the MuMu dataset [19], Discogs [1] or the Free Music Archive (FMA) dataset [6] all contain hundreds of genre tags and hundreds of thousands multi-label genre track annotations.

Every dataset with genre annotations has its own genre representation: usually it is a tag set which is sometimes organized with a basic taxonomy (Discogs, MuMu, FMA) or a basic ontology (GAS).

However these representations usually suffer from ambiguity issues. First, tag definition may not be explicit: for the same tag name, definition may not be coherent from a dataset to another which prevents from doing correct translation from one tag set to another with a simple string matching. Second, there may be duplicated tags *i.e.* tag with different names but referring to the exact same genre such as *Bossa Nova* and *Bossanova* (without space) in Discogs. Thirdly, there may be polysemy issues for some tags: it happens that a single tag refers to different concepts. In Discogs, the tag *hardcore* may refer to *hardcore punk* or to *hardcore electronic music* which are quite different genres. Finally, while a tag set may be structured in a taxonomy or an ontology, those have limitation for expressing all relations between tags: for instance the tag *Blues Rock* in the MuMu taxonomy is a subgenre of *Rock* and is not related to *Blues*, which makes it as close to *Electric Blues* as to *Drum & Bass* according to the taxonomy. Moreover taxonomy and ontology are generally designed with a particular purpose in mind [20], possibly clarity for the customer for the MuMu taxonomy (which is the Amazon taxonomy), while it may be musicologic precision for DBpedia [1], which may result in different meaning for tags and different relationship between them.

Building a genre representation from these tag systems in order to deal with these ambiguity issues can be done using a top-down approach, using an expert-level ontology such as the DBpedia ontology and trying to project the tag system into this ontology [7]. Mapping tags to an external expert ontology is not trivial, as a genre can have several different name and some tags may have several meanings: the tag *funk* for instance may refer to a genre born in the 60s derived from soul and jazz, or, in Brazil, to Funk carioca which is a totally different style inspired by gangsta rap music. It also can be done using a bottom up approach, inferring relations between entities from data. The latter was mainly done using the genre tag distribution of a dataset with Latent Semantic Analysis (LSA) [26] or with a straight use of cooccurrences [24, 25] which all rely on the distributional hypothesis (similar tags are tags that cooccur a lot with same other tags). However, it is sometimes not possible to rely only on tag distributions: the MuMu dataset has no overlap with the GAS, which prevents from using tags cooccurrences to infer relationship between MuMu genre tags and GAS ones.

So far, the literature has been mainly focusing on music genre classification on flat tag systems from audio [4, 8, 9, 21, 22, 28], text such as reviews [12, 18] or lyrics [3, 16], album covers [15] or combinations of the previous modalities [17, 19, 23], while rarely addressing the actual semantic relationships that exist between genres. In [27], the authors pointed out that focusing on classification metrics was not sufficient and suggested a deeper results analysis such as explanation of the confusion of the classifiers in term of musicological aspects. In this paper, we suggest going deeper in this direction and seeing how the confusion of the classifier is able to generate a structured genre representation: if the classifier is good enough, the confusion it makes should be able to encode the relation of proximity between genres. Showing this property has two implications: it shows in a qualitative way that the classifier performs well and allows generation of a structured representation of a tag system using audio.

In this paper, we thus aim to disambiguate genre tags

[1] wiki.dbpedia.org

and relations between them using audio as an alternative to the distributional hypothesis: we propose a method able to spot inconsistencies, help reducing them and relate tags between themselves, possibly across different non overlapping datasets with different tag systems. We enforce that the representation is based on audio only information and not on tag distribution using a monolabel learning scheme. While extracting a semantic representation from audio annotated with a flat tag representation was already sparsely addressed (in [13], basic ontological relations between a few instruments are learnt back from isolated music instruments sounds and in [14] a simple music genre taxonomy is learnt with a few genre concepts), in this paper, we propose to learn representations at a large scale for tag systems with several hundreds of genre tags and with datasets of several hundreds of thousands of songs.

In Section 2, we explain how we compute genre tag embeddings using an audio-based genre classifier and use them to define an audio-based similarity between genre tags. In Section 3, we validate the learnt similarity by showing that it performs fairly on two artificial tasks (Discogs taxonomy learning and artificial deduplication). In Section 4, we show how we can use the learnt similarity to translate tags from a dataset tag system to another. Finally, we draw conclusions in Section 5.

2. LEARNING A GENRE REPRESENTATION

In this section, we explain how we build embeddings of genre tags using a genre classifier with audio input. We associate to each genre tag t_i in the genre tag set $T = \{t_1, ..., t_{N_c}\}$ an embedding vector $\mathbf{f}(t_i) = \mathbf{v}_{t_i} \in \mathbb{R}^n$, such that $d(\mathbf{v}_{t_1}, \mathbf{v}_{t_2})$ should correspond to an audio similarity between genre tag t_1 and genre tag t_2.

2.1 Datasets

We use two large-scale genre annotated datasets for our experiments: The MuMu dataset [19], and the genre provided by the Discogs website [2]. We matched both datasets to Deezer track IDs using song metadata (album and artist names, and track titles). We extracted a 30s-long excerpt for each track (the position of the excerpt was sampled at random between the beginning and the end of the track). For tags with too few occurrences, we extracted several excerpts for balancing (as explained in Section 2.2). To avoid overlap between datasets we removed the 7260 tracks that belong to both datasets (in order to not affect the translation experiment of Section 4).

While each dataset provides a simple genre taxonomy, we do not rely on it in the classification stage and consider the genre annotations as flat tag systems with no links between tags. The provided taxonomies are used afterwards for evaluation of the built genre representation.

2.1.1 Discogs

Discogs is referred as the "*largest open database containing explicit crowd-sourced genre annotations*" in [1]. It contains genre annotations at the album level for hundreds of thousands of albums. Genre tags in Discogs are organized in a two-level hierarchy: the first level, referred as *genre*, includes generic genre categories (*genre:Rock* [3], *genre:Jazz*, etc…) and the second level, referred as *style*, corresponds to subgenres (*Psychedelic Rock*, *Cool Jazz*, etc…). It contains a total of more than 500 genre/style tags. Only the 250 most common tags were kept in our experiments (235 style tags and 15 genre tags).

After cleaning, balancing (see Section 2.2) and matching, the Discogs dataset we used contained 418184 tracks.

2.1.2 MuMu dataset

The MuMu dataset [19] has genre annotation based on the Amazon 4-level genre taxonomy. It contains genre annotations at the album level for 31471 albums which contain a total of 147295 tracks. It contains a total of 446 different genres. Only the top 211 tags (those with less than 300 annotated tracks are discarded) are kept. After cleaning, balancing (see Section 2.2) and matching, the final MuMu dataset we used contained 122014 tracks.

2.1.3 Dataset split

When training the system described in Section 2.3, we split the datasets into a training dataset (70%), a validation dataset (10%) used for early stopping, and a test dataset (20%) used for building genre representations (Section 2.4). The split was done at the artist level meaning two tracks by the same artist are in the same part of the split in order to avoid overfitting on variables such as album or artist.

2.2 Monolabel learning

The annotations in a multilabel dataset carry information of popularity (through number of occurrences of a tag) and of similarity (through cooccurrences of tags). This information was already used in several papers to build genre taxonomies from a flat tag system [24–26] or to build a target representation to improve classification results [19].

The goal of the paper is to learn a genre representation only through audio and to avoid using non-audio information such as the one provided by the tag distribution. As this distribution can be easily learnt as a side information in the last layer of a neural network, where bias can encode popularity (higher bias for more popular genre) while weights can encode similarity between genres (important value of dot product between weights corresponding to similar genres and vice versa), simply training a multilabel audio classifier based on a neural net will result in taking advantage of this information, and it may be difficult to assess at what point the actual audio information is relevant in building the representation from this classifier.

In order to avoid influence of these non-audio information in the built genre representation, we propose to turn the multilabel classification problem into a monolabel one using the following learning scheme:

[2] https://discogs.com

[3] We prefix Discogs genre by *"genre:"* to distinguish them from style

- To remove cooccurrences information, we transform the multilabel dataset into a monolabel one by sampling a tag among the multilabel tag annotation of every track.
- To remove the popularity information, we balance equally all classes using a sampling probability inversely proportional to the global popularity of a tag (note, that it does not enforce perfect balancing).

For instance, if *Rock* appears 1000 times in the dataset and *Punk* appears 100 times, a song with (multi-)labels $\{$*Rock*, *Punk*$\}$ will get as monolabel *Rock* with probability $1/11$ and *Punk* with probability $10/11$. This ensures that rare genre tags have a high probability of being drawn, and that we keep the maximum of available information for rare tags while discarding somewhat redundant information for very common tags.

To enforce balancing, tags with too many occurrences are downsampled to keep a maximum of 2000 occurrences per tag. Genre with not enough occurrences are upsampled to 2000 occurrences by duplicating tracks (different 30s excerpts are chosen for each track).

In order to avoid fitting independent variables, the sampling is done at the album level, which means that every track from the same album gets the same label. It also ensures that different excerpts of the same track have the same label. Using this learning scheme, the confusion between genres should result only from similarities in audio.

2.3 Classification system

We use a convolutional neural network with a recurrent layer on top of it as a monolabel classifier. We feed it with Mel-spectrograms computed with 1024 samples long Hann windows without overlap, with 96 Mel filters. Audio is first downsampled to 22050Hz and stereo channels are summed up. Mel-spectrogams were log compressed using the function $f(x) = \log(1 + Cx)$ where we chose $C = 10000$. It results in 646×96 input matrices.

The architecture of the neural network is quite similar to the one used in [4] for automatic tagging, but with half as many filters in the convolutional layers (we noticed that it resulted in less overfitting) and a Gated Recurrent Unit [2] on top of the conv layer (which improved overall classification accuracy). The gated linear unit was used for temporal pooling (only last temporal output is forwarded to the last layer which removes the time dimension) and was used in conjunction with dropout to reduce overfitting. The architecture is summed up in Table 1.

The network was trained with a categorical cross-entropy loss with mini-batch stochastic gradient descent using Adadelta [30] and early stopping on the validation loss. The system was implemented with Keras [5] using the Tensorflow [10] backend.

As the main goal of the paper is not to perform in terms of classification results, we did not try to optimize thoroughly the architecture and we just checked that our proposed system had similar classification results as in [19].

Layer	output shape	N param.
Log-comp Mel-spec	646×96×1	0
Conv 3×3×64 - MP 2×2	323×48×64	640
Conv 3×3×128 - MP 3×4	107×12×128	1280
Conv 3×3×256 - MP 2×3	53×4×256	2560
Conv 3×3×512 - MP 3×4	17×1×512	5120
GRU 512	512	1574400
Dense Softmax	N_c	512×N_c

Table 1. Architecture of the Neural Network. MP stands for Max Pooling.

2.4 Genre embeddings from classification

There are several ways of extracting an embedding from a neural net based classification system. We describe the three kinds of genre embeddings we generated from the audio classifier in the following subsections. Whereas the first embedding only uses parameters of the classifiers, the other two make use of the test set.

2.4.1 Last hidden layer weights

The weights of the last hidden layer $\mathbf{W}$ are a $512 \times N_c$ matrix. The i-th column of this matrix is then chosen as the embedding of genre tag t_i:

$$\mathbf{f}_w(t_i) = \mathbf{v}_{t_i} = \mathbf{W}_{:,i}. \tag{1}$$

This is a straightforward representation of a genre tag in the network: the output of the last hidden layer for a track annotated with some genre should be similar (in terms of dot product) to the weight vector of this genre. However, it necessitates retraining to incorporate new genre tags in the embedding.

2.4.2 Columns of output

We can also build an embedding using the test dataset: for every track s in the test dataset, we denote T_s the set of tags associated to s. We note the test dataset $S = \{s_1, s_2 \dots s_{N_s}\}$ where s_k are the track excerpts. The output of the network when fed with track excerpt s_i is a vector $\mathbf{p}_k \in [0,1]^{N_c}$ (with $\sum_{j=1}^{N_c} [\mathbf{p}_k]_j = 1$). We note $\mathbf{P}$ the matrix in $\mathbb{R}^{N_s \times N_c}$ with $\mathbf{p}_k$ as k-th row. The embedding of tag t_i is then defined as the i-th column of matrix $\mathbf{P}$:

$$\mathbf{f}_c(t_i) = \mathbf{v}_{t_i} = \mathbf{P}_{:,i}. \tag{2}$$

This embedding does not require annotation information about the tracks of the test set and the cosine similarity matrix between embeddings of all pairs of genre can be understood as a normalized confusion matrix and is the audio counterpart of the occurrence based representation defined in (4). However it has very large dimension (that may be reduced using dimension reduction techniques such as LSA) and it is quite difficult to add extra genres without retraining the whole system.

2.4.3 Mean of output

This third embedding type also uses the test dataset and takes advantage of the annotations. We note $S_{t_i} = \{s_{k_1}, s_{k_2} \dots s_{k_{N_t}}\}$ the set of tracks annotated with genre tag t_i. We then associate to each t_i the set of outputs of the classifier $\{\mathbf{p}_k | s_k \in S_{t_i}\}$. Ideally each genre tag t_i would be represented by the distribution of all possible outputs for this genre. In practice, we compute statistics on these distributions. We then define the third type of genre tag embeddings as the mean of the output:

$$\mathbf{f}_m(t_i) = \mathbf{v}_{t_i} = \frac{1}{|S_{t_i}|} \sum_{s_k \in S_{t_i}} \mathbf{p}_k. \quad (3)$$

As $\mathbf{p}_k$ is a categorical probability distribution, $\mathbf{f}_m(t_i)$ is too. Embedding $\mathbf{f}_m$ makes it possible to incorporate new tags without retraining the whole system, by simply adding tracks annotated with the new genre tag in the dataset (the only constraints would be that the classifier was trained with similar genres): this is an important property of the embedding since it makes it much easier to incorporate new knowledge from another tag system.

2.4.4 Occurrence based representation

In order to compare the audio-based representation we also define the following representation which is not based on audio but on tag distribution only. We note $\mathbf{M} \in \{\mathbf{0}, \mathbf{1}\}^{\mathbf{N_s} \times \mathbf{N_c}}$ the multilabel tag occurrence matrix with coefficient $M_{ij} = 1$ iff track s_i is annotated with tag t_j. The coocurrence embedding of tag t_i is then defined as the i-th column of matrix $\mathbf{M}$:

$$\mathbf{f}_{\text{dist}}(t_i) = \mathbf{M}_{:,i}. \quad (4)$$

This definition then shares similarity with the audio-based representation $\mathbf{f}_c$.

2.4.5 Similarity measure

To compare tags, we use the cosine similarity applied to the four types of genre tag embeddings defined in Equations (1), (2), (3) and (4).

3. MODEL VALIDATION

In this section, we validate that the audio-based similarities learnt in Section 2 have a semantic meaning by showing that the original Discogs taxonomical relations can be inferred from the similarities and that they make it possible to spot duplicate tags in a dataset. In order to reproduce the results, we make available the embeddings, the similarity matrices we obtained for the different representation [4] as well as dataset files (as lists of Deezer song IDs).

3.1 Taxonomy Learning

In this section, we use similarity obtained from the genre embeddings described in Section 2, to infer hierarchical links between genres. We trained the classification system with the Discogs dataset and the purpose of the experiment is to infer the genre/style links of the two-level Discogs taxonomy from audio.

The cosine similarity computed between genre tag embeddings provides a measure of similarity between genre tags. This can be used to rank for each style the similarity with each of the 15 genres. The ground truth is the actual genre associated to the style in the Discogs taxonomy (note that some rare style are associated to 2 music genres, such as *hardcore* and *noise* which are associated to both *rock* and *electronic*). We measure the quality of this ranking with classic ranking metrics: Hit Rate (HR)@k which is the percentage of style for which the associated genre is in the top-k according to the similiarity score.(HR@1 can be considered as a classification accuracy) and Mean Average Precision (MAP) as defined in [31]. MAP takes into account the rank of the related genre in the similarity list.

	$\mathbf{f}_w$	$\mathbf{f}_c$	$\mathbf{f}_m$	$\mathbf{f}_{\text{dist}}$
HR@1	85.1±4.6	89.4±3.9	87.7±5.2	96.2±2.5
HR@2	91.9±3.5	98.3±1.7	96.2±3.2	100.0±0
MAP	90.6±2.9	94.2±2.2	93.1±3.0	98.1±1.2

Table 2. Average ranking metrics (in %) for the Discogs taxonomy learning task with 95% confidence intervals.

Results are presented in Table 2. As a reference, we report results for the occurrence based embedding $\mathbf{f}_{\text{dist}}$. As style tags are always present together with their parent genre tag in the annotations, the performance of the occurrence based representation should be interpreted as an upper-bound for the results of the other representations, the errors being likely due to incoherences in the Discogs taxonomy (which is confirmed by the perfect HR@2 score of $\mathbf{f}_{\text{dist}}$). Among the audio-based representations, $\mathbf{f}_c$ performs better than the two others. Despite being smaller than the occurrence based representation, we can see that the metrics for the audio representations are quite high, notably for $\mathbf{f}_c$ which has a near perfect HR@2. This is noteworthy, since only audio information is used to infer the relations.

A qualitative analysis of the error shows that most of the "errors" (in the sense that the most similar genre to a style is not is related genre) actually make sense: for instance *blues rock* which is a subgenre of *genre:rock* in Discogs taxonomy has the greatest similarity (for $\mathbf{f}_c$) with *genre:blues* which makes as much sense as the other (the same phenomena with hybrid subgenre appears with *jazz-funk* and *genre:funk / soul* instead of *genre:jazz*, *pop rock* and *genre:rock* instead of *genre:pop* and *soul-jazz* and *genre:funk / soul* instead of *genre:jazz*). Other noteworthy examples are *bossa nova* (subgenre of *genre:jazz*) associated with *genre:latin*, *musique concrète* (subgenre of *genre:electronic*) associated to *genre:non-music* or *rnb/swing* (subgenre of *genre:hip hop*) associated to *genre:funk / soul*. These qualitative results confirms that most of the "errors" are actually due to limitations of the original taxonomy and that HR@2 may be the most revealing metric. In Figure 1, we plot a 2D t-distributed stochastic neighbor embedding (t-SNE) [29] of the learnt audio representation $\mathbf{f}_w$ in order to get visual insights about it: most music style tags are gathered in coherent clusters and are most of the time close to their related genre tag. A noteworthy exception is the style tags related to *folk, world, & country* that form several clusters, one of which being next to *latin*, another one being next to *blues* and an-

[4] `github.com/deezer/audio_based_disambiguation_of_music_genre_tags.git`

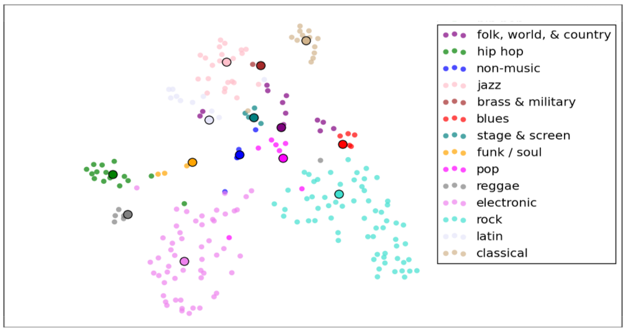

Figure 1. 2D t-SNE of $\mathbf{f}_w$ for the Discogs tags. Each *style* is colored with the same color as its related *genre*. Main genres are depicted with bigger circle and black edges.

other one next to *pop*. This is pretty coherent since the tag *folk, world, & country* is supposed to gather several very different styles that may be closely related to other genres.

3.2 Tag deduplication

In this section, we show how the audio-based similarities learnt in Section 2 can be used to spot duplicates in a tag system. To do that we rely on the ability of a classifier based on audio data to discriminate between two genre tags. If two genre tags cannot be discriminated, they probably have some strong relation (even if they have very dissimilar names). There may be several reasons for two tags having high confusion similarity: First, they may represent the exact same genre. Second, genre related audio characteristics may be very similar (the genre may be very similar with respect to audio). Thirdly, there may be differences of distribution in the datasets: datasets are usually an imperfect sample of the set of all music. Some genre may be biased toward a subgenre in a dataset while not in another one, which may result in strong differences in the meaning of some genres. Last, the classifier may not be able to distinguish them while there exists difference in some audio characteristics (that the classifier is not able to handle).

As it is very difficult to assess a ground truth for such a deduplication experiment, we propose the following artificial tag duplication: we use the Discogs genre dataset. We artificially duplicate every genre tag by creating two duplicate tags: for instance, *Rock* is duplicated into *Rock1* and *Rock2*, which means that half of the tracks originally annotated with *Rock* get the annotation *Rock1* instead while the other half get the annotation *Rock2*. To avoid learning the similarity through artist specific characteristics, we perform the split at the artist level, meaning that tracks of the same artist annotated with *Rock* will get all the same subtag (either *Rock1* or *Rock2*). Note that a subtag of group 1 cannot cooccur with a subtag of group 2, which results in two separate tag systems (that we will refer as system 1 and system 2), with no overlap. While all tags from system 1 having a semantically equivalent counterpart in system 2 is quite artificial, the total separation between the tag systems in term of cooccurrences is realistic. There is, for instance, no overlap between the GAS and the MuMu dataset which means we can only rely on audio for linking them.

In a similar way as in the experiment of Section 3.1,

	$\mathbf{f}_w$	$\mathbf{f}_c$	$\mathbf{f}_m$
HR@1	92.0 ± 2.4	92.8 ± 2.3	74.8 ± 3.8
HR@2	95.8±1.8	97.0±1.5	83.0±3.3
MAP	98.1 ± 0.6	98.4 ± 0.5	93.3 ± 1.1

Table 3. Average ranking metrics (in %) for the Discogs deduplication task with 95% confidence intervals.

we use the similarity between genre tags embeddings as a *duplication score*. The task is then for each genre tag, to retrieve its duplicated tag. Once again, we present quantitative results in terms of HR@k and MAP in Table 3. As opposed to the taxonomy learning task, it does not make sense to compare the audio based representations to the occurrence-based representation since the sampling scheme we use avoid a tag of group 1 cooccurring with a tag of group 2 which means that the cosine similarity between any tag of group 1 with any tag of group 2 is 0. $\mathbf{f}_w$ and $\mathbf{f}_c$ performs similarly, both performing significantly better than $\mathbf{f}_m$. Once again the score seems reasonably high for a representation based on audio information only.

It is interesting to look at the "errors" (when the most similar tag is not the actual duplicate) done by the system using $\mathbf{f}_c$. Some errors were actual duplicates in Discogs: *bossa nova* was associated to *bossanova* (without a space) which is clearly a duplicate issue in Discogs. Other example are *style:reggae* and *genre:reggae* (where a *style* tag as the same name as its related *genre* tag) or *thug rap* and *gangsta* (considered as the same genre in Wikipedia). This shows that the genre similarity computed from the embeddings is able to spot actual duplicates and that HR@2 may be again the most revealing metric. Some errors are matching between quite different concepts but with very similar audio, such as *field recording/musique concrète*, *poetry/spoken word*, *spoken word/genre:non-music* and *conscious/genre:hip hop*. Other errors are with very similar genres: *bop/hard bop*, *honky tonk/country blues*, *space rock/post rock* A few errors are more difficult to explain such as *ragtime/tango* which may have some audio similarities (the use of piano is quite common in both genres, and both are intended for dancing). These errors may come from the classification system we use or from a strong bias or annotation noise in the Discogs annotations.

4. TAGS TRANSLATION

In this section, we perform another experiment that aims at translating tags from MuMu dataset to Discogs dataset. For sack of clarity Discogs tags are prefixed with "*D:*" and MuMu tags with "*M:*". When there are no or few overlaps between two datasets, we cannot rely on cooccurrences of tags to model relation between the tag systems. The only media we can rely on is then audio.

To train the classifier (see Section 2.3), we used the concatenation of the tags from the MuMu dataset and the Discogs dataset. Tags of each dataset were considered different even if they had the exact same name: *e.g.*, there were a *M:jazz* tag that was considered different from the *D:jazz* tag. The experiment of translation is then very similar to the deduplication task presented in 3.2:

Audio-based translation $\mathbf{f}_c$		Cooccurrence-based translation $\mathbf{f}_{\text{dist}}$	
Mumu tag	Discogs tag	Mumu tag	Discogs tag
bebop	bop	irish folk	celtic
movie scores	score	contemporary big band	big band
indie & lo-fi	lo-fi	latin music	genre:latin
electric blues	modern elec. blues	rap & hip-hop	genre:hip hop
electronica	leftfield	vocal blues	ragtime
punk-pop	pop punk	dance & electronic	genre:electronic
modern postbebop	genre:jazz	today's country	country
special interest	avantgarde	electric blues	genre:blues
singer-songwriters	folk rock	children's music	genre:children's
r&b	rnb/swing	comedy & spoken word	comedy

Table 4. Top 10 most similar tags between MuMu and Discogs according to $\mathbf{f}_c$ (left columns) and $\mathbf{f}_{\text{dist}}$ (right columns), removing string matched tags.

the translation task consists of deduplicating the whole MuMu/Discogs tag set, focusing on pairs of duplicates for which the first element is a MuMu tag and the second element is a Discogs tag.

This allows to translate tags from one tag system to another, but also to spot possible genre definition differences between datasets: if two genre tags from two different datasets, with the exact same name can be discriminated with audio, this is probably because they do not carry the exact same meaning (provided we can move appart overfitting of the audio classifier used to build the representation).

We only consider here simple one-to-one tag mappings between MuMu and Discogs although it is restrictive since there may exist one-to-many mapping (e.g. between *M:avant garde & free jazz* and *D:avant-garde jazz*/*D:free jazz*) or even more complex relationships.

As the Discogs and MuMu datasets have some common tracks, we can compare the audio-based similarities with the cooccurrence-based one derived from $\mathbf{f}_{\text{dist}}$.

There are two aspects that may be qualitatively assessed: why would two tags with different names be associated? and why would two tags with same name have a very low audio similarity.

In the two first columns of Table 4, we present the 10 Discogs tags that are most similar (according to $\mathbf{f}_c$) to MuMu tags while not having the same normalized name. As can be seen, when the names are different, it can be due to the following reasons:

- Two different names are used for the exact same concept: *M:bebop*/*D:bop*, *M:punk-pop*/*D:pop punk*, *M:movie scores*/*D:score*.
- Some genres were considered sufficiently similar to be grouped under the same tag name in one of the tag system while they were not in the other one *e.g. M:indie & lo-fi*/*D:lo-fi*, *M:r&bD:rnb/swing*.
- One genre is a subgenre of the other: *M:electric blues*/*D:modern electric blues*, *M:modern postbebop*/*D:genre:jazz*

The association between *M:singer-songwriters* (a subgenre of *M:rock*) with *D:folk rock* (a subgenre of *D:genre:rock*) seems to link quite similar concepts (which seems to be confirmed by the cooccurrence based similarity that is quite high). *M:electronica* and *D:leftfield* seem to be quite broad electronic genres: the span of the former and the lack of precise definition of the latter while both seem not intended for dancing could explain the association. The association *M:special interest*/*D:avantgarde* remains quite unclear, while the tags are quite vague.

In the two last columns of Table 4 are presented top 10 most similar tags between MuMu and Discogs according to the cooccurrence based similarity (excluding string matched pairs with basic normalization as in [24]). It can be seen that the top 10 for cooccurrences and the top 10 for audio similarity contains mostly different tags, with the exception of *M:electric blues* which is not mapped to the same Discogs tag: this tends to show that cooccurrence similarity is complementary to the audio-based similarity, and when cooccurrence information is available (overlap between dataset), using both similarities should provide the best analysis. This is confirmed with some MuMu/Discogs pairs such as *M:bebop*/*D:bop* and *M:post hardcore*/*D:post-hardcore* which seems to be perfect mapping and have very high audio similarity but very low (less than 0.1) cooccurrence similarity. The low cooccurrence similarity may be explained by a lack of data for these tags.

On the other hand, it is also interesting to check tags with the exact same name in both datasets, but with quite low similarity score: the tags *electronic*, *instrumental* have very low similarity (according to both $\mathbf{f}_c$ and $\mathbf{f}_{\text{dist}}$) from one database to another. *D:electronic* refers to a generic term for describing all electronic music while this exact same concept seemed to be carried by *M:dance & electronic* in MuMu. *M:electronic* is actually a subgenre of *M:progressive* which is a subgenre of *M:rock* and then has a very different meaning than the one in Discogs. *instrumental* (which is not a genre by itself) is considered a subgenre of *M:new age* and *M:country* in the MuMu taxonomy and a subgenre of *D:hip hop* in Discogs (while a large number of non-hip-hop songs seems to have the *D:instrumental* tag).

Thus, audio made it possible to spot significantly different genres that were represented by the exact same string. This highlights that the meaning of some genre may vary significantly from a database to another and that string matching can result in wrongly matched concepts.

5. CONCLUSION

In this paper we presented a way of learning genre embeddings from audio and showed that they are able to encode semantic similarities between genre tags: we showed that these embeddings were able to build genre taxonomies, to spot duplicates in a dataset or to translate genre from one tag set to another one. In future works, we plan to explore extraction of structured representation of other tag types than genre (mood, instruments, country...) from audio and to exploit other datasets such as the FMA dataset or GAS to learn a more global representation. We also plan to explore in more details how we can use several sources (audio, expert based ontology, string matching, cooccurrences) to build richer representation from flat tag systems.

6. ACKNOWLEDGEMENTS

The authors would like to thank Guillaume Salha for fruitful conversations and Matt Mould for proof-reading.

7. REFERENCES

[1] Dmitry Bogdanov and Xavier Serra. Quantifying Music Trends and Facts Using Editorial Metadata From the Discogs Database. In *International Society for Music Information Retrieval Conference*, pages 89–95, 2017.

[2] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Empirical Methods in Natural Language Processing*, 2014.

[3] Kahyun Choi, Jin Ha Lee, and J. Stephen Downie. What is this song about anyway?: Automatic classification of subject using user interpretations and lyrics. In *ACM/IEEE Joint Conference on Digital Libraries*, pages 453–454. IEEE, sep 2014.

[4] Keunwoo Choi, George Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks. In *International Society for Music Information Retrieval Conference*, pages 805–811, 2016.

[5] François Chollet. Keras: Deep learning library for theano and tensorflow, 2015.

[6] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. FMA: A Dataset For Music Analysis. In *International Society for Music Information Retrieval Conference*, 2017.

[7] Dennis Diefenbach, Pierre René Lhérisson, Fabrice Muhlenbach, and Pierre Maret. Computing the semantic relatedness of music genres using semantic web data. In *CEUR Workshop*, volume 1695, 2016.

[8] Sander Dieleman, Philémon Brakel, and Benjamin Schrauwen. Audio-based Music Classification with a Pretrained Convolutional Network. In *International Society for Music Information Retrieval Conference*, pages 669–674, 2011.

[9] Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6964–6968, 2014.

[10] Martín Abadi et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2015.

[11] Jort F Gemmeke, Daniel P.W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio Set: An ontology and human-labeled dataset for audio events. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 776–780, 2017.

[12] Xiao Hu, JS Downie, Kris West, and AF Ehmann. Mining Music Reviews: Promising Preliminary Results. In *International Society for Music Information Retrieval Conference*, pages 536–539, 2005.

[13] Şefki Kolozali, Mathieu Barthet, György Fazekas, and Mark Sandler. Automatic ontology generation for musical instruments based on audio analysis. *IEEE Transactions on Audio, Speech and Language Processing*, 21(10):1–14, 2013.

[14] Tao Li and Midsunori Ogihara. Music genre classification with taxonomy. In *Acoustics, Speech, and Signal Processing*, pages 197–200, 2005.

[15] Janis Libeks and Douglas Turnbull. You can judge an artist by an album cover: Using images for music annotation. *IEEE Multimedia*, 18(4):30–37, apr 2011.

[16] Rudolf Mayer, Robert Neumayer, and Andreas Rauber. Rhyme and Style Features for Musical Genre Classification By Song Lyrics. In *International Society for Music Information Retrieval Conference*, pages 337–342, 2008.

[17] Robert Neumayer and Andreas Rauber. Integration of Text and Audio Features for Genre Classification in Music Information Retrieval. In *Advances in Information Retrieval*, pages 724–727. 2007.

[18] Sergio Oramas, Luis Espinosa-anke, Aonghus Lawlor, Xavier Serra, Horacio Saggion, Music Technology Group, Universitat Pompeu Fabra, and Universitat Pompeu Fabra. Exploring Customer Reviews for Music Genre Classification and Evolutionary Studies. In *International Society for Music Information Retrieval Conference*, 2016.

[19] Sergio Oramas, Oriol Nieto, Francesco Barbieri, and Xavier Serra. Multi-label Music Genre Classification from Audio, Text, and Images Using Deep Features. In *International Society for Music Information Retrieval Conference*, 2017.

[20] François Pachet and Daniel Cazaly. A Taxonomy of Musical Genres. In *Content-Based Multimedia Information Access Conference*, pages 1238–1245, 2000.

[21] Jordi Pons, Thomas Lidy, and Xavier Serra. Experimenting with musically motivated convolutional neural networks. In *International Workshop on Content-Based Multimedia Indexing*, volume 2016-June, pages 1–6. IEEE, jun 2016.

[22] Chris Sanden and John Z. Zhang. Enhancing Multi-label Music Genre Classification Through Ensemble Techniques. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 705–714, New York, New York, USA, 2011. ACM Press.

[23] Alexander Schindler and Andreas Rauber. An Audio-Visual Approach to Music Genre Classification through Affective Color Features. In *European Conference on Information Retrieval*, pages 61–67, 2015.

[24] Hendrik Schreiber. Improving genre annotations for the million song dataset. In *16th International Society for Music Information Retrieval Conference*, pages 241–247, 2015.

[25] Hendrik Schreiber. Genre Ontology Learning: Comparing Curated With Crowd-Sourced Ontologies. *17th International Society for Music Information Retrieval Conference*, pages 400–406, 2016.

[26] Mohamed Sordo, Oscar Celma, Martín Blech, and Enric Guaus. The Quest for Musical Genres: Do the Experts and the Wisdom of Crowds Agree? In *9th International Conference on Music Information Retrieval*, pages 255–260, 2008.

[27] Bob L. Sturm. Aalborg Universitet Classification Accuracy Is Not Enough Classification Accuracy Is Not Enough On the Analysis of Music Genre Recognition Systems. *Journal of Intelligent Information Systems*, 41(3):371–406, 2013.

[28] George Tzanetakis and Perry Cook. Musical Genre Classification of Audio Signals. *IEEE Transactions On Speech And Audio Processing*, 10(5), 2002.

[29] Laurens Van Der Maaten and Geoffrey Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[30] Matthew D Zeiler. ADADELTA: An Adaptive Learning Rate Method, 2012.

[31] Mu Zhu. Recall, precision and average precision, 2004.