A Case Study Towards Segment-Aware Version Identification

Simon Hachmeier[0000−0003−4843−5196] and Robert Jäschke[0000−0003−3271−9653]

Berlin School of Library and Information Science, Humboldt-Universität zu Berlin,
Berlin, Germany
simon.hachmeier@hu-berlin.de, robert.jaeschke@hu-berlin.de

Abstract. Version identification (VI) or cover song identification is the task of automatically detecting whether musical tracks are originating from the same work. An important step in approaches for solving this task is the shingling along the time axis. Recently proposed models show a better retrieval accuracy when using shorter shingles (e.g., of 20 seconds) rather than relying on longer ones (e.g., more than 1 minute) or even full tracks. However, all current approaches define a fixed length for the shingles, even though the actual segments in a musical sense, such as the verse or chorus, are usually varying in length and might even vary between different versions of the same musical work. This case study explores new perspectives on VI beyond fixed-length shingles. Based on a new VI dataset with manually annotated segment labels, we investigate the distributions of pairwise distances of version embeddings obtained from a state-of-the-art VI model. We further examine the impact of different shingle-to-segment offsets to uncover the potential performance degradation in current VI testing methods.

Keywords: version identification · segment · shingle

1 Introduction

Musical versions – often referred to as cover songs – are renditions of an original musical work. Automatically detecting whether versions are derived from the same original work is referred to as *version identification* (VI). Recent approaches to solve that task are mostly based on representation learning, where an audio representation such as a constant-Q transform (CQT) is encoded into a learned representation of a fixed-length vector [2, 3, 6, 7, 9, 16].

An important detail in VI is the strategy which transforms the full-length audio track of a version into a variable number of shorter segments – so-called *shingles*. This enables the detection of versions even if only sub-segments (e.g., chorus, verse) of the original work are covered, which is not uncommon on online video platforms [5]. In an ideal scenario, one would match only the segments of interest, that is, only the chorus of one version against the chorus of another version. However, segmentation methods have not yet been studied in conjunction with VI, and instead, fixed-length segments (shingles) are used.

In this case study we aim to uncover the potential of using actual musical segments (e.g., chorus, verse) in VI rather than fixed-length shingles. We create a new dataset with 8 versions of 4 musical works (also called *cliques*) of Western popular music with a total of 136 segments and use CLEWS [9] as a strong baseline to obtain segment-wise VI embeddings. We then analyze the relation between the pairwise cosine similarities of embedded segments with their segment labels. Given the fixed-length segment strategy in current training and evaluation methods in VI, systems likely exhibit offsets between the boundaries of the actual segments and shingles. To investigate the potential impact of these offsets during inference, we simulate different shingle-to-segment offsets.

This paper is organized as follows: In the following section, we outline related work in the field of VI. In Section 3 we describe our dataset and then present our results in Section 4. Our case study closes with a conclusion in Section 5.

2 Related Work

Recent VI systems mostly rely on CQT as input representation in conjunction with representation learning by a contrastive loss (e.g., triplet loss) to obtain vector representations for which versions of the same work are closer (e.g., by cosine similarity) to one another than versions of different works. Earlier, the shingling was done with longer lengths (e.g., 40 seconds or more) [2, 6, 16]. More recently, shorter shingles have been considered, motivated by applications of VI on social media platforms. For example, ByteCover3 [3] is trained using 20-second shingles and evaluated on 30 second shingles. CoverHunter [7] follows a coarse-to-fine training scheme. In the fine stage, the shingles are between 15 and 45 seconds long and in the evaluation setup fixed to 45 seconds. CLEWS [9] achieves state-of-the-art performance in VI and was trained using supervised contrastive learning. The model was tested at different shingle lengths, with the best performance with 20 seconds. Due to its strong retrieval accuracy and its public availability,[1] we select this model for our case study.

Although the datasets used for training and evaluation VI models are rather large [1, 14, 15], none of the current datasets contains segment annotations. While these annotations are rather expensive to obtain, several datasets have been proposed targeting the task of music structure analysis [4, 8, 10]. The task deals with the segmentation of musical tracks into meaningful segments, such as chorus and verse and could therefore be beneficial for the task of VI.

3 Methodology

3.1 Dataset

To the best of our knowledge, there is no dataset which contains multiple versions per musical work (as required by VI) *and* segment annotations. For this reason,

[1] See `https://github.com/sony/clews`.

Table 1: Versions in VerSegD. For each work we collect one original (shown by the performing artist marked in **bold**) and another version.

Song Title	**Performing Artist**
Ain't No Sunshine	**Bill Withers** Michael Jackson
Bohemian Rhapsody	**Queen** The Braids
Smoke on the Water	**Deep Purple** Dave Rogers
Yesterday	**The Beatles** Ray Conniff & The Singers

Table 2: Segment labels in VerSegD.

Label	**Count**
Verse	33
Chorus	23
Riff	16
(Silence)	9
Outro	7
Intro	5
Bridge	5

we developed and here present the new **Ver**sion **Seg**ment **D**ataset (VerSegD). Given that CLEWS was trained and validated on the respective subsets of Discogs-VI-YT [1], we selected four popular cliques of its respective test subset to ensure that these were not seen by CLEWS during training. Furthermore, we selected cliques with originals that we were familiar with, to simplify the annotation process. For each of the selected cliques, we annotated two versions with function labels (see Table 2), similar to ones proposed in the SALAMI dataset [10]. We use Audacity[2] to listen to segment boundaries and annotate up to a temporal resolution of a tenth of a second.

We publicly provide the metadata (i.e., timestamps for function labels per version and YouTube identifiers) for our dataset.[3] While VerSegD contains only 8 versions, it comprises 136 total segments. The mean and median lengths of segments are 14.60 seconds and 13.95 seconds, respectively.

3.2 Analysis Design

Version-Segment Relationships We want to analyze different version-segment relationships in the context of VI. Our dataset comprises the set V of versions (music tracks) which are partitioned into C cliques, where one clique represents all versions of one work. Moreover, each version consists of multiple segments (e.g., verse, chorus), where S denotes the set of all segments and $v : S \to V$ maps each segment to its corresponding version, and $c : S \to C$ to the clique of its version. We define the following relationships between pairs of segments:

$$V^{=} := \{(s,t) \in S^2 \mid v(s) = v(t)\} \tag{1}$$

$$V^{+} := \{(s,t) \in S^2 \mid v(s) \neq v(t) \wedge c(s) = c(t)\} \tag{2}$$

$$V^{-} := \{(s,t) \in S^2 \mid c(s) \neq c(t)\} \tag{3}$$

[2] `https://www.audacityteam.org/`

[3] `https://github.com/progsi/VerSegD`

Similarly, we define pairwise relationships based on the segment label (with $l : S \to L$ mapping segments to their label, e.g., *chorus* or *intro*):[4]

$$L^{+} := \{(s,t) \in S^2 \mid s \neq t \wedge l(s) = l(t)\} \tag{4}$$

$$L^{-} := \{(s,t) \in S^2 \mid l(s) \neq l(t)\} \tag{5}$$

Combining the obtained sets, we model version-segment relationships. For these, we can define our expected outcomes with respect to the similarity of segments. For instance, we expect the version segments in $V^{+} \cap L^{+}$ to have a rather high similarity, because both resemble positive pairs in terms of VI shingles. In contrast, $V^{+} \cap L^{-}$ are positive pairs in the VI task, but could be considered negative due to the non-matching segments. Lastly, $V^{-} \cap L^{+}$ and $V^{-} \cap L^{-}$ are both strictly negative, since the compared versions are different. However, one could argue that the matching segment label in the former could influence the similarity, due to functional characteristics (e.g., higher loudness of the chorus compared to the verse). Such characteristics are exploited in some approaches in music structure analysis [11–13].

Impact of Offsets Current systems solely consider shingles of a fixed length. Thus, we perform a second analysis in which we compare an anchor segment s (with its length $|s|$ in seconds) against another segment $s_{\delta,w}$ for which we apply a fixed shingle length $w \in \{10, 20, 30\}$ (measured in seconds, corresponding to three of the values tested for CLEWS [9]). Additionally, we consider an offset $\delta \in \{-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10\}$ which is the distance to the middle point of the anchor segment s. We illustrate some examples of different offsets and fixed-length shingle lengths in Figure 3.

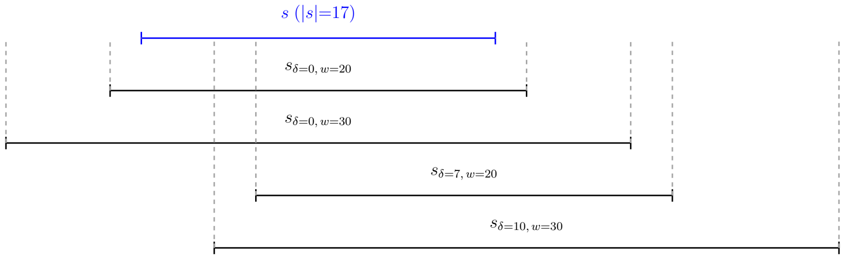
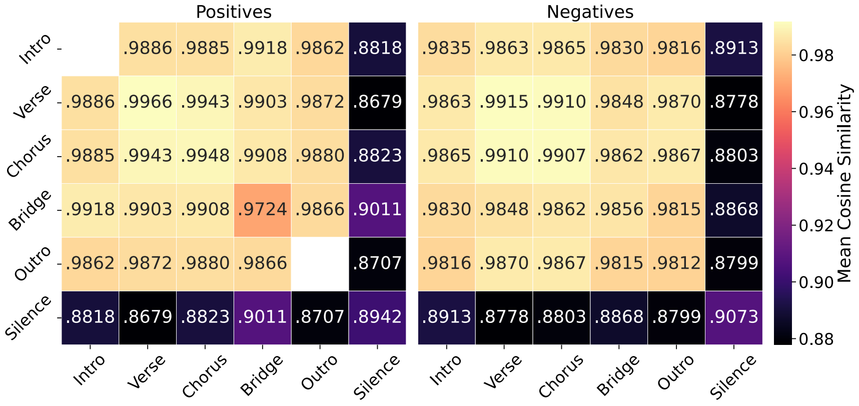


Fig. 1: Example of an anchor segment s of length 17 and segments of another version of the same work with different offsets δ from the middle of the anchor segment and shingle lengths w.

[4] We omit a definition for $L^{=}$, since it would represent the identity relationship (e.g., comparing the chorus s of version v to itself) which is not useful for our analyses.



Positives	Intro	Verse	Chorus	Bridge	Outro	Silence
Intro		.9886	.9885	.9918	.9862	.8818
Verse	.9886	.9966	.9943	.9903	.9872	.8679
Chorus	.9885	.9943	.9948	.9908	.9880	.8823
Bridge	.9918	.9903	.9908	.9724	.9866	.9011
Outro	.9862	.9872	.9880	.9866		.8707
Silence	.8818	.8679	.8823	.9011	.8707	.8942

Negatives	Intro	Verse	Chorus	Bridge	Outro	Silence
Intro	.9835	.9863	.9865	.9830	.9816	.8913
Verse	.9863	.9915	.9910	.9848	.9870	.8778
Chorus	.9865	.9910	.9907	.9862	.9867	.8803
Bridge	.9830	.9848	.9862	.9856	.9815	.8868
Outro	.9816	.9870	.9867	.9815	.9812	.8799
Silence	.8913	.8778	.8803	.8868	.8799	.9073

Fig. 2: Segment-wise mean cosine similarities for the most frequent segment label combinations for positive (V^+) and negative (V^-) version-segment pairs. White boxes indicate the absence of the pairs for *Intro* and *Outro*, since these occur only with one segment each in the respective version.

3.3 VI System

We use the VI system CLEWS [9] with its provided checkpoint by the authors. We set all parameters to the defaults except for the shingling parameters for which we align start and end times of the shingle embeddings with the segment timestamps in the ground truth which results in variable length shingles corresponding to anchor segments. After the pooling and projection layer, we obtain embedding vectors of length 1,024. The authors propose the dimension-normalized Euclidean distance to operate on the version embeddings during training and inference. For better interpretability, we use the cosine similarity instead, due to its fixed interval of possible values between -1 and 1. As we will see in the next section however, the range of cosine values obtained from the pre-trained CLEWS model on our dataset is quite narrow, with minimum values above 0.8. While VI evaluation usually relies on retrieval metrics such as mean average precision (MAP), we avoid these metrics in our case study due to their dependence on the dataset size which is rather small in our case.

4 Results

4.1 Version-Segment Relationships

Figure 2 shows an overview of version-segment relationships for positives V^+ and negatives V^- aggregated by label. Generally, the similarities are all rather high. While the segment *Silence* consistently is more dissimilar to the other segments, we also see that *Chorus* and *Verse* appear to be generally the most similar to

Table 3: Statistics of pairwise cosine similarities by version-segment relationships.

Set	Mean	Std.	Median	# Pairs
$V^{=} \cap L^{+}$	.9958	.0034	.9968	174
$V^{=} \cap L^{-}$	.9910	.0066	.9929	456
$V^{+} \cap L^{+}$	.9936	.0040	.9950	232
$V^{+} \cap L^{-}$	.9891	.0080	.9914	440
$V^{-} \cap L^{+}$	.9908	.0036	.9913	1,238
$V^{-} \cap L^{-}$	.9880	.0068	.9901	2,716

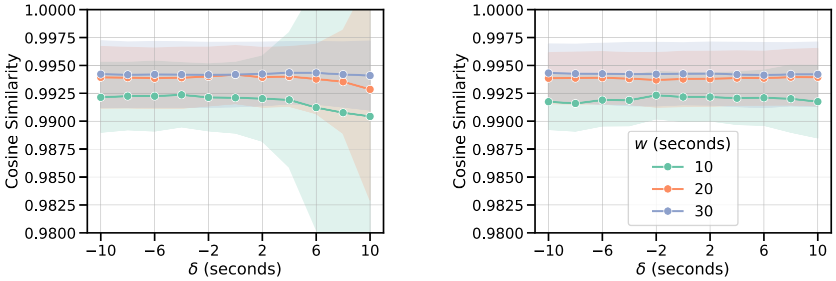
each other and other segments in the case of positives and negatives. A potential issue is that these two segments are even more similar in negative cases compared to some other positive relationships, such as all the ones to *Outro* or most of the ones to *Intro*. We also see that the *Bridge* appears to be rather dissimilar across the versions in our dataset, and is even lower than the comparison of these segments to different cliques or other segments.

To not only focus on the most frequently occurring segment label combinations and gather more general results, Table 3 provides statistics of our defined version-segment relationships. For all pair combinations of version-segment relationship groups, we observe significant differences measured with the Mann-Whitney-U test ($p < 0.05$) except for the comparison of $V^{-} \cap L^{+}$ to $V^{+} \cap L^{-}$. This confirms the previous observation in Figure 2 that in fact segments with a different label but of actual positive version pairs are competing with segments with the same label but of actual negative version pairs. With respect to other expected observations we can confirm that $V^{=} \cap L^{+}$ and $V^{+} \cap L^{+}$ have the highest similarities. We also see that the negative pair $V^{-} \cap L^{+}$ has slightly lower distances than the positive pair $V^{+} \cap L^{-}$. Furthermore, $V^{+} \cap L^{+}$ has a higher mean than $V^{=} \cap L^{-}$. This is favorable, since the cross-version similarity of the same segment is more important in VI than the intra-version similarity of different segments.

4.2 Segment-to-Shingle Offsets

We investigate the impact of offsets to the middle of the anchor segments for different fixed shingle lengths in Figure 3. We distinguish between *short* and *long* anchor segment lengths by splitting at 15 seconds, which roughly corresponds to the mean and median segment length in our dataset.

We generally see high similarities for fixed-length shingles of lengths 20 and 30 seconds, but low ones for shingles of 10 seconds. This is similar to the observation by [9] and might be due to a natural lower limit for the segment length in VI. Furthermore, a drop in similarity can be seen starting at $\delta = 6$ for short segments and $w \in \{10, 20\}$. Intuitively this is due to the low coverage of the anchor segment where only a fraction of it is covered at the end. In contrast, we do not see this effect for the analog negative offsets, which indicates that the information at the



(a) Short: length $<$ 15 seconds.

(b) Long: length $>=$ 15 seconds.

Fig. 3: Mean Cosine similarities and 95% confidence intervals of version-segment pairs in $V^+ \cap L^+$ for different offsets of the middle δ and different fixed shingle lengths w.

beginning of the embedding is prioritized. Some deeper analysis is necessary to examine this observation further.

While its range of similarity values appears to be rather small, considering the generally high similarities observed before, these deviations can still impact the overall retrieval accuracy. However, overall the similarity still seems to be rather stable for $-10 \leq \delta \leq 4$ and for the full range of tested values in the case of long segments except for $w = 10$.

5 Conclusion

In this paper, we conducted a case study to encourage a more segment-aware perspective on the task of VI. While previous studies focus on shingles of fixed-length, we provide a novel VI dataset with segment annotations. While the dataset is rather small in terms of versions, we believe that our analysis provides insights about potential problems in VI regarding bias in correspondence with segment similarities of negative pairs, possibly due to general characteristics of segments because of the nature of segments. In future work, we plan to annotate a larger dataset. Since the annotation effort of segments is rather large, we want to evaluate whether we can find versions for existing datasets in the field of music structure analysis. Additionally, existing music structure analysis algorithms can be exploited to automatically annotate larger datasets.

References

1. Araz, R.O., Serra, X., Bogdanov, D.: Discogs-VI: A musical version identification dataset based on public editorial metadata. In: International Society for Music Information Retrieval Conference (ISMIR). pp. 478–485 (2024)

2. Du, X., Chen, K., Wang, Z., Zhu, B., Ma, Z.: Bytecover2: Towards dimensionality reduction of latent embedding for efficient cover song identification. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 616–620. IEEE (2022)
3. Du, X., Wang, Z., Liang, X., Liang, H., Zhu, B., Ma, Z.: Bytecover3: Accurate cover song identification on short queries. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2023)
4. Goto, M., Hashiguchi, H., Nishimura, T., Oka, R.: Rwc music database: Popular, classical and jazz music databases. In: International Society for Music Information Retrieval Conference (ISMIR). pp. 287–288 (2002)
5. Hachmeier, S., Jäschke, R.: On the robustness of cover version identification models: a study using cover versions from YouTube. Information Research an international electronic journal **30**(iConf), 1103–1122 (Mar 2025). https://doi.org/10.47989/ir30iConf47077
6. Hu, S., Zhang, B., Lu, J., Jiang, Y., Wang, W., Kong, L., Zhao, W., Jiang, T.: Wideresnet with joint representation learning and data augmentation for cover song identification. In: Interspeech. pp. 4187–4191 (2022)
7. Liu, F., Tuo, D., Xu, Y., Han, X.: Coverhunter: Cover song identification with refined attention and alignments. In: International Conference on Multimedia and Expo (ICME). pp. 1080–1085. IEEE (2023)
8. Nieto, O., McCallum, M.C., Davies, M.E., Robertson, A., Stark, A.M., Egozy, E.: The harmonix set: Beats, downbeats, and functional segment annotations of western popular music. In: International Society for Music Information Retrieval Conference (ISMIR). pp. 565–572 (2019)
9. Serrà, J., Araz, R.O., Bogdanov, D., Mitsufuji, Y.: Supervised contrastive learning from weakly-labeled audio segments for musical version matching. In: Forty-second International Conference on Machine Learning (ICML) (2025)
10. Smith, J.B.L., Burgoyne, J.A., Fujinaga, I., De Roure, D., Downie, J.S.: Design and creation of a large-scale database of structural annotations. In: International Society for Music Information Retrieval Conference (ISMIR). pp. 555–560 (2011)
11. Van Balen, J., Burgoyne, J.A., Wiering, F., Veltkamp, R.C.: An analysis of chorus features in popular song. In: International Society for Music Information Retrieval Conference (ISMIR). pp. 107–112 (2013)
12. Wang, J.C., Hung, Y.N., Smith, J.B.: To catch a chorus, verse, intro, or anything else: Analyzing a song with structural functions. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 416–420. IEEE (2022)
13. Wang, J.C., Smith, J.B., Chen, J., Song, X., Wang, Y.: Supervised chorus detection for popular music using convolutional neural network and multi-task learning. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 566–570. IEEE (2021)
14. Xu, X., Chen, X., Yang, D.: Key-invariant convolutional neural network toward efficient cover song identification. In: International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2018)
15. Yesiler, F., Tralie, C., Correya, A., Silva, D.F., Tovstogan, P., Gómez, E., Serra, X.: Da-TACOS: A dataset for cover song identification and understanding. In: International Society for Music Information Retrieval Conference (ISMIR). pp. 327–334 (2019)
16. Yu, Z., Xu, X., Chen, X., Yang, D.: Learning a representation for cover song identification using convolutional neural network. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 541–545. IEEE (2020)