SEMI-AUTOMATED MUSIC CATALOG CURATION USING AUDIO AND METADATA

Brian Regan
Spotify
brianr@spotify.com

Desislava Hristova
Spotify
desih@spotify.com

Mariano Beguerisse-Díaz
Spotify
marianob@spotify.com

ABSTRACT

We present a system to assist Subject Matter Experts (SMEs) to curate large online music catalogs. The system detects releases that are incorrectly attributed to an artist discography (misattribution), when the discography of a single artist is incorrectly separated (duplication), and predicts suitable relocations of misattributed releases. We use historical discography corrections to train and evaluate our system's component models. These models combine vector representations of audio with metadata-based features, which outperform models based on audio or metadata alone. We conduct three experiments with SMEs in which our system detects misattribution in artist discographies with precision greater than 77%, duplication with precision greater than 71%, and by combining the approaches, predicts a correct relocation for misattributed releases with precision up to 45%. These results demonstrate the potential of such proactive curation systems in saving valuable human time and effort by directing attention where it is most needed.

1. INTRODUCTION

Online music catalogs such as Spotify's contain millions of releases, and new ones are added daily by providers ranging from professionally-staffed music labels to DIY artists via aggregators. In such large catalogs, it is common that multiple artists share the same or similar names, or that content by one artist comes from different providers. For example, there are 14 distinct metal bands with the name *Burial* [1] . When a new release by a *Burial* makes it to the catalog, in the absence of a unique artist identifier, we must make a decision of where to place the content: Is it by the Italian doom metal band, the English death metal band, one of the other 12 bands named *Burial*, or an entirely new one? In general, *to which artist do we attribute a release when there are multiple artists with the same name?*

Music streaming services have multiple systems to ensure that releases are correctly placed on artist discographies. However, given the large volumes of content and the diversity of sources, it is inevitable that on rare occasions a release is incorrectly attributed (e.g. due to incomplete or incorrect metadata, extreme ambiguity, or human error). These errors can manifest in two different ways: 1) *Misattribution*: when a release is incorrectly attributed to an artist, so that their discography now contains releases from two separate real-world artists; 2) *Duplication*: when a release is not attributed to the correct existing discography but to a new one, so that a single artist's work is split across the two discographies. These errors negatively impact the experience of both artists and users on the platform.

The problem of Named Entity Disambiguation (NED) has been extensively researched to attribute scientific papers to homonym authors using metadata such as the author's fields of research, academic affiliations, and co-authors [1–3]. In Music Information Retrieval (MIR), NED is primarily tackled as artist identification or multi-class classification with known artist classes. Approaches to this problem rely primarily on audio feature representations [4–6]. These methods cannot be applied to catalogs with a large or unknown number of artists, and do not take advantage of all existing information.

Here we present a semi-automated proactive curation system to detect and correct attribution errors across large music catalogs. The system consists of two machine learning sub-systems: a system for detecting misattribution by splitting discographies with releases from multiple real-world artists into their constituent sub-discographies (Fig. 1a), and a deduplication system that takes pairs of discographies or sub-discographies and decides if they should be combined (Fig. 1b). Both sub-systems rely on metadata and the acoustic similarity between releases, using deep convolutional network embeddings of their mel-spectrograms [7]. We show that combining audio and metadata features improves average precision in misattribution and duplicate detection by 10% and 6% respectively.

"In the wild" experiments with music catalog curation Subject Matter Experts (SMEs) show that our system achieves over 77% precision on misattribution detection, over 71% precision on duplicate detection, and 45% precision on finding the correct relocation of misattributed releases. Together these results demonstrate the power of proactive catalog correction systems in assisting human-led curation efforts.

[1] https://www.metal-archives.com/bands/Burial

2. RELATED WORK

Recent advances in audio feature representation using deep learning [8] have applications to recommendations [7], audio classification [4] and artist identification [4, 6, 9, 10]. These works typically focus on the audio and do not include additional information (the method in [9] uses genre in its negative sampling method, but the model takes only audio). Work in other Named Entity Disambiguation (NED) applications shows that combining learned feature representations and manually crafted diverse features outperforms using either in isolation [11, 12]. This suggests that combining multiple data types (e.g. content and metadata) can improve the performance of music NED systems.

Duplicate entity detection (also known as entity matching or entity resolution) across or within databases typically has a *blocking* step [13] optimised for recall to reduce the set of pairwise comparisons, followed by an *entity matching* step optimised for precision. If labelled pairs of entities are available, supervised machine learning approaches can be used for matching. These are typically based on various string-based similarity features, such as entity name similarity [14].

Although state-of-the-art NED research focuses on automation [1], a human-in-the-loop (HITL) paradigm is commonly used in practice. A HITL approach is useful for resolving highly ambiguous cases and correcting automated decisions. In [3] the authors describe a machine learning approach that optimises human effort spent on labelling for author disambiguation. In the Microsoft Academic Graph [2], the author disambiguation system uses crowdsourced data as supervision signals.

Crowdsourced and authoritative sources such as MusicBrainz [15], VIAF [16], Wikidata [17], or ISNI [18] are useful for artist name disambiguation, but their benefit is limited for artists in the long tail or for brand new releases without unique artist identifiers.

3. METHODOLOGY

Our system operates on music releases (i.e. albums) denoted as a, and on artist credits in them. The set of releases credited to an artist forms the artist's discography: $\mathcal{A} = \{a_i, a_2, \ldots\}$. The objective of our system is twofold:

Correct discographies Every release within a discography should credit the same real-world artist; i.e. there is no misattribution in the discography.

Complete discographies A real-world artist's releases should not be split across multiple discographies; i.e. there should be no more than one discography per artist.

Figure 1 illustrates our approach to achieve these goals; we achieve correctness and completeness by relocating misattributed releases and resolving (i.e. merging) duplicate discographies. Note that there are cases where a single real person performs under distinct artist identities (e.g. Dan Snaith performs as Caribou and Daphni). These

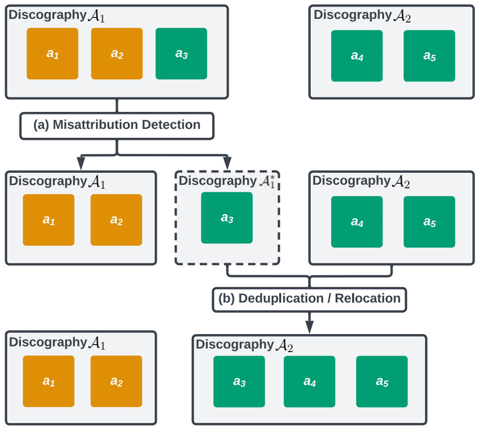


Figure 1: **System Overview:** (a) Misattribution detection is performed on each discography $\mathcal{A}$. The misattributed release a_3 is split out from $\mathcal{A}_1$ into *sub-discography* $\mathcal{A}_1^*$. (b) All (sub-)discographies are considered for deduplication; $\mathcal{A}_1^*$ is merged into $\mathcal{A}_2$, relocating the misattributed releases into the correct discography.

discographies should not be considered duplicates. In addition, some releases can belong to multiple discographies if they credit multiple distinct artists (e.g. collaborations and remixes); however, a discography should always contain releases under a common artist.

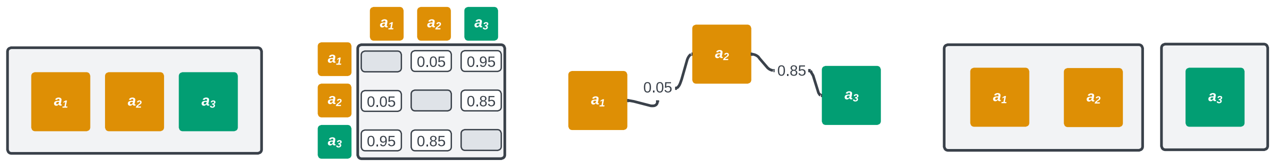
3.1 Misattribution Detection

The misattribution detection method, illustrated in Fig. 2, processes an artist's attributed discography $\mathcal{A}$ in two stages: First, we obtain a distance $\mathrm{dist}(a_i, a_j)$ between all pairs of releases $a_i, a_j \in \mathcal{A}$ using the combination of audio and metadata signals in Table 1. Second, we partition $\mathcal{A}$ using this distance by constructing a Minimum Spanning Tree (MST) [19] and imposing a threshold θ_{dist}. When we cut the MST edges where $\mathrm{dist}(a_i, a_j) > \theta_{\mathrm{dist}}$, the remaining connected components should contain releases from the same artist. These partitions are disjoint subsets: $\mathcal{A}_i \subseteq \mathcal{A}, i = 1 \ldots m$, for which all releases belong to the same real-world artist. If the cardinality of the partition is $m > 1$, then there is at least one misattributed release in the discography (i.e. more than one artist's content is detected) and the discography should be split.

3.1.1 Pairwise Model

To obtain the pairwise distance between releases in a discography, we train a Random Forest ensemble classifier [20] $\mathrm{dist} : \mathcal{A} \times \mathcal{A} \to (0, 1]$, where high values indicate that the releases are likely to be from different artists.

Data. The training data consists of ∼45K release pairs from ∼28K artist discographies. This data, which we call the *Relocations* dataset, contains historical corrections of artist misattributions. The genres of the releases in this data are representative of Spotify's catalog. Each relocation is a move of an incorrectly-placed release from an artist's discography to the correct one. To construct the training



(a) The artist discography containing misattribution (b) Compute pairwise distances (c) Compute Minimum Spanning Tree (d) Threshold Minimum Spanning Tree

Figure 2: **Misattribution detection**: *(a):* An artist discography $\mathcal{A} = \{a_1, a_2, a_3\}$ in which release a_3 is misattributed. *(b):* The pairwise distance matrix D computed using our model. *(c):* A Minimum Spanning Tree (MST) is computed from the distances. *(d):* After applying a threshold θ_{dist} to the MST, the discography $\mathcal{A}$ is split into two partitions, which correspond to the two distinct real world artists present in the discography.

data, consider a release a_1^i that was moved from discography $\mathcal{A}_j$ to $\mathcal{A}_i$. We pair a_1^i with a release $a_1^j \in \mathcal{A}_j$ from the discography where it was incorrectly located: $(a_1^i,\, a_1^j)$, and give it the *"mismatched"* label. Then, we pair a_1^i with a release from the correct discography: $(a_1^i,\, a_2^i)$, $a_2^i \in \mathcal{A}_i$, and give it the *"not mismatched"* label.

Model Features. We use a combination of metadata and audio-based features, summarised in Table 1. Audio features include deep acoustic embeddings from a proprietary model trained in a fashion similar to [7], originally developed for music recommendations, and *speechiness* - a probability that a track contains spoken word as determined by another proprietary model [21]. An advantage of audio features is that they are available for every release. In general, we expect releases from the same artist to sound similar to each other. As mentioned in Sec. 2, previous works report good performance using audio-based methods alone [4,9,10]. However, releases from different artists can also sound similar (e.g. if they come from the same genre), and releases from the same artist can be musically different (e.g. an artist whose style evolved or spans many genres).

On the other hand, metadata features such as music labels, composers or lyricists can have high precision (e.g. releases from the same discography delivered by the same label are likely to be by the same artist), but in isolation metadata matches can be sparse, or have mistakes. Therefore, we supplement audio similarity with metadata based features to improve the performance of our classifier.

3.1.2 Grouping releases in a discography

Our distance allows comparisons between individual pairs of releases to decide whether they belong to distinct artists. For example, if $\text{dist}(a_i,\, a_j) > \theta_{\text{dist}}$ for a given $\theta_{\text{dist}} \in (0, 1]$, we could say that it is unlikely that the releases share an artist. However, this comparison ignores the context of the whole discography $\mathcal{A}$, and may fail when the sound of an artist has evolved in time, the artists changed collaborators or labels throughout their career. To mitigate these issues, we consider each comparison in the context of all the releases in $\mathcal{A}$.

We construct the matrix $D \in \mathbb{R}^{m \times m}$ where $D_{ij} = \text{dist}(a_i,\, a_j)$, and use it to obtain a MST, which is a graph with node set $\mathcal{A}$, and edges with weight equal to the nodes' pairwise distance (see Fig. 2c). The MST connects releases that are "close" to each other, and provides a global summary of how the releases are organised in a latent space, while capturing the continuity of the data arising from evolution in the style and career of an artist. We can attribute two dissimilar releases to the same artist if there is a path of short hops along the MST that connects them. Put another way, if we cut very long hops (i.e. long edges) in the MST, we get connected components in which we can only go between nodes by a series of short hops. Our hypothesis is that these components (partitions of $\mathcal{A}$) are releases that

Attribute	Functions
Music Label*	Exact Match*, Dice Score [2]
Music Licensor*	Exact Match
Music Source*	Exact Match
Release Name	Exact Match, Dice Score
Release Group* [3]	Exact Match
Release Artists	Overlap, Dice Overlap [4]
Release Track Names*	At Least 1 Exact Match, Min Dice Score
Release Track Artists	Max Overlap, Max Dice Overlap
Release Track Language*	At Least One Exact Match
Release Type†*	Categorical
Release Is Remix†	Categorical
At Least One Track Is Remix†*	Categorical
Track Audio Vectors*	Min/Max/Mean Cosine Similarity
Track Speechiness†	Min/Max/Mean

Table 1: **Pairwise Model Inputs**. The features above the line are metadata, and below are audio-based. Features with * were included in the model for the SME experiment. Track level attributes are aggregated to release level with the functions described. Attributes with † produce two features, one for each release. Random permutations of underlined feature values decreased test-set performance >95% of the time.

[2] The Dice score is the average of the Dice coefficient [22] for n-gram values of 1,2,3 and 4.

[3] Indicates whether the pair of releases have been identified by other systems as duplicates

[4] Number of pairs of artists with Dice score > 0.7

are likely to be from the same artist. Specifically, we need to find a threshold θ_{dist} and cut all edges in the MST that are larger. The remaining connected components preserve transitive relations even when the distance is not transitive: if $\text{dist}(a_i, a_j)$ is low and $\text{dist}(a_j, a_k)$ is low, $\text{dist}(a_i, a_k)$ can still be high, but one can traverse from a_i to a_k with short hops via a_j. This approach preserves the diversity of releases over the careers of artists. If no edge is larger than θ_{dist}, then the MST connects all releases with paths of short hops, and we assume that they are all correctly attributed to the same artist.

3.2 Discography Deduplication

The goal of deduplication is to merge existing discographies, or sub-discographies split out from misattribution detection, that belong to the same artist (e.g. release a_3 in Fig. 1). Deduplication consists of two steps: (1) generating candidates for deduplication through a blocking strategy, and (2) a prediction step that determines whether the pair of discographies belong to the same real-world artist.

3.2.1 Blocking

To reduce the comparisons between pairs of discographies while maintaining high recall, we want to create small *blocks* of discographies that could belong to the same artist. One way is to simply take homonym artist discographies as a block; however, errors which lead to misattribution and duplication in music catalogs are often associated with varied spellings or aliases of the same real-world artist. Therefore, we need a more robust blocking strategy.

We build an Elasticsearch [23] index of all artist names in the catalog which we use to match and rank deduplication candidates. The matching strategy combines three conditions: (1) n-grams with $n = 2, 3, 4$; (2) fuzzy string matching with edit distance $\leq$ 2; and (3) normalised string matching without spaces and stop-words. If one or more of these conditions match a *seed* discography artist name, Elasticsearch returns a list of all matching candidates ranked by their *elastic score* [24]. We evaluate this strategy on a dataset of source and target artist name pairs from the *Merges* dataset (described below), and obtain a recall@10 of 97%.

3.2.2 Duplicate detection model

We train a Random Forest classifier to compute the similarity $\text{sim}(\mathcal{A}_i, \mathcal{A}_j) \in (0, 1]$ between pairs of artist discographies within each block. A high similarity score means that the two discographies are likely to come from the same real-world artist and should be merged, while a low score indicates that they are from different artists and should remain separate.

Data. The training data consists of $\sim$224K discography pairs. This data, which we call the *Merges* dataset, contains historical corrections of duplicate artist discographies. We assign a positive label to each merged pair and generate up to 10 negative examples for each positive one using the blocking strategy. During training we balance the data by applying a weight to each sample to be inversely proportional to its class frequency.

Attribute	Functions
<u>Elasticsearch relevance score</u>	See [24]
<u>Artist name similarity</u>	2-gram Dice coefficient
<u>Release Names</u>	Jaccard similarity
<u>Release Track Names</u>	Jaccard similarity
<u>Release Artists</u>	Overlap between artist names of collaborators on releases
<u>Release Track Artists</u>	Overlap between artist names of collaborators on release tracks
<u>Number of releases</u>	$\lvert\mathcal{A}_i \bigcup \mathcal{A}_j\rvert$
<u>Track Audio Vectors</u>	Mean Cosine Similarity

Table 2: **Duplicate Discography Detection Model Inputs.** Features above the line are metadata, and below are audio-based. Random permutations of <u>underlined</u> feature values decreased test-set performance >95% of the time.

Model Features. As in the misattribution model, we combine engineered metadata features with acoustic embeddings (see Table 2). Duplicate entity detection systems typically rely heavily on string similarity, but there are some challenges. For example, consider merging the discography referencing the artist *Prince*, with the one referencing his alias *Prince of Funk*, while remaining distinct from another artist called *Princess*. Relying solely on string similarity would suggest that the discographies from *Prince* and *Princess* are more likely to belong to the same artist than the ones from *Prince* and *Prince of Funk*. In this scenario, including audio representations in the model can improve performance in the absence of other distinctive features.

4. EVALUATION

We evaluate our system's performance with a series of experiments: First, we examine the offline performance of each sub-system under different feature ablations, including audio and metadata signals alone, using the *Relocations* and *Merges* datasets. Second, we conduct three experiments with Subject Matter Experts (SMEs) showing the performance "in the wild" of the misattribution and deduplication models, and their unification for the relocation of misattributed releases, as described in Fig. 1.

4.1 Audio and Metadata Feature Ablations

We test the hypothesis that metadata and learned audio representations model catalog correction tasks (i.e. misattribution and duplicate detection) better together than separately. Figure 3 shows the performance of the two models in three configurations: audio features only, metadata features only and combined. The features for each model and the distinction between audio-based and metadata-based features can be found in Tables 1 and 2. For each set of features, we separately tuned the hyperparameters with 5-fold cross-validation.

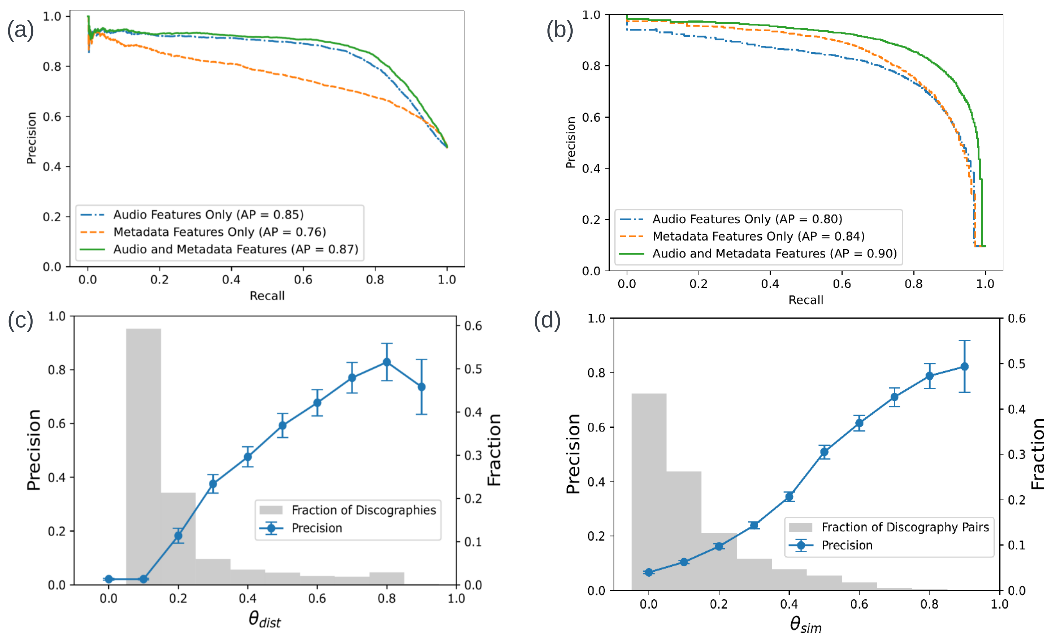


Figure 3: *(a) - (b)*: Precision-Recall curves in offline experiments with combinations of audio and metadata features for misattribution detection *(a)* and deduplication *(b)*. Average precision (AP) is reported in the legend for each set of features. *(c) - (d)*: Annotation experiment results for misattribution detection *(c)* and deduplication *(d)*. Precision is calculated for each threshold bucket and reweighed by the distribution of predictions shown on the second y axis.

Figure 3a shows that the pairwise misattribution model using audio-based features alone has good performance, but combining both audio and metadata produces the best performance. The full model has an average precision (AP) increase of 10.69% over the metadata-only model, and 1.95% AP over the audio-based model. These improvements come from a reduction in false positives (e.g. when the sound is not similar, but metadata similarities exist between two releases). For example, the test data contains the releases *SHOOT MY SHOT* and *Hurts Like Hell (feat. Offset)* from the American rapper *Offset*. The audio-only model predicts these releases come from different artists (their distance is 0.77). The full model gives the pair a distance of 0.1 because *"Kiari Kendrell Cephus"* (which is Offset's real name), appears in the credits of both releases as a writer and a composer/lyricist.

Figure 3b shows the performance of the duplicate detection task under the different ablations. Using metadata features alone outperforms audio features alone by 4% in AP. This is not surprising, as entity resolution tasks are usually heavily based on string similarity across aligned fields. Here too we can achieve good performance with metadata based features alone, but combining the features boosts AP by 6%. This boost is driven by cases where metadata features are insufficient. In the example of the *Prince* and *Prince of Funk* discographies, in the absence of shared collaborators or similarity on release titles we would get a false negative. However, the acoustic similarity between the two discographies is high, which allows us to correctly identify them as by the same real-world artist.

4.2 Experiments with SMEs

We conducted three experiments with SMEs to understand the performance of each task independently, and of the entire correction system (Fig. 1) in the context of its intended use, for a range of decision thresholds. We use precision as our evaluation metric since we want to reduce human effort spent reviewing and correcting the catalog.

4.2.1 Misattribution Detection

We ran the misattribution detection method from Sec. 3.1 using an early version of the pairwise model that was ready when the SMEs were available. The difference between the full model and this early version is that the latter uses only subset of the features of the full model (marked with * in Table 1). We selected a subset of artist discographies from the Spotify catalog, biased toward more popular artists, that reviewers are able to cross-reference externally. Then, we randomly sampled a pair of releases from each artist and calculated the value of the threshold θ_{dist} that would split the pair into two different partitions of the discography. This value is the largest edge weight along the path connecting the releases in the MST of the artist's discography. In the example in Fig 2c, the threshold between releases a_1 and a_3 would be $\theta_{\text{dist}} = 0.85$. We stratified our sample by these bucketed threshold values in 10 equally sized bins between 0 and 1, with a maximum of 100 pairs per bucket. The sampling produces $\sim$1K pairs, each of which was reviewed by a SME who classified it as *"by the same artist"* or *"by different artists"*. Figure 3c shows the precision for each value of θ_{dist} (blue line, left y-axis). For example, at a $\theta_{\text{dist}} > 0.7$, we can achieve 77%

precision. When θ_{dist} is small, many single-artist discographies are split into more than one group. This lowers precision but increases the fraction of artists that would have their discography partitioned into more than one group at each threshold (grey bars, right y-axis).

4.2.2 Duplicate Detection

To evaluate the duplicate detection model from Sec. 3.2, we generated a list of 140K seed artist discographies of popular artists from the catalog. Then, we generated 10 candidates for each seed artist using our blocking strategy to form artist-candidate pairs. For each pair we compute $\text{sim}(a_i, a_j)$, and bucket the scores in the same way as for the misattribution detection task above, sampling up to 100 per bin. For this task, 3 SMEs reviewed each sample and answered the question: *Do the two discographies belong to the same real-world artist?* We aggregated the annotations per sample to reflect the majority vote (i.e. at least 2 out of 3 of the annotators agree) and got 94% agreement. The remaining 6% of cases are ambiguous, and were excluded from the analysis. These cases are interesting and give insight into edge cases for future iterations of the model. For example, when the discographies were related but not technically by the same artist, e.g. the *Thelonious Monk Quintet* and the *Thelonious Monk Quartet*.

As in the misattribution task, as the threshold θ_{sim} in Fig. 3d increases so does precision, but with fewer candidate pairs (shown as grey bars, right y-axis). At a θ_{sim} > 0.7, we achieve 71% precision.

4.2.3 Predicted Relocation

Discography pairs that have been reviewed and determined to be duplicates can be merged in the catalog in a straightforward way. However, correcting misattributions is not so easy, and we still need to identify the correct discography in which they belong. Having validated both steps in our discography correction system, we can use the duplicate detection method to predict the correct discography (if any) for misattributed releases. To do this, we identify misattributions, using θ_{dist} > 0.7 based on the previous experiments, and we treat the misattributed releases as a sub-discography. Then, we generate and score candidate duplicate discographies for these sub-discographies using the deduplication model.

We evaluate performance on $\sim$1K release-discography pairs. Since the model generates up to ten predictions per seed, we take the highest predicted placement as a candidate for annotation. We asked SMEs to review the release and its predicted relocation and answer the question: *Does the release belong with the discography?*

Figure 4 shows the precision as a function of the two steps in the correction system θ_{dist} and θ_{sim}. The highest precision is 45%, which is achieved when both the misattribution step and deduplication (relocation) step have a high θ (top right corner of Fig. 4, representing 17% of the sample). The relocation task is more difficult and less precise because it inherits the uncertainty and performance of misattribution and duplicate detection. Additionally, we expect that a large number of misattributed releases might not

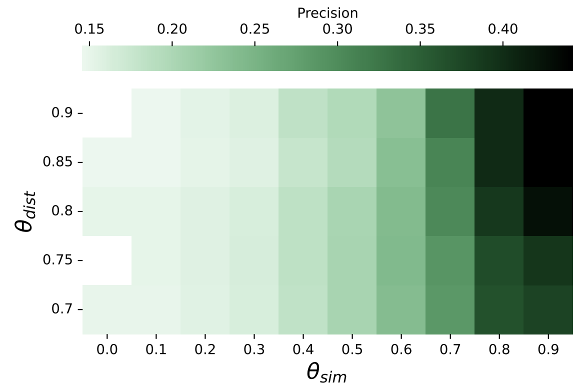


Figure 4: Precision of the combined system on the task of predicted relocation of misattributed releases for varying thresholds of the misattribution (θ_{dist}) and duplicate detection (θ_{sim}) methods.

belong anywhere, and will become standalone discographies. This means that even if the system considered this relocation to be the best out of ten candidates, a relocation might not be possible at all. Even in this scenario, the human effort to detect and correct misattributed content is significantly reduced.

5. DISCUSSION

We present a system designed for SMEs to maintain the correctness and completeness of artist discographies in a large online catalog. We demonstrate that leveraging both audio and metadata-based signals for misattribution detection and deduplication of discographies outperforms either in isolation. We validated each task separately, and the entire correction system across different thresholds, showing strong performance in three experiments with SMEs.

The power of this system is that it can scan a large catalog efficiently and direct the attention of human reviewers to where errors are most likely to be found, as well as suggest corrections for cases of misattribution and deduplication. This makes our system a key part of proactive catalog curation strategies. It is possible that some curation steps could be automated for high confidence predictions; however, due to the downstream impact of curation decisions (e.g. recommendations, search, user experience) the tolerance for incorrect relocations is low.

The current implementation of this system runs weekly, and the top-scoring candidates for misattibution, deduplication and predicted relocations are flagged for SMEs review. These reviews, in turn, become new labelled data on which the model can be re-trained and further improved.

Although discography errors are rare, it is important to minimise them as much as possible. Systems such as this are one tool among many that streaming platforms can use to ensure their catalog is correct, and to safeguard the experience of users and artists.

6. ACKNOWLEDGMENTS

We thank Nicola Montecchio and Glenn McDonald for discussions around the original idea for this work, Dimitrios Korkinof for valuable consultations throughout this project, Dana Puleo for assistance running experiments with SMEs, and Antonio Lima for comments on the manuscript.

7. REFERENCES

[1] A. A. Ferreira, M. A. Gonçalves, and A. H. Laender, "A brief survey of automatic methods for author name disambiguation," *Acm Sigmod Record*, vol. 41, no. 2, pp. 15–26, 2012.

[2] K. Wang, Z. Shen, C. Huang, C.-H. Wu, Y. Dong, and A. Kanakia, "Microsoft academic graph: When experts are not enough," *Quantitative Science Studies*, vol. 1, no. 1, pp. 396–413, 2020.

[3] Y. Qian, Y. Hu, J. Cui, Q. Zheng, and Z. Nie, "Combining machine learning and human judgment in author disambiguation," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011, pp. 1241–1246.

[4] H. Lee, P. Pham, Y. Largman, and A. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," *Advances in neural information processing systems*, vol. 22, 2009.

[5] A. Brinkman, D. Shanahan, and C. Sapp, "Musical stylometry, machine learning and attribution studies: A semi-supervised approach to the works of josquin," in *Proceedings of the Biennial International Conference on Music Perception and Cognition*, 2016, pp. 91–97.

[6] Z. Nasrullah and Y. Zhao, "Music artist classification with convolutional recurrent neural networks," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[7] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," *Advances in neural information processing systems*, vol. 26, 2013.

[8] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Moving beyond feature design: Deep architectures and automatic feature learning in music informatics." in *International Society for Music Information Retrieval Conference*. Citeseer, 2012, pp. 403–408.

[9] J. Royo-Letelier, R. Hennequin, V.-A. Tran, and M. Moussallam, "Disambiguating music artists at scale with audio metric learning," *International Society for Music Information Retrieval Conference*, 2018.

[10] J. Park, J. Lee, J. Park, J.-W. Ha, and J. Nam, "Representation learning of music using artist labels," *International Society for Music Information Retrieval Conference*, 2018.

[11] K. Kim, S. Rohatgi, and C. L. Giles, "Hybrid deep pairwise classification for author name disambiguation," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2369–2372.

[12] Q. Zhou, W. Chen, W. Wang, J. Xu, and L. Zhao, "Multiple features driven author name disambiguation," in *2021 IEEE International Conference on Web Services (ICWS)*. IEEE, 2021, pp. 506–515.

[13] G. Papadakis, D. Skoutas, E. Thanos, and T. Palpanas, "A survey of blocking and filtering techniques for entity resolution," *arXiv preprint arXiv:1905.06167*, 2019.

[14] P. V. Konda, *Magellan: Toward building entity matching management systems*. The University of Wisconsin-Madison, 2018.

[15] "Musicbrainz," musicbrainz.org, accessed: 2023-04-14.

[16] "Virtual international authority file," viaf.org, accessed: 2023-07-03.

[17] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

[18] "International standard name identifier," isni.org, accessed: 2023-07-03.

[19] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical society*, vol. 7, no. 1, pp. 48–50, 1956.

[20] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[21] "Spotify for developers: Get track's audio features," developer.spotify.com/documentation/web-api/reference/get-audio-features, accessed: 2023-07-03.

[22] G. Kondrak, "N-gram similarity and distance," in *International symposium on string processing and information retrieval*. Springer, 2005, pp. 115–126.

[23] "Elasticsearch," https://www.elastic.co/, accessed: 2023-07-06.

[24] "Lucene's practical scoring function," www.elastic.co/guide/en/elasticsearch/guide/current/practical-scoring-function.html, accessed: 2023-04-14.