Music Version Retrieval from YouTube: How to Formulate Effective Search Queries?

Simon Hachmeier, Robert Jäschke and Hadi Saadatdoorabi

L3S Research Center, Hanover, Germany

School of Library and Information Science, Humboldt-Universität zu Berlin, Berlin, Germany

Abstract

Various versions of musical works are published on YouTube, such as remixes or reaction videos. While some research has focused on tasks like audio-based version identification of these videos, it is still unclear how to effectively retrieve a large amount of relevant versions with textual queries. In this paper, we formulate search queries with YouTube search suggestions, evaluate these based on multiple dimensions and compute optimal ranks of queries on work-level. We show that queries containing the artist string retrieve results with higher relevance, but have higher overlaps. Additionally, we demonstrate that the amount of reasonable queries can be increased by applying frequently suggested expansions to works which tend to contextualize queries to the music domain.

Keywords

query formulation, music on youtube, audio based version identification

1. Introduction

In the context of western popular music, musical *works* correspond to cliques of *versions*.[1] A work is instantiated by a live performance or recording of an artist, which we will refer to as *original version*. Further versions can be instantiated in various ways[2]. Versions essentially have an m-to-n relationship (e.g., *medley*) and can be represented in a multimodal way (e.g., metadata, audio, music sheet). On the online video platform YouTube versions can be instantiated not only by property right owners, but also by others parties such as hobby musicians. This motivates a need for property right owners to find means to effectively find these versions on large scale.

Since YouTube does not provide a functionality to retrieve content related to specific musical works, one could exert online platforms listing versions of music work entities on YouTube,[3] and magazines.[4] However, these seem to aim for high quality content and contain rather official versions associated with professional or semi-professional musical artists.[5] Maximizing the

✉ hachmeier@l3s.de (S. Hachmeier); jaeschke@l3s.de (R. Jäschke); saadatdoorabi@l3s.de (H. Saadatdoorabi)

[1]Smith et al. [1] refer to the term as *derivative work*. SHS uses the term *performance*.

[2]Serrà [2] lists some examples of versions, such as *remix* or *demo*, but other types are thinkable and multiple types can apply to single version (e.g., *remix live*).

[3]E.g., Secondhandsongs (SHS) (https://secondhandsongs.com/) or cover.info (https://cover.info/).

[4]Articles by Mental Floss (https://www.mentalfloss.com/article/20811/most-covered-songs-in-music-history) and Stacker (https://stacker.com/stories/3975/most-covered-songs-all-time) list different covers.

[5]For instance, some works contain a lot of versions but not many *web covers*: "House of the rising sun" (https://secondhandsongs.com/work/44942/web-covers) and "Nothing else matters" (https://secondhandsongs.com/work/

retrieved versions therefore requires querying YouTube directly.

The identity of a musical work arises from its musical content. Unlike other services like Shazam,[6] YouTube does not provide an interface to external users to query its database by audio content explicitly. Implicitly, this is realized with YouTube's content ID system,[7] but it is not freely accessible. Moreover, the system seems to exhibit a problem of false positives which is particularly disruptive to the uploaders as shown in numerous studies [3, 4, 5]. It is also questionable how well it scales to different kinds of versions and types of versions. Alternatively, one can formulate text queries with the artist and title strings. While this presumably retrieves relevant videos, it is unclear how well the queries are contextualized. For instance, only querying for "hurt" by Nine Inch Nails might be too general leading to videos not related to the music domain. Querying for "nine inch nails hurt" instead might be too specific targeted towards videos with performances by the initial artist. Query expansions like "reaction", "live" or "cover" can be used to contextualize to the music domain without the necessity to contextualize to the artist. This results in a set of queries which can be formulated on the work level. In this paper, we leverage the knowledge captured within YouTube by using its search suggestion service to formulate sets of expanded queries on the work level. We evaluate these queries individually and compute their near-optimal ranks on the work level to evaluate them in context of their work-level sets.

Our means of evaluation are three-fold: We evaluate by matching against occurrences of YouTube URLs on the platform Secondhandsongs (SHS), by musical similarity computed by an audio-based version identification model and by manual annotation. We provide our dataset publicly.[8]

Our results provide insights into the quality of expansions which can be applied to web crawls to retrieve versions. Property right owners, collecting societies and artists could apply these to find versions of interest. In addition, music researchers could be supported to effectively generate new datasets. Our research is targeted towards the following research questions:

RQ1 How do queries with expansions retrieved on work-level compare to queries with frequent expansions in retrieval relevance?

RQ2 How to (re)order the respective queries to most efficiently retrieve relevant versions?

We first introduce the terms *work* and *version* in the context of music information retrieval. Then we outline related work before presenting our query formulation and expansion approach in Section 3 and evaluation setup in Section 4. We present our results in Section 5.

2. Related Work

Music on YouTube A classification approach by Agrawal and Sureka [6] aims for copyright violation detection of music and considers text similarities of work, video, artist and channel title strings. Similarities are determined before filtering out non-violations. Another approach

---

430/web-covers).

by Smith et al. [1] aims at detecting music versions and subsequently classifying their version type. In contrast to our work, both approaches use a fixed set of queries per work.

Audio-Based Version Identification   Audio-based version identification (VI) aims at automatically identifying whether two audio-based representations contain versions of the same musical work. Accuracy and scalability motivate recent research efforts to rely mainly on metric learning approaches which learn to model similarities between representations, such as pitch class profiles (PCP) by distance functions [7, 8, 9, 10, 11]. In this paper we use the MOVE model by Yesiler et al. [11] as a query evaluation tool. It is based on a multi-layer convolutional network architecture with a multi-channel adaptive attention mechanism to summarize temporal content. It processes the PCP variant named CREMA-PCP[9] by McFee and Bello [12] to compute embeddings which model the musical dissimilarity by Euclidean distance.

Query Expansion   Most recent approaches in query expansion research rely on language modeling [13, 14, 15], external knowledge bases [16, 17] or query logs [18, 19]. YouTube search suggestions[10] rely on prior searches of the authenticated user profile and searches by other users including trends. Thus, these are mainly based on user-logs but might also incorporate knowledge from external sources and apply language modeling methods to capture semantic similarities.

3. Music Version Retrieval from YouTube

3.1. Problem Definition

We assume that a work W is realized in a set of different versions on YouTube $V = \{v_1, v_2, \ldots, v_N\}$ where the actual number of versions N for each W is unknown. Videos on YouTube are not organised in terms of versions and therefore we have to facilitate the relationship between versions and videos to be m-to-n relationships. We limit our objective to maximizing the number of videos we can find that contain relevant versions even if they contain irrelevant content that is non-musical (e.g., interviews, comments, cheering) or (also) related to other works (e.g., medleys, concert videos). Because YouTube does not provide direct access to query videos by audio, we rely on YouTube as a black box which can be accessed via text-based queries. This way we further leverage its internal knowledge about the versions. For each work we formulate a set of queries $S = \{Q_1, Q_2, \ldots, Q_K\}$. These are expected to return relevant results. Since multiple queries are formulated on work level returning one result set each, this problem can be understood as a set cover problem.

3.2. Base Queries

Given a work from our seed dataset, we use the original version as a metadata representation to extract artist and title strings. Accordingly, we formulate two types of queries: solely the title string and the artist and title string concatenated by a space character (e.g., "led zeppelin

---

[9]CREMA stands for *convolutional and recurrent estimators for music analysis.*
[10]https://support.google.com/youtube/answer/9872296?hl=en

Figure 1: A screenshot of the expansions suggested by YouTube for the query "led zeppelin kashmir". The expansions highlighted in green are examples of universal ones since they occurred with high frequency in the whole set. "guitar lesson" occurred frequently, but was not among the top 30 and the other expansions including "egyptian orchestra" seem to be targeted towards a specific version of the work.

kashmir" or "adele hello"). These serve as base queries that are expanded in a later step. Thus, for each work W_i we instantiate a corresponding *title* query Q_i^τ and a combined *artist+title* query Q_i^α.

3.3. Expansions

Query expansion is a procedure to reformulate queries aiming for an improved information retrieval effectiveness in search engines. Given an input query Q, the reformulated query $\hat{Q}$ is defined as: $\hat{Q} := Q + T$ where T is an expansion string and $+$ represents string concatenation with a space character. It is also common to remove stop words from the query, which we do not, because our queries contain fixed titles. We propose two types of expansions: *individual* expansions that are specific for each work, and *universal* expansions that are independent of any specific work.

To find sets of effective individual expansions T to the base queries Q_i^τ and Q_i^α of W_i, we utilize the Google Search Suggestion Service and retrieve up to nine expansions T_{ij}^τ and T_{ij}^α for each of the base queries. One key advantage of using these expansion is the dependence on prior search requests by users[11] and hence the high probability of further relevant contextualization provided to the base query string. By this means, we expect to find expansions that might be targeted towards finding specific performances (e.g., "live aid 1985") while others might be generally applicable to all works (e.g., "cover", "live"). These might essentially correspond to version types or relevant entities, such as instruments for instance. Therefore we expect a contextualization to either the music domain, the work itself or some other possible unknown but relevant dimensions. Note that each expansion can consist of several terms joined with space as shown in Figure 1. Due to the dependency on availability of individual suggestions, we further want to find a set of generally applicable expansion terms. We did this by combining the individual expansions into the sets T^τ and T^α and ranked them by their frequency. As we will see in Section 5, the expansions T^τ are less generic and thus less useful than T^α. Therefore, we restricted the analysis to T^α. Specifically, we extracted the 30 most frequently suggested expansions from T^α, resulting in the set of universal expansions $T^U = \{T_1^U, T_2^U, \ldots, T_{30}^U\}$.

Combining each base query with its corresponding individual expansion and with the universal expansion results in the following four types of expanded queries:

---

[11]c.f. https://support.google.com/youtube/answer/9872296

- *individual title:* $\hat{Q}_{ij}^{\tau+I} := Q_i^{\tau} + T_{ij}^{\tau}$
- *individual artist+title:* $\hat{Q}_{ij}^{\alpha+I} := Q_i^{\alpha} + T_{ij}^{\alpha}$
- *universal title:* $\hat{Q}_{ij}^{\tau+U} := Q_i^{\tau} + T_j^{U}$
- *universal artist+title:* $\hat{Q}_{ij}^{\alpha+U} := Q_i^{\alpha} + T_j^{U}$

Due to the varying number of individual expansions and the potential match between individual and universal expansions for each work, the number of produced queries varies among the works. Each query Q has a corresponding result set R consisting of videos: $R = \{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_M\}$ which can be understood as candidate versions. We need to assign a score or binary label to indicate the relevance of the video which we describe in Section 4 which enables the query result relevance evaluation.

3.4. Near-Optimal Rank Computation

We compute the near-optimal ranks of queries per work by a greedy algorithm inspired by Zhai et al. [20]. At each iteration, the query with the highest value of the remaining unranked queries is ranked next. Our value function takes into account the increase of relevance by the aggregation of the inverse of the mean MOVE-based distances and the increase of novelty measured in new videos. The value for each result set R_i with respect to the result sets of the queries ranked before $L = \{R_1, \ldots, R_{i-1}\}$ is computed as follows:

$$\text{value}(R_i, L) := \alpha \cdot \text{rel}(R_i, L) + (1 - \alpha) \cdot \text{nov}(R_i, L), \tag{1}$$

where

$$\text{nov}(R_i, L) := \frac{|R_i \setminus L|}{|L|} \tag{2}$$

and

$$\text{rel}(R_i, L) := \frac{\frac{1}{|R_i \setminus L|} \sum_{\hat{v} \in R_i \setminus L} \frac{1}{\text{MeanDist}(\hat{v}, V)}}{\frac{1}{|L|} \sum_{\hat{u} \in L} \frac{1}{\text{MeanDist}(\hat{u}, V)}} \tag{3}$$

where α is an adjustable hyperparameter set to 0.5 to equally prioritize the number of new videos and the inverse of the MOVE-based distances which models musical similarity. $\text{MeanDist}(\hat{v}, V)$ is the mean of the MOVE-based distances between the candidate $\hat{v}$ and k sampled example versions in V.

4. Evaluation

In the following, we give insights about our dataset used, some aspects about the implementation and the different types of evaluations.

4.1. Dataset & Implementation Details

Our seed dataset is a subset of the Da-Tacos benchmark dataset by Yesiler et al. [21] where each work is represented by 13 different versions with a variety of different audio features. Our subset is constrained to works with an original version flag on SHS. It consists of 983 works from the SHS database with a mean of 96 and a median of 77 performances per work. 95% of the declared original works have lyrics in English language and the remaining 5% are in French, Hebrew, Portuguese and Spanish. We extracted all performances for the works in our dataset to obtain metadata representations (title, artist, identifier, YouTube ID, flag about the original property) from SHS using the official API.[12] These representations were used as a seed set for base query formulation. We retrieved the YouTube suggestions with the Google Suggestion URL with the parameters set to YouTube and Firefox respectively on machines located in Germany.[13] The service returns up to nine suggestions and no user authentication is required. Since user-specific suggestions only apply to authenticated users as outlined in Section 2 we do not expect user-specific bias. Trending expansions based on the geographic location can still occur. Since the returned list contains expansions including the requested query string, we removed the query strings to be able to store the expansions as individual entities and allow for aggregation counts and application of expansions on other query strings. We processed 66,589 requests in total which corresponds to roughly 68 queries per work and limited each result set to 100 result videos.[14] As reported in Table 1, we downloaded the audio data for a total of around 648,714 videos with a sampling rate of 44.1 kHz. These cover the found videos for 295 works, excluding videos which were not downloaded due to unavailability. We also omitted downloading videos with a length of more than 10 minutes due to capacity constraints.

We use the MOVE default parameters: an embedding dimension of 16,000, autopool summarization and a final linear layer with batch normalization and normalized Euclidean distances by the embedding dimension in the evaluation process. Apart from the datasets in Table 1, we matched the YouTube IDs of these processed audio files with the metadata retrieved from SHS of all the versions in our dataset and found around 12,597 matches for 868 works.[15] These matches were used to generate another evaluation dataset. We used the matches mapping to their work ID as positive examples and randomly sampled an unrelated work ID of the remaining 867 works to generate negative examples. This dataset was exclusively used to evaluate the MOVE model as a binary classifier.

4.2. Query Result Relevance

To evaluate queries regarding their retrieval relevance we rely on three approaches to determine the relevance of videos in relation to the works they were retrieved for:

---

[12]https://secondhandsongs.com/page/API

[13]https://suggestqueries.google.com/complete/search?client=firefox&ds=yt&q=QUERY

[14]We used the YouTube search Python API by Hitesh Kumar Saini, cf. https://pypi.org/project/youtube-search-python/.

[15]Please note that this number of works is higher than the ones reported in Table 1, due to the exclusion of works in the MOVE-based evaluation for which we could not download all the video results. Additionally, some result videos matched other works of the set that we did not intent to download.

Table 1
Basic statistics of the query evaluation datasets.

	Seed	MOVE-based	Manual
works	983	295	108
distinct expansions	3,680	1,314	124
queries	66,589	19,591	124
result videos	1,993,759	648,714	116

SHS Matching: Since our seed dataset is entirely based on SHS, we can assign binary labels indicating matches of retrieved YouTube IDs in the result sets with YouTube IDs from the SHS metadata.

Manual: We randomly sampled 116 candidate videos stratified along the dimensions of the video result page, the base query and query expansion type. Six evaluators received each 62 pairs of candidate video URLs with URLs from the dataset seed and were asked to define the relationship between the videos by a fixed dropdown of four possible options for selection. These included two indicating an existing version relationship,[16] one stating otherwise and one for uncertainty. Each pair was evaluated by three evaluators and we label each pair as positive if at least two voted for a relationship. These labels were used for the MOVE model evaluation and the query relevance evaluation.

MOVE-Based: We use the VI model MOVE and compute the mean MOVE-based distance of the candidate video $\hat{v}_i$ to multiple example versions of the work queried for.

The labeled result sets of candidates per query are then used to evaluate the query result relevance along multiple dimensions, such as the base query, expansions and expansion types as well as the computation of optimal ranks. Due to the utilization of the MOVE model as a measurement instrument, we perform another evaluation specifically for the model.

4.3. MOVE Model Evaluation

We wanted to use multiple example versions per work to determine the relevance of videos to compensate version-specific musical bias in the evaluation. Therefore, we had to find an appropriate number of k example versions of the respective work to compare with the candidate when computing the distance as well as an aggregation function. We used the manually labeled dataset and the binary labels by SHS matches and processed CREMA-PCP files to evaluate the model as a binary classifier with four different tested thresholds applied to the Euclidean distance outputs and k and the aggregation function[17] as a hyperparameter. For each of the combinations of these hyperparameters we ran 10 iterations of which we report the mean of F1 in Section 5.

---

[16] One label indicates that a version is contained in the candidate and the other that the candidate is a original version. This was for instance relevant in cases, where the candidate matched the seed dataset entry.

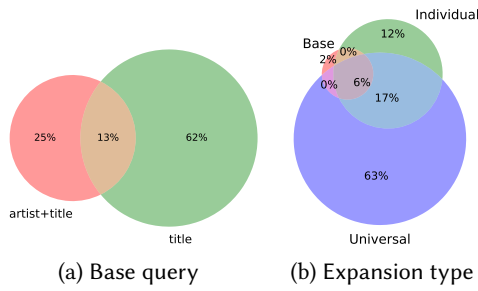[17] We tested with mean, median and maximum and minimum

(a) Base query   (b) Expansion type
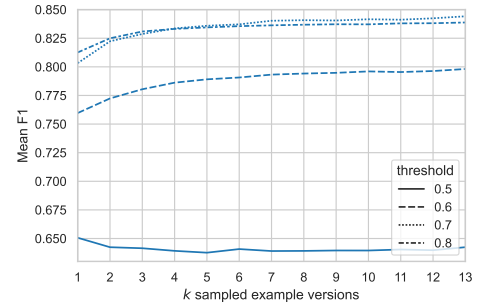
Figure 2: Result set overlaps.

Figure 3: MOVE evaluation results.

5. Results

5.1. SHS-Based Query Result Relevance

The retrieval relevance results per query type are generally rather lower in the SHS-based evaluation. Base queries yield a precision of 0.05 and a recall of 0.06 on average. Individual queries undershoot this with a mean precision of around 0.02 and a mean recall of 0.03 per query. Both of these measures are 0.02 for universal queries. The work-level maximum precision and recall per query are 0.13 and 0.12. We argue that these low numbers are mainly caused due to the incompleteness of versions documented on SHS since they are based on manual evaluation processes. In the following we substantiate our argumentation about higher numbers of versions on YouTube than on SHS by our MOVE model and manual evaluation.

5.2. MOVE Model Results

In Figure 3 we present the mean F1 per threshold as a function of k sampled example versions with the mean as aggregation function which performed best. We decided to set $k = 6$ for the subsequent query relevance evaluation, to balance capacity constraints and evaluation performance. We apply these hyperparameters to MOVE and evaluate it as a binary classifier with the manually annotated dataset which yields a precision of 0.76, a recall of 0.79 and an F1 of 0.78.

5.3. Seed Dataset Expansion Frequencies

Table 2 lists the ten most frequently suggested expansions for each of the two base query types. It can be seen that some of these expansions match version types (e.g., "remix", "instrumental") and instruments which is expected but favorable since they are generally applicable. Overall, there are a lot more distinct expansions for *title* queries (2,847) than for *artist+title* queries (833). Consequently, the fraction of works with no individual suggestions is also much higher for *artist+title* queries (54%) than for *title* queries (4%) which seems to explain the higher counts for *title* expansions. A reason for this could be that *artist+title* queries are longer, leading the suggestion algorithm to interpret the input query as saturated.

Table 2
Most frequent expansions generated from the seed dataset.

rank	*artist+title*	count	*title*	count
1	lyrics	288	karaoke	464
2	live	276	lyrics	442
3	cover	196	piano	353
4	karaoke	187	cover	334
5	remix	141	remix	176
6	reaction	139	live	160
7	piano	97	guitar	141
8	instrumental	94	backing track	119
9	guitar lesson	57	frank sinatra	101
10	guitar	56	ella fitzgerald	89

Beside this difference in numbers of expansions, we also realized that the *title* expansions often matched artist strings contained in the dataset (e.g., "frank sinatra", "ella fitzgerald"), possibly because the provided base queries did not contain the artist string. Since these expansions might contextualize only for specific works within the dataset and would therefore induce a bias when expanding base queries of works not found in the respective subsets, we argue that *title* expansions are generally less useful than expansions based on *artist+title* queries in the context of general music version retrieval. Thus we used the *artist+title* expansions as universal expansions limited to the top 30.

5.4. Result Set Overlaps

In Figure 2 we present the overlaps of the result sets of the base query and expansion type dimension. Striking are the higher amount of candidate videos retrieved by *title* base queries which make up around 62% and the high overlap of universal and individual queries. The sheer amount of universal queries also leads to around 46% which are solely retrieved by those. The overlaps based on these two dimensions motivate the evaluation of queries in context of their result set, which we do at the end of this section.

5.5. MOVE-based Query Result Relevance Evaluation

Work-Based We present the median MOVE-based distances per work in Figure 4. The apparent variance per work also encourages the use of investigation of some work-specific properties and their potential impact on query relevance performance. We therefore computed the Spearman's rank correlation coefficient ρ for the following work properties in relation to the median MOVE-based distance per work. We can report a weak correlation in the number of words in the artist string (ρ=0.21, p<0.01) and the days published since the initial release (ρ=0.27, p<0.01) with significance. Additionally, a negative correlation of medium strength of the YouTube viewcount of the original version can also be measured (ρ=-0.31, p<0.01). We cannot report a correlation for the number of words in the title string (ρ=-0.04, p=0.42).
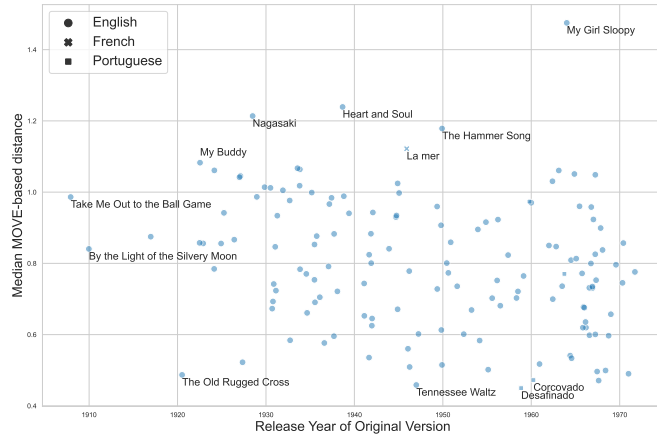
Figure 5: Accumulated gains of optimal ranks.

Figure 4: Median MOVE-based distances on work level.
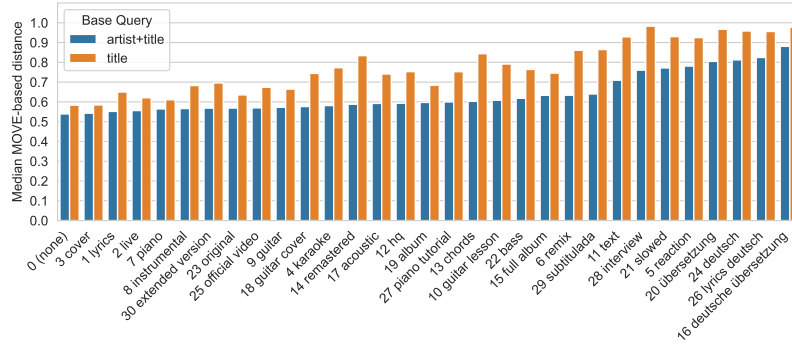

Figure 6: Median MOVE-based distances of result sets of universal expansions with their initial rank by suggested frequency in the seed dataset.

Universal Expansions We evaluate the universal expansions specifically, since these were used for all the works in the evaluation set. In Figure 6 we report the median MOVE-based distances of queries with the universal expansion terms and the sole base queries as baselines. Generally, the *artist+title* queries seem to perform better. It can also be seen that the three most frequently suggested expansions are also among the best performing expansions by relevance. However, there are also some strong shifts in ranks visible (e.g., "extended version", "original", "reaction"). The four weakest performing expansions are all in German language. Next, we compare the performance of these expansions with the individual ones.

Base Query and Expansion Type We compare the retrieval relevance by MOVE-based distances per query type in Table 3. In the group of universal result sets we only consider sets for works where the universal expansion did not match an individual one, since we essentially want to evaluate how non-suggested expansions perform compared to suggested ones. Interestingly, the individual queries perform even better than the sole base queries in the MOVE-based

Table 3
Query type evaluation based on the median MOVE-based distances and binary labels by human annotation.

| Query Type | Base Query | MOVE-based distance | | | Annotated | |
		Mean	Median	Support	Precision	Support
Base	*artist+title*	0.58	0.56	291	**0.83**	12
	title	0.63	0.58	295	0.64	22
Individual	*artist+title*	**0.54**	**0.51**	855	0.68	19
	title	0.70	0.63	2204	0.48	21
Universal	*artist+title*	0.67	0.65	7756	0.56	18
	title	0.80	0.84	8163	0.42	26

evaluation and second best according to the manual labels. The achieved performance is comparable to the top universal expansion presented before. The superior performance of *artist+title* queries is apparent again. The universal queries on average perform generally weaker, but are also supported by a far higher amount of result sets. We also checked the videos which were positively labeled by the evaluators and only found two matches with SHS metadata out of a total of 26 videos with full agreement of the evaluators. This validates our point about the limited amount of versions on SHS to some extent. Besides our sole evaluation of the query dimensions individually, we now want to evaluate them in the context of their query sets per work.

Near-Optimal Query Ranks   In Figure 5 we show the mean accumulated gains per query index of the near-optimally ranked queries per work. As expected, the ratio of individual terms is slightly higher at the earlier indices, since they have a high retrieval performance. It is also visible, that the accumulated unique videos are saturated faster than the inverse of MOVE-based distances. In Figure 7 we show boxplots per universal expansion indicating how their respective queries tend to to be ranked within the set of all the queries. Interestingly, *title* queries are generally ranked higher in spite of their lower performance in the prior results. Furthermore, some specific expansions are generally ranked higher, such as "karaoke", "slowed" and "remastered" indicated by the shorter inter-quartil-range. These terms are not among the top universal terms. Overall it must be considered that the whiskers are still rather broad for the majority of the universal terms.

6. Conclusion and Future Work

We showed that we can leverage internal knowledge captured within YouTube to generate effective search queries to retrieve music versions. Addressing **RQ1** our results reveal that sole base queries and individual expansion terms have a higher retrieval performance but are, depending on the work, just available in a limited amount. To scale up the number of queries, universal expansions based on global suggestion frequency can be applied. In this regard, the order of queries on work-level is important as well where we can demonstrate that *title* queries are generally ranked higher since there result sets have less overlaps. Furthermore, the
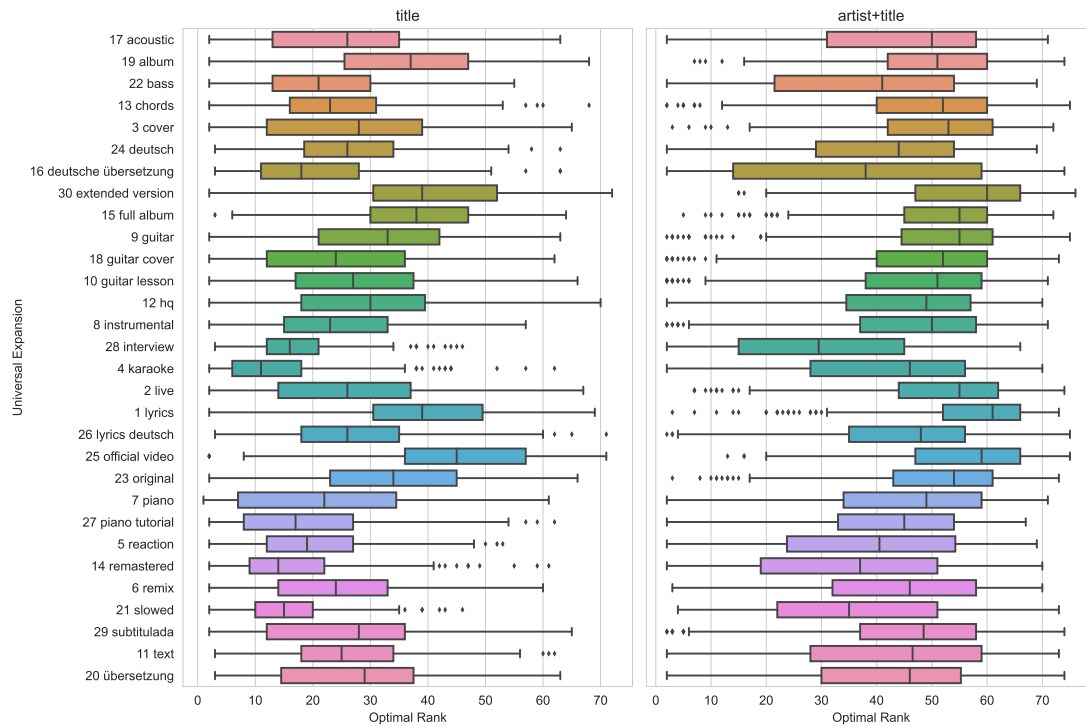
Figure 7: Optimal Rank Boxplot of universal expansions.

performance of some universal expansions appears to be better when considered in a sequence of queries than when considered in isolation. With regard to **RQ2**: A general strategy to query YouTube for musical works might therefore incorporate first querying by base queries and individual queries with the potential upscaling by using universal queries with *title* queries first. However, the retrieval process might depend highly on the work and its age, initial artist name length or popularity could be influencing factors. Worth mentioning are some limitations of our work. Firstly, the SHS-based and manual evaluation are just limited in terms of labeled candidates. The MOVE-based evaluation addresses this issue but the MOVE model might suffer from specific bias leading to an underestimation towards video version types like reactions or remixes where the relevant sections of the works are underrepresented. Another limitation is the seed set itself, which mostly represents western popular music in English language from the 20th century with one artist name. Further research could therefore experiment with other datasets of other genres, languages and ages and use additional artist names per work.

References

[1] J. B. L. Smith, M. Hamasaki, M. Goto, Classifying derivative works with search, text, audio and video features, 2017 IEEE International Conference on Multimedia and Expo (ICME) (2017) 1422–1427.

[2] J. Serrà, Identification of versions of the same musical composition by processing audio descriptions, Ph.D. thesis, Universitat Pompeu Fabra, Barcelona, 2011. URL: http://hdl.handle.net/10803/22674.

[3] B. Boroughf, The next great youtube: improving content id to foster creativity, cooperation, and fair compensation, Alb. LJ Sci. & Tech. 25 (2015) 95.

[4] T. Lester, D. Pachamanova, The dilemma of false positives: Making content id algorithms more conducive to fostering innovative fair use in media creation, UCLA Ent. L. Rev. 24 (2017) 51.

[5] L. Zapata-Kim, Should youtube's content id be liable for misrepresentation under the digital millennium copyright act, BCL Rev. 57 (2016) 1847.

[6] S. Agrawal, A. Sureka, Copyright infringement detection of music videos on youtube by mining video and uploader meta-data, in: V. Bhatnagar, S. Srinivasa (Eds.), Big Data Analytics, Springer International Publishing, Cham, 2013, pp. 48–67.

[7] X. Qi, D. Yang, X. Chen, Triplet convolutional network for music version identification, in: K. Schoeffmann, T. H. Chalidabhongse, C. W. Ngo, S. Aramvith, N. E. O'Connor, Y.-S. Ho, M. Gabbouj, A. Elgammal (Eds.), MultiMedia Modeling, Springer International Publishing, Cham, 2018, pp. 544–555.

[8] C. Jiang, D. Yang, X. Chen, Similarity learning for cover song identification using cross-similarity matrices of multi-level deep sequences, in: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 26–30. doi:10.1109/ICASSP40776.2020.9053257.

[9] G. Doras, G. Peeters, Cover detection using dominant melody embeddings, in: ISMIR, 2019, pp. 107–114.

[10] C. Jiang, D. Yang, X. Chen, Similarity learning for cover song identification using cross-similarity matrices of multi-level deep sequences, in: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 26–30. doi:10.1109/ICASSP40776.2020.9053257.

[11] F. Yesiler, J. Serrà, E. Gómez, Accurate and scalable version identification using musically-motivated embeddings, in: Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), 2020.

[12] B. McFee, J. P. Bello, Structured training for large-vocabulary chord recognition., in: ISMIR, 2017, pp. 188–194.

[13] Z. Zheng, K. Hui, B. He, X. Han, L. Sun, A. Yates, BERT-QE: contextualized query expansion for document re-ranking, CoRR abs/2009.07258 (2020). URL: https://arxiv.org/abs/2009.07258. arXiv:2009.07258.

[14] I. S. Kaushik, G. Deepak, A. Santhanavijayan, Quantqueryexp : A novel strategic approach for query expansion based on quantum computing principles, Journal of Discrete Mathematical Sciences and Cryptography 23 (2020) 573–584. URL: https://doi.org/10.1080/09720529.2020.1729506. doi:10.1080/09720529.2020.1729506. arXiv:https://doi.org/10.1080/09720529.2020.1729506.

[15] M. Esposito, E. Damiano, A. Minutolo, G. De Pietro, H. Fujita, Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering, Information Sciences 514 (2020) 88–105. URL: https://www.sciencedirect.com/science/article/pii/S0020025519311107. doi:https://doi.org/10.1016/j.ins.2019.12.002.

[16] H. K. Azad, A. Deepak, A new approach for query expansion using wikipedia and wordnet, Information Sciences 492 (2019) 147–163. URL: https://www.sciencedirect.com/science/article/pii/S0020025519303263. doi:https://doi.org/10.1016/j.ins.2019.04.019.

[17] J. A. Nasir, I. Varlamis, S. Ishfaq, A knowledge-based semantic framework for query expansion, Information Processing Management 56 (2019) 1605–1617. URL: https://www.sciencedirect.com/science/article/pii/S030645731830339X. doi:https://doi.org/10.1016/j.ipm.2019.04.007.

[18] J. Gao, S. Xie, X. He, A. Ali, Learning lexicon models from search logs for query expansion, in: Proceedings of EMNLP, ACM, 2012. URL: https://www.microsoft.com/en-us/research/publication/learning-lexicon-models-from-search-logs-for-query-expansion/.

[19] K. Cao, C. Chen, S. Baltes, C. Treude, X. Chen, Automated query reformulation for efficient search based on query logs from stack overflow, in: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 2021, pp. 1273–1285. doi:10.1109/ICSE43902.2021.00116.

[20] C. Zhai, W. W. Cohen, J. Lafferty, Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval, SIGIR Forum 49 (2015) 2–9. URL: https://doi.org/10.1145/2795403.2795405. doi:10.1145/2795403.2795405.

[21] F. Yesiler, C. J. Tralie, A. A. Correya, D. F. Silva, P. Tovstogan, E. Gómez, X. Serra, Da-tacos: A dataset for cover song identification and understanding, in: ISMIR, 2019, pp. 327–334.