On the Evaluation of Automatic Segment Boundary Detection

Marcelo Rodríguez-López and Anja Volk

Department of Information and Computing Sciences,
Utrecht University, Utrecht, NL.
{m.e.rodriguezlopez,a.volk}@uu.nl

Abstract. Research in automatic music segmentation has been conducted by subdividing the segmentation problem into a number of different tasks, the most fundamental one being segment boundary detection (i.e. automatically locating the time instants separating contiguous segments). Traditionally, the evaluation of segment boundary detection is treated as a binary classification problem. That is, automatic and human annotated boundaries are classified into hits, if they coincide or are temporally close, or misses, if boundaries have no near neighbours. Most often F-scores are used to evaluate the classification results. This evaluation method has two problems: first, no partial score is given to near misses, and second, there is no way to assess the 'veracity' of full misses. In this paper we discuss these two problems, and propose strategies to tackle them.

Keywords: Automatic Music Segmentation, Melody Segmentation, Evaluation, Audio Music Processing, Symbolic Music Processing.

1 Introduction

In cognitive science, segmentation refers to a mechanism of human cognition by which cognitively relevant 'units of information' are abstracted from sensory information (e.g. words or phrases from spoken utterances and objects or parts of objects from a visual scene). Segmentation is considered core to cognitive activities such as learning, reasoning, and comprehension [1]. In the context of music listening, segmentation allows listeners to break down a stream of acoustic information into units such as notes, figures, phrases, sections, and so on. Automatic music segmentation is hence highly important for fields concerned with simulating human-like music processing, such as Generative Arts (to determine musical units in systems that include music generation stages, or to synchronise music with other media), Music Information Retrieval (for music archiving, retrieval, and visualisation), and Computational Musicology (for automatic or computer-assisted music analysis).

Research in music segmentation modelling has been conducted by subdividing the segmentation problem into a number of different tasks, the oldest and most fundamental being segment boundary detection. Generally, two scenarios have been defined to evaluate segmentation tasks: a *direct* scenario, in which automatic segmentations are evaluated by comparing them to manual (human annotated) segmentations; an *indirect* scenario, in which automatic segmentations are evaluated by assessing the role of the produced segments within other (larger) music processing tasks, such as retrieval or visualisation. Segment boundary detection has been most often evaluated in a direct scenario, where the problem of comparing automatic and manual segmentations is treated as a binary classification problem. That is, first predicted boundaries are classified as either hits or misses, and then the traditional Precision, Recall, and F-score measures from information retrieval are used to evaluate the classification results. In the context of music, hits (true positives) correspond to predicted and annotated boundaries that fully coincide or are in close temporal proximity. Conversely, misses (false positives or false negatives) correspond to predicted or annotated boundaries without close neighbours.

In this paper we centre our discussion on two issues related to the evaluation of boundary detection in a direct scenario. First, *currently no partial score is given to near misses.* If a near miss is considered to be a hit it receives full score, and if not it receives a null score. This results in performance estimates which are either too harsh or too permissive. We survey evaluation measures proposed in the field of text segmentation that attempt to overcome this problem. In a case study we showed how these measures can deal with near misses better than measures currently used in music segmentation. Second, *the concept of full miss is ill-defined.* That is, due to the fact that the number human segmenters participating in the development of annotated databases is relatively small, perceptually valid boundaries might not be present in the annotations. Consequently, there is no way to assess the perceptual 'veracity' of a full miss. We discuss a number of strategies (including new tasks and evaluation scenarios) that can help to ameliorate this problem.

The remainder of this paper is structured as follows. In §2 we formally describe the task and evaluation of segment boundary detection. In §3 we discuss alternatives to deal with the problem of near misses. In §4 we discuss strategies (tasks and scenarios) to deal with the problem of full misses. Finally, in §5 we summarise our conclusions.

2 Preliminaries: Task definition and Evaluation Scenario

Formal task definition: Segment boundary detection is the task of automatically locating the time instants separating contiguous segments. The input to an automatic segmenter is a piece or fragment of music, represented as a sequence $x = \langle x_1 \dots x_n \rangle$ of 'atomic' events $x_{i=1,\dots,n}$ deemed appropriate for the description of the music.[1] The output is most often a vector of potential-boundary-locations $\mathbf{b} = (b_1, \dots, b_n)$ where $b_{i=1,\dots,n} \in \mathbb{R}$ are used as an indication of boundary presence strength.

[1] More precisely, if the input data is a music recording, $x_i \in \mathbb{R}^d, \forall i = 1, \dots, n$ are audio windows often lasting a few tens or hundreds of milliseconds represented using d-dimensional feature vectors (commonly MFCCs or Chroma). If the input data is a symbolic music encoding, $x_i \in \xi$, where ξ is a finite and discrete attribute space approximating the attribute space of music theoretic notes, i.e. the space defined by ξ is at least $\texttt{onset} \otimes \texttt{offset} \otimes \texttt{pitch}$, with $\otimes$ denoting the Cartesian product.

Evaluating automatic segmentation: Segment boundary detection is generally evaluated in a direct scenario, where automatically identified boundaries are compared to manually identified boundaries (i.e. annotated by human listeners).[2] Automatically and manually identified boundaries need to be made comparable, and so both are encoded, respectively, as binary vectors $\mathbf{a} = (a_1, \ldots, a_n)$ and $\mathbf{m} = (m_1, \ldots, m_n)$, where $a_i, m_i \in \{0, 1\}, \forall i = 1, ..., n$. Vector element positions encode potential-boundary-locations, a 1 encodes boundary presence, and a 0 encodes boundary absence.

Once the binary encoding procedure is carried out, the most common evaluation strategy is to first check for boundary misplacing, and then use misplacement information to compute the similarity between **a** and **m**. A value of 0 should reflect that all boundaries in **a** are misplaced in respect to those of **m**, and a value of 1 should reflect that all boundaries in **a** perfectly coincide with those of **m**.

3 Near Misses: Moving Away from Traditional Measures

The problem of near misses: Boundary perception studies show that, even when listeners roughly agree on the total number of boundaries in a piece, constructing histograms of boundary indications reveals clusters of closely located boundaries (e.g. see [2, 3] for boundaries of phrases in melodies, and [4][ch. 2] for boundaries of sections in polyphonic music). The reasons for these differences are diverse. To name one, different listeners might be using different strategies when marking boundaries (one giving more importance to long musical rests, while another one might be focusing on the repetition of melodic figures or harmonic progressions). The presence of closely-located boundary clusters suggests that, when checking for boundary misplacement, both full and near misses should be considered. Figure 1 provides an example of near misses between two hypothetical segmentations $s_{1,2}$.

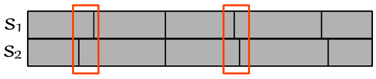


Fig. 1. Example of near misses

Traditional evaluation measures in music segmentation: Boundary misplacement is viewed as a classification problem. That is, taking **a** and **m**, each

[2] The process of annotation requires human annotators to listen to a piece or fragment of music, and 'mark' the time points where they believe segments have finished/begun. The marking process can be actual or notional. Actual marking refers to the case when boundaries are identified by marking a visual (waveform, score, other) depiction of the music. Notional marking refers to the case when no visual aid is provided, and listeners are required simply to 'press a button' to indicate the times when boundaries occur. Manually identified boundaries are stored as a sequence of time values (in some cases absolute and in others relative).

pair of corresponding vector elements is classified as either a true positive tp ($a_i = 1 \wedge m_i = 1$), true negative tn ($a_i = 0 \wedge m_i = 0$), false positive fp ($a_i = 1 \wedge m_i = 0$), or false negative fn ($a_i = 0 \wedge m_i = 1$). Then, the similarity between $\mathbf{a}$ and $\mathbf{m}$ is most often computed using the F_β measure (with $\beta = 1$)

$$F_\beta = \frac{(1+\beta^2) \cdot P \cdot R}{(\beta^2 \cdot P) + R} \in [0,1], \tag{1}$$

where Precision P and Recall R are defined as

$$P = \frac{TP}{TP + FP}, \tag{2}$$

$$R = \frac{TP}{TP + FN}, \tag{3}$$

and TP, FP, and FN correspond to the total number of tp, fp, and fn occurrences, respectively.

Benefits of the F_1, *Precision*, and *Recall* measures: Quantifying binary vector similarity using the F_1, P, and R measures has the benefit of not considering information on true negatives, which due to the strongly unequal proportions of boundary presence/absence values in music segmentation data would result in biased performance estimates.[3] Moreover, the P and R measures allow two interpretations of boundary misplacing: ‘over-segmentation’, i.e. introducing too many spurious boundaries (high R, low P), and ‘under-segmentation’, i.e. missing too many annotated boundaries (high P, low R).

3.1 Evaluation of Near Misses Using Traditional Measures

The most common strategy to handle the near miss problem is to allow for a small tolerance δ when determining boundary matches.

In an ideal situation, a significant number of human listeners would have annotated the pieces, and this would allow to compute distributions of possible boundary locations. These distributions could then be used to estimate how large δ should be, and what score should be awarded to near misses. Some measures that formalise these ideas have been proposed (see [2, 3]). However, at present large benchmark datasets for segmentation have been annotated by at most three human listeners, which impedes a reliable estimation of boundary location distributions. Since segment boundary annotation is a time consuming

[3] Segment boundaries are sparse. For example, in [5] is indicated that in a melodic dataset adding up to ∼79000 notes, only about 12% of the note locations correspond to phrase level segment boundaries. Thus, standard evaluation measures in information retrieval using TN information, such as $accuracy = \frac{TP+TN}{TP+TN+FP+FN}$, would result in a biased assessment value. For instance, if a manual segmentation for a piece marks 20% of possible-boundary-locations with boundary presence, a naïve automatic segmentation predicting only boundary absences (an all-zero vector) would still receive an accuracy score of 80%.

and costly process, it is unrealistic to expect densely annotated datasets to be created in the short term.

At present, δ is most often set according to intuition. In the MIREX Structural Segmentation track (audio input) two tolerance settings have been used: narrow $\delta = \pm 0.5$ seconds and broad $\delta = \pm 3$ seconds. In comparative studies of melody segmentation (symbolic input) three tolerance settings have been used: no tolerance $\delta = 0$, narrow $\delta = \pm 1$ note events, and broad $\delta = \pm 2$ note events. Moreover, no partial score is awarded to near misses, i.e. if the automatically determined boundary falls within the interval set for δ then it is classified as a true positive, otherwise it is classified as a false positive. As a result of not awarding partial scores to near misses, narrow tolerance intervals result in overly pessimistic performance estimates, while broad tolerance intervals result in overly optimistic estimates. These inaccurate estimates complicate the interpretation of the 'true' performance of an automatic segmenter, directly affecting the ranking of the segmenters participating in the evaluation. Additionally, inaccurate estimates might also affect subsequent analyses of performance, such as correlation analyses or outlier analyses.

Alternative performance measures and near misses: In MIREX the $mt2g$ measure has been used as an alternative to evaluate boundary detection performance. The $mt2g$ computes the median distance from each annotated boundary to the nearest predicted boundary. The $mt2g$ can be interpreted in terms of Recall (a high score corresponds to low Recall), and can also be seen to provide a rough account of near misses (a low score indicates a dominance of close near misses). However, assessing the influence of near misses on boundary detection performance can only be achieved indirectly, i.e. by cross-analysing F_1 and $mt2g$ scores, which makes the analysis complex and ultimately unreliable.

Other measures have been tested to complement/replace the F_1, Precision, and Recall measures, such as the *kappa statistic* and the *sensitivity index* d' (tested in [5]), and also the $1 - f$, $1 - m$, $mg2t$, and $mt2g$ measures (tested in [6]). However, aside from the previously discussed $mt2g$, none of the measures takes into account near misses.

3.2 Evaluation of Near Misses in Text Segmentation

In the field of text segmentation, the possibility of near misses also constitutes a major problem when evaluating automatic segmenters [7–11]. In this section we review two measures developed to handle the problem of near misses: *WindowDiff* (WD) [9] and the *Boundary Edit Distance based boundary Similarity* (*BED-S*) [12]. Our reasons to focus only on these two measures are: (1) WD constitutes the current standard measure to quantitatively evaluate automatic text segmenters, and (2) *BED-S* is a recently proposed measure that overcomes a number of limitations of WD.

For the description of these measures we will assume that the automatically and manually segmented text is encoded in the same way as done with music segmentation, i.e. as binary vectors $\mathbf{a}$ and $\mathbf{m}$ of size n.

Description of WindowDiff: WD [9] uses a sliding window of size k to simultaneously scan both $\mathbf{a}$ and $\mathbf{m}$. If within the window the number of boundaries in the manual segmentation differs from the number of boundaries in the automatic segmentation, a penalty is given. The penalty score is assigned according to Equation 4, where $b(m_{i...j})$ and $b(a_{i...j})$ represent the number of reference (manual) and predicted (automatic) boundaries within a window of size k (from position i to j), and n is the number of potential-boundary-locations. The value of WD represents the degree of error between the segmentations, and so often the segmentation score is given by taking $1 - WD$.

$$WD(\mathbf{m}, \mathbf{a}) = \frac{1}{n-k} \sum_{i=1,j=i+k}^{n-k} (|b(m_{i...j}) - b(a_{i...j})| > 0), \tag{4}$$

WD is an improved version of a previously proposed measure (P_k [8]). The proposed improvements of WD (in respect to P_k) were tested and validated in [9]. WD is at present the standard measure for evaluation of text segmentation. However, a number of issues with WD have been identified, the most severe being (i) it under-penalizes errors at the beginning and end of a segmentation [10]; (ii) it favours segmentations with few boundaries [11]; (iii) varying the parameter k leads to difficulties in interpreting and comparing WD values [7].

Description of the Boundary Edit Distance based boundary Similarity: $BED\text{-}S$ [12] takes a different approach to comparing segmentations. Instead of using a sliding window, it models the problem of identifying misplaced boundaries as an alignment problem. To this end [7] proposes a new edit distance called *boundary edit distance* (BED) which differentiates between full and near misses between $\mathbf{a}$ and $\mathbf{m}$. BED uses two main edit operations to model boundary misplacements:

- additions/deletions (A) for full misses
- n-wise transpositions (T) for near misses

BED is based on the Darmeau-Levenshtein edit distance, which formalises A and T operations. An A type operation is a single-unit edit, which as seen in Figure 2 can correspond to either a false positive or a false negative. A T type operation is an adjacent-unit edit, i.e. the act of swapping one unit in a sequence with adjacent units (e.g. the sequence of characters 'ab' becomes 'ba'). Figure 2 depicts a transposition spanning one unit. Since in text segmentation (and also music segmentation) near misses can span more than one possible-boundary-location unit, BED extends the Darmeau-Levenshtein edit distance, which is limited to single-unit transpositions, to accommodate for multiple-unit transpositions. Lastly, if $a_i = m_i$ for $i \in \{1, \ldots, n\}$ (M in Figure 2), BED stores it as a full match (true positive).

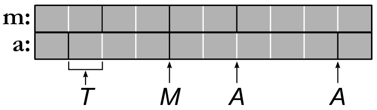


Fig. 2. Boundary edit operations, adapted from [12].

The counts of edit operations are then used to model boundary misplacement penalties as specified in Table 1.

Operation	*Codomain*	*Range*	*Penalty-per-Edit*	*Description*
A_e	$\mathbb{N}_0^2$			set of AD edits
T_e	$\mathbb{N}_0^2$			set of T edits
B_M	$\mathbb{N}_0^2$			set of matching boundaries
$\|A_e\|$	$\mathbb{N}_0$	$[0, n-1]$	1	number of AD edits
$\|T_e\|$	$\mathbb{N}_0$	$[0, \frac{1}{2}\lfloor n-1 \rfloor]$	1	number of T edits
$\|B_M\|$	$\mathbb{N}_0$	$[0, \frac{1}{2}\lfloor n-1 \rfloor]$	0	number of B_M
$W_T(T_e, n_t)$	$\mathbb{Q}_+$	$[0, \frac{1}{2}\lfloor n-1 \rfloor]$	$[0, 1]$	Weighted T_e operations

Table 1. Details for the edits determined using BED, adapted from [7].

Using the counts $|A_e|$, $|T_e|$, and $|B_M|$, *BED-S* can be defined as:

$$\textit{BED-S}(m, a) = 1 - \frac{|A_e| + W_T(T_e, n_t)}{|A_e| + |T_e| + |B_M|}, \tag{5}$$

where

$$W_T(T_e, n_t) = \sum_{j=1}^{|T_e|} \left(bc_t + \frac{abs(T_e[j][1] - T_e[j][2])}{max(n_t) - 1} \right), \tag{6}$$

and n_t, bc_t are user defined parameters that control the maximum transposition distance (in potential-boundary-location units) and a bias constant, respectively.

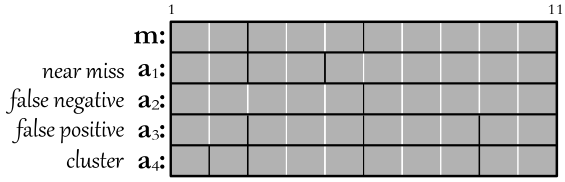
The intuition for using $W_T(T_e, n_t)$ is simple. It is assumed that penalties for near misses should be proportional to the distance between the reference and predicted boundaries. $W_T(T_e, n_t)$ then corresponds to a distance function whose purpose is to scale transposition errors.

The output value of *BED-S* serves as a summary measure of the similarity between **a** and **m**, just like the F_1 score. However, during evaluation one might also want to have higher interpretative power, e.g. in terms of over-segmentation and under-segmentation. To achieve greater interpretability, in [12] a confusion matrix is defined so that TP, TN, FP, and FP are computed using counts of $|A_e|$, $|T_e|$, and $|B_M|$. The confusion matrix can then be used to compute BED-based Precision, Recall, and F_1-measures, which would have the advantage that near misses are accounted for (i.e. TP$= |B_M| + W_T(T_e, n_t)$).

Limitations of *BED-S*: While *BED-S* is conceptually simple, some of the heuristics introduced to deal with multiple-unit transpositions make its implementation more complex than that of the WD or F_1 measures. However, to overcome this limitation, the authors of *BED-S* provide a python implementation[4], and complement it with a detailed documentation (see [7]). The heuristics used within *BED-S* also make the common dynamic programming solution to edit distance computation non-viable, making *BED-S*' time complexity to scale poorly as a function of the number of potential-boundary-locations. For music segmentation this is not problematic as long as the input consists of relatively short sequences, such as melodies or harmonic progressions, but it might become an obstacle if we wish to segment whole musical pieces where each potential-boundary-location is in the order of note events or beats.

3.3 Case Study Evaluation: *BED-S* vs. WD vs. F_1

In Figure 3 (left) we depict a hypothetical group of segmentations, adapted from [12]. The group consists of one reference manual segmentation **m** and four automatic segmentations $\mathbf{a}_1$-$\mathbf{a}_4$ produced by four different automatic segmenters. Each of the first three segmentations contains a single type of error: $\mathbf{a}_1$ has a *false negative*, $\mathbf{a}_2$ has a *near miss*, $\mathbf{a}_3$ has a *false positive*. Segmentation $\mathbf{a}_4$ contains two errors, a *near miss* next to a true positive (called a 'cluster' in Figure 3), and a *false positive*. We consider that a reasonable ranking of the automatic segmentations is (from 'best' to 'worst'): $\mathbf{a}_1$, $\mathbf{a}_2 \approx \mathbf{a}_3$, $\mathbf{a}_4$. We rank $\mathbf{a}_1$ at top position based on the idea that a near miss should be preferable to an insertion/deletion. We rank $\mathbf{a}_4$ at bottom position because it is the only segmentation to contain more than one error type. For the case of $\mathbf{a}_1$ and $\mathbf{a}_3$, we argue that the information provided in this case study is not sufficient to justify giving preference to an insertion over a deletion (or vice-versa),[5] and so we rank both in second position.



k=2	δ=0	δ=±1	nt=2
1-WD	F1	F1	BED-S
$0.7\bar{7}$	$0.6\bar{6}$	1.00	0.75
$0.7\bar{7}$	$0.6\bar{6}$	$0.6\bar{6}$	0.5
$0.7\bar{7}$	0.8	0.8	$0.6\bar{6}$
$0.6\bar{6}$	$0.6\bar{6}$	0.80	0.5

Fig. 3. Hypothetical case study evaluation

In Figure 3 (right) we present a table with the scores of 1-WD (window size $k = 2$), F_1 (with tolerance $\delta = 0$ and $\delta = \pm 1$), and *BED-S* (transposition

[4] Freely available at `http://segeval.readthedocs.org/en/latest/`.

[5] Deciding how to rank $\mathbf{a}_1$ in respect to $\mathbf{a}_3$ is not as intuitive as it might seem. How to penalise misses or insertions ultimately depends on how confident we are that the manual segmentation is correct (number of human annotators and their level of agreement) and the specific application scenario in which the automatic segmentation will be used (we discuss this further in §4).

parameter $n_t = 2$), where n_t, k, δ are measured in potential-boundary-location units. 1-WD ranks $\mathbf{a}_1$–$\mathbf{a}_3$ equally, and thus is only able to discriminate between relatively sparse ($\mathbf{a}_1$–$\mathbf{a}_3$) and clustered ($\mathbf{a}_4$) segmentations. F_1 ($\delta = 0$) ranks $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_4$ equally, and, what its perhaps even less desirable, gives top rank to $\mathbf{a}_3$ (false positive). F_1 ($\delta = \pm 1$) gives top rank to $\mathbf{a}_1$, which matches the rank given to $\mathbf{a}_1$ in our preferred ranking. However, F_1 ($\delta = \pm 1$) also ranks $\mathbf{a}_3$ and $\mathbf{a}_4$ higher than $\mathbf{a}_2$, showing a preference for correctly identifying the reference (manual) boundaries, even if spurious boundaries are added, to missing reference boundaries. Moreover, the score values are overly optimistic, a near miss should not obtain the best possible score. Finally, BED-S scores result in the following ranking: $\mathbf{a}_1$, $\mathbf{a}_3$, $\mathbf{a}_2 \approx \mathbf{a}_4$, being the one that most closely resembles our preferred ranking. Just like the F_1 ($\delta = \pm 1$), BED-S seems to prefer adding a spurious boundary ($\mathbf{a}_3$) to missing a reference boundary ($\mathbf{a}_2$). That said, both BED-S and the F measure provide parameters that allow the user to control this behaviour. In the case of BED-S, the number of additions/deletion edit operation can be weighted according to user preference. Likewise, the general version of the F measure, defined in Equation 1, provides the parameter β, which can be tuned to give preference to either Precision or Recall.

From this case study we can conclude that BED-S can be a suitable replacement for the F-measure if near misses are to be considered during evaluation. Also, $1 - WD$ can be used to investigate if automatic segmenters are producing clusters of true positives, which in music segmentation is most often undesirable.[6]

4 Full Misses: an Ill-Defined Concept

The problem of full misses: As mentioned at the beginning of §3, boundary annotation studies have shown that, even when humans listeners agree on the bulk of boundaries for a segmentation, the specific location of these boundaries might still, to some extent, differ. This observation allowed us to argue that near misses should receive a partial score when evaluating automatic segmenters. However, the complications when evaluating boundary detection do not end there. Boundary annotation studies have shown that humans often also disagree on the total number of marked boundaries [4, 13]. If to that we add the fact that large test databases are at present annotated only by a handful of human listeners, then a major issue becomes apparent: the concept of full misses in direct evaluation scenarios is ill-defined. That is, perceptually valid boundaries might not be present in the annotations, and, consequently, there is no way to assess the 'veracity' of a full miss.

[6] It can be argued that, for text segmentation, deciding whether boundary clusters are unwanted or not depends both on the granularity of segmentation and the units used as potential-boundary-locations. This is because segment granularity can range from syllables to paragraphs, and thus potential-boundary-locations can range from phonemes to whole sentences. Conversely, in music segmentation test databases have annotations only at the level of phrases and sections, and potential-boundary-locations are either at the level of short-time windows (on the order of milliseconds), note events (on the order of seconds), or beats (also on the order of seconds). Hence, boundary clusters often result in cognitively unlikely segments and are thus most often unwanted.

4.1 Tackling Full Misses by Considering Hierarchical Organisation

In recent discussions at ISMIR's late breaking sessions [14, 15], a way to tackle the uncertainty associated to full misses has been suggested, namely to take into account segment hierarchy. The motivation being that a false positive might just be the result of an automatic segmenter producing boundaries at different hierarchical levels, while the annotated boundaries are only at one hierarchical level. This situation is depicted in Figure 4. The manual segmentation **m** marks boundaries at level two of the hierarchy. The automatic segmentation **a** predicts boundaries at levels one and two, and so in the absence of hierarchy information the level one boundaries appear to be full misses (false positives).

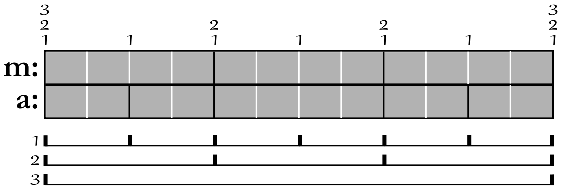


Fig. 4. Full boundary misses explained by hierarchy mismatch.

If we take an automatic segmenter that produces a hierarchy of boundaries (e.g. [16, 17]), each set of boundaries comprising one hierarchical level could be compared against each hierarchical set of annotated boundaries (assuming that the reference boundaries have also been annotated with hierarchy). The similarity of the highest matching pair of automatic and manual segmentations can be taken as an estimate of performance.

Issues with using segment hierarchy to disambiguate full misses: At present, large annotated datasets consist of boundaries annotated with at most two levels of hierarchy (see [18, 19]), which does not get us very far. Hence, if we are to use segment hierarchy as a means to deal with full misses, hierarchy annotations would need to be expanded.

Now, the situation depicted in Figure 4 is what we could call a 'strict' hierarchy in that every segment is completely and exactly contained within every larger segment, thus rejecting the possibility of *overlapping segments*. Also, Figure 4 represents *only one possible hierarchical organisation* of segments. Theories of music cognition, on the other hand, do accept the possibility of both overlapping segments and multiple hierarchical organisations of segments (see for example [20, p. 13] for a classic view on the topic of segment hierarchy, and [21, pp. 134-139] for a more recent one).

Even for pieces of moderate length and a modest number of hierarchy layers, annotating a segment hierarchy resembling that of Figure 4 would be a time consuming process, as it requires the annotator to consider the relational structure of segments within and across the different layers of the hierarchy. If on top of that we consider the possibility of overlapping segments and multiple hierarchical organisations, then not only the annotation time is expected to increase, but also a number of other issues. For instance, the annotators would need to design and use complex representational formalisms (i.e. hierarchies being formally represented as multiple trees or semi latices), and as consequence we could also expect low inter-annotator-agreement. It would then seem that the hierarchy annotation problem escalates to the point that the complications of annotating hierarchy overshadow its benefits as means to give more appropriate scores to full misses.

4.2 Other Alternatives to Tackle the Problem of Full Misses

Below we present three alternatives to deal with full misses, which are perhaps simpler (and hence better suited) than considering segment hierarchy.

Extend boundary annotations to account for boundary salience: By boundary salience we mean how 'clear' a listener might perceive the boundary (starting or ending point) of a segment to be. There are three potential benefits of boundary salience annotation over segment hierarchy annotation. First, the former task seems relatively easy to communicate to human annotators (see for instance the salience annotation study in [4][Ch. 4]). Second, there seems to be relatively high inter-annotation-agreement for boundary salience annotation (see ibid.). Third, the task of annotating boundary salience is in principle less time consuming than hierarchy annotation.

From the perspective of scoring full misses, the motivation to encourage boundary salience annotation is that it can be used to attenuate/strengthen the score given to false negatives. That is, salience-based scoring functions can be designed so that the penalty applied to false negatives is scaled by the annotated salience of the boundary. In consequence, by using salience annotations the initially strong assumptions regarding the perceptual plausibility of 'false negative' boundaries is ameliorated.

Extend boundary annotations to account for segmentation strategy: By segmentation strategy we mean the musical 'cues' a given listener might use to mark boundaries during a segment annotation study. To the best of our knowledge segment strategy annotation has not been previously attempted. However, a few studies in boundary perception [22, 23] have asked participant listeners to document the segmentation strategies they employed after the segmentation experiment was conducted. In these studies segment strategy classes have been created by manually inspecting the participant data. These classes can hence be used as a template or guide to enrich currently available boundary annotated corpora with segmentation strategy labels. Moreover, enriching boundary annotations with segmentation strategy can also be assisted by automatic means. That is, computational methods (e.g. see [24, 25]) can be used to suggest what strategies might have been used for a particular segmentation.

From the perspective of scoring full misses, the motivation to encourage segmentation strategy annotation is that it can be used to give partial scores to false negatives. For instance, if a given boundary was annotated focusing mainly on a cue that is not modelled by the automatic segmenter, a resulting false negative on this boundary would be expected and thus should not be penalised.

Segmentation strategy annotation can also be used to identify pieces that should be excluded from the evaluation. For instance, if the boundary annotations for a given piece of music indicate a majority of boundaries annotated with cue ‘A’ and a given automatic segmenter uses only cues of type ‘B’, then the piece can be excluded from the evaluation on the grounds that many ‘unverifiable’ false positives would be expected.

Complement direct with indirect evaluation scenarios: In an indirect scenario automatic segmentations are evaluated by assessing the role of the produced segments within other (larger) music processing tasks, such as retrieval or visualisation. The motivation for employing indirect scenarios is that having a clear goal for the segmentations can result in having a more clear preference for segment lengths, as well as for the quantity and quality of boundaries.

For instance, lets say we have a set of hour long improvised music recordings annotated with segment boundaries indicating the different parts of the improvisations. Lets then assume a hypothetical situation where a digital audio workstation (DAW) user wants to quickly edit the recordings. Lets also assume that the user expects that, when one of the recordings is loaded into the DAW, markers indicating the boundaries between different parts of the improvisation are automatically displayed over the waveform. In this hypothetical scenario, the user might prefer a segmentation which avoids visual clutter, so as to quickly start editing. Taking this hypothetical scenario can then introduce an additional constraint to evaluate what a ‘good’ segmentation should be: sparseness. That is, we can evaluate the segmentations by comparing them to the annotations and additionally by requiring them to be sparse. With the former (direct) evaluation we can focus only on determining whether automatic boundaries match annotated ones (hits and near misses). With the latter (indirect) evaluation we can evaluate those boundaries that did not match the human annotated ones (full misses) in respect to the user-determined sparsity constraints. For instance, a simple user-determined sparsity constraint could be that segments need to be separated by at least 5 seconds. So, during evaluation all automatic boundaries in a 5 second vicinity of a hit boundary are penalised. An appropriate measure to set a score based on vicinity penalty is WD described in §3.2.

5 Conclusions

In this paper we have discussed two important issues related to the evaluation of boundary detection. First, current evaluation measures do not award partial scores to near misses, which results in performance estimates that are either overly optimistic or overly pessimistic. We hence surveyed measures proposed in the field of text segmentation, designed to deal with near misses. In a case study we showed how these measures can deal with near misses better than measures currently used in music segmentation. Second, we show the concept of full miss is ill-defined, due to annotated data sparsity. Consequently there is no way to assess the ‘veracity’ of a full miss from the point of view of perception and cognition. We hence discussed three strategies that can help to ameliorate this problem: (1) extend boundary annotations to account for boundary salience;

(2) extend boundary annotations to account for segmentation strategy; (3) complement direct with indirect evaluation scenarios. These strategies move away from assessing the perceptual 'veracity' of full misses, thus helping to tackle the evaluation problem from a different angle.

Acknowledgments We thank the anonymous reviewers for the useful comments on earlier drafts of this document. M. Rodríguez-López and A. Volk are supported by the Netherlands Organization for Scientific Research (NWO-VIDI grant 276-35-001).

References

1. Gobet, F., Lane, P.C., Croker, S., Cheng, P.C., Jones, G., Oliver, I., Pine, J.M.: Chunking mechanisms in human learning. Trends in cognitive sciences **5**(6) (2001) 236–243
2. Melucci, M., Orio, N.: A comparison of manual and automatic melody segmentation. In: Proceedings of the International Conference on Music Information Retrieval. (2002) 7–14
3. Spevak, C., Thom, B., Höthker, K.: Evaluating melodic segmentation. In: Music and Artificial Intelligence. Springer (2002) 168–182
4. Bruderer, M.J.: Perception and modeling of segment boundaries in popular music. PhD thesis, Doctoral dissertation, JF Schouten School for User-System Interaction Research, Technische Universiteit Eindhoven, Nederlands (2008)
5. Pearce, M., Müllensiefen, D., Wiggins, G.: Melodic grouping in music information retrieval: New methods and applications. Advances in music information retrieval (2010) 364–388
6. Smith, J.B., Chew, E.: A meta-analysis of the mirex structure segmentation task. In: Proc. of the 14th ISMIR, Curitiba, Brazil. (2013) 251–256
7. Fournier, C.: Evaluating text segmentation. Master's thesis, University of Ottawa (2013)
8. Beeferman, D., Berger, A., Lafferty, J.: Statistical models for text segmentation. Machine learning **34**(1-3) (1999) 177–210
9. Pevzner, L., Hearst, M.A.: A critique and improvement of an evaluation metric for text segmentation. Computational Linguistics **28**(1) (2002) 19–36
10. Lamprier, S., Amghar, T., Levrat, B., Saubion, F.: On evaluation methodologies for text segmentation algorithms. In: Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on. Volume 2., IEEE (2007) 19–26
11. Niekrasz, J., Moore, J.D.: Unbiased discourse segmentation evaluation. In: Spoken Language Technology Workshop (SLT), 2010 IEEE, IEEE (2010) 43–48
12. Fournier, C.: Evaluating text segmentation using boundary edit distance. In: Proc. of the 51st Annual Meeting of the Association for Computational Linguistics. (2013) 1702–1712
13. Pearce, M., Müllensiefen, D., Wiggins, G.: The role of expectation and probabilistic learning in auditory boundary perception: a model comparison. Perception **39**(10) (2010) 1365
14. Rocha, B., Smith, J.B., et al.: Late-break session on music structure analysis. In: Late-breaking and demo session, ISMIR. (2012)
15. Nieto, O., Smith, J.B., et al.: 2013 late-break session on music segmentation. In: Late-breaking and demo session, ISMIR. (2013)
16. McFee, B., Ellis, D.: Analyzing song structure with spectral clustering. In: International Society for Music Information Retrieval Conference. (2014)

17. Thornton, C.: Generation of folk song melodies using bayes transforms. Journal of New Music Research **40**(4) (2011) 293–312
18. van Kranenburg, P., de Bruin, M., Grijp, L.P., Wiering, F.: The meertens tune collections. Meertens Online Reports (2014)
19. 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: Proceedings of the International Society for Music Information Retrieval Conference (ISMIR). (2011) 555–60
20. Lerdahl, F., Jackendoff, R.: A generative theory of tonal music. MIT press (1985)
21. Wiggins, G.A., Forth, J.: IDyOT: A computational theory of creativity as everyday reasoning from learned information. In: Computational Creativity Research: Towards Creative Machines. Springer (2015) 127–148
22. Clarke, E.F., Krumhansl, C.L.: Perceiving musical time. Music Perception (1990) 213–251
23. Bruderer, M.J., Mckinney, M.F., Kohlrausch, A.: The perception of structural boundaries in melody lines of western popular music. Musicae Scientiae **13**(2) (2009) 273–313
24. Smith, J.B.L., Chuan, C.H., Chew, E.: Audio properties of perceived boundaries in music. IEEE Transactions on Multimedia (2013)
25. Rodríguez-López, M., Volk, A.: Symbolic segmentation: A corpus-based analysis of melodic phrases. In: Sound, Music, and Motion. (2014) 548–557