Music in Use

Novel perspectives on content-based music Retrieval

Yadati, Karthik

DOI
10.4233/uuid:24437481-873f-4bc6-84a3-57d0a6e4e0ae

Publication date
2019

Document Version
Final published version

Citation (APA)
Yadati, K. (2019). *Music in Use: Novel perspectives on content-based music Retrieval*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:24437481-873f-4bc6-84a3-57d0a6e4e0ae

MUSIC IN USE

NOVEL PERSPECTIVES ON CONTENT-BASED MUSIC RETRIEVAL

MUSIC IN USE

NOVEL PERSPECTIVES ON CONTENT-BASED MUSIC RETRIEVAL

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op woensdag 15 mei 2019 om 12.30 uur

door

Narasimha Karthik YADATI

Master of Science in Computing,
National University of Singapore, Singapore,
geboren te Cuddapah, India.

Dit proefschrift is goedgekeurd door de

promotor: Prof. dr. A. Hanjalic
copromotor: dr. C.C.S Liem

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. A. Hanjalic,	Technische Universiteit Delft
dr. C.C.S Liem,	Technische Universiteit Delft

Onafhankelijke leden:

dr. D. R. Turnbull	Ithaca College
Prof. dr-ing. S. Stober	Otto-von-Guericke-U. Magdenburg
Prof.dr. ir. W. Kraaij	Leiden University
Prof. dr. A. van Deursen	Technische Universiteit Delft
Prof. dr. M. A. Neerincx	Technische Universiteit Delft

Prof. dr. M. A. Larson heeft als begeleider in belangrijke mate aan de totstandkoming van het proefschrift bijgedragen.

Keywords: music as technology, music for activities, music event detection

Printed by: Ridderprint BV | www.ridderprint.nl

Front & Back: Beautiful cover art that captures the entire content of this thesis in a single illustration.

ISBN 978-94-6375-416-3

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

Incredible things can be done simply
if we are committed to making them happen.

Sadhguru

CONTENTS

SUMMARY

Music consumption has skyrocketed in the past few years with advancements in internet and streaming technologies. This has resulted in the rapid development of the inter-disciplinary field of Music Information Retrieval (MIR), which develops automatic methods to efficiently and effectively access the wealth of musical content. In general, research in MIR has focused on tasks like semantic filtering, annotation, classification and search. Observing the evolution of MIR over the years, research in this field has been focusing on"what music is" and in this thesis we move towards building tools that can analyse "what music does" to the listener. There is little research on building systems that analyse how music affects the listener or how people use music to suit their needs. In this thesis, we propose methods that push the boundaries of this perspective.

The first major part of the thesis focuses on detecting high-level events in music tracks. Research on event detection in music has been restricted to detecting low-level events viz., onsets. There is also an abundance of literature on music auto-tagging, where researchers have focused on adding semantic tags to short music snippets. However, we look at the problem of event detection from a different perspective and turn to social music sharing platform – SoundCloud to understand what events are of importance to the actual listeners. Using a case-study in Electronic Dance Music (EDM), we design an approach to detect high-level events in music. The high-level events in our case-study have a certain impact on the listeners causing them to comment about these events on SoundCloud. Through successful experiments, we demonstrate how these high-level events can be detected efficiently using freely available but noisy user comments. The results of this approach inspired us for further research to investigate other tasks that can give us more insight into how music affects the listener.

The second major part of the thesis concerns identifying music that can support different common activities – working, studying, relaxing, working out etc. A certain type of music is suitable for enabling listeners to perform a certain task. We first investigate what activities are important from a listeners' perspective, for which music is sought, through a data-driven experiment on YouTube. After illustrating how existing music metadata like genre, instrument is insufficient, we propose a method that can successfully classify music based on the activity categories. An important insight from our experiments is that dividing the music track into short frames is not an effective method of feature extraction for activity-based music classification. This task requires a longer time window for feature extraction. Additionally, presence of high-level events like drop can affect the classification performance.

After successful validation of our idea on activity-based music classification, we went on to investigate what can potentially distract a listener while doing a task. For this, we gathered valuable input from users of Amazon Mechanical Turk (AMT) on what musical characteristics distract them while doing their tasks. Based on this input, we built a system that can automatically detect a derail moment in a given music track, where

the listener could potentially get distracted (derailed). Though this task seems to have a likely subjective component, we demonstrated that there are universal aspects to it as well. Through a literature survey and computational experiments, we demonstrate that we can automatically detect a derail moment.

Throughout the thesis, we also stress on the importance of crowdsourcing platforms like AMT and social media sharing platforms like SoundCloud, and YouTube in understanding the user's requirements and gathering data. We believe that our proposed methods and their outcomes will encourage future researchers to focus on this breed of MIR tasks, where the focus is on how music affects the listener. We also hope that the insights gained through this thesis will inspire designers and developers to build novel user interfaces to enable effective access of music.

SAMENVATTING

In de afgelopen jaren is men mede door de technische vooruitgang op het gebied van internet- en streamingtechnologiën enorm meer naar muziek gaan luisteren. Dit had tot gevolg dat de ontwikkeling van het interdisciplinaire onderzoeksgebied Music Information Retrieval (MIR), waarin men automatische methoden ontwikkelt om een overvloed aan muziekcontent efficiënt en effectief te kunnen benaderen, in een stroomversnelling raakte. In het algemeen heeft MIR-onderzoek zich voornamelijk gericht op taken als het semantisch filteren, annoteren, classificeren en zoeken van muziek. Als we beschouwen hoe het onderzoeksgebied zich over de jaren heeft geëvolueerd, zien we dat MIR zich vooral bezig heeft gehouden met de vraag "wat muziek is". In dit proefschrift richten we de aandacht op een andere vraag en richten we ons op het ontwikkelen van methoden om te kunnen onderzoeken "wat muziek met de luisteraar doet". Tot nu toe is er weinig onderzoek gedaan naar het bouwen van systemen die analyseren hoe muziek de luisteraar raakt of hoe men muziek gebruikt om in hun behoeften te voorzien. In dit proefschrift stellen we methoden voor die de grenzen van dit perspectief zullen verleggen.

In het eerste grote deel van dit proefschrift richten we ons op het detecteren van veranderingen in muzieknummers, oftewel gebeurtenissen, op een hoger, semantisch niveau, in tegenstelling tot eerder onderzoek dat zich vooral beperkte tot het detecteren van veranderingen op laag niveau, namelijk het detecteren van onsets. Er is ook een overvloed aan literatuur over het automatisch taggen van muziek voorhanden, waarin onderzoekers zich hebben gericht op het toekennen van semantische labels aan korte muziekfragmenten. Wij benaderen het vraagstuk van gebeurtenisdetectie echter van vanuit een ander perspectief en keren ons tot het sociale muziekdeelplatform SoundCloud om te leren begrijpen wat voor gebeurtenissen werkelijk interessant zijn voor luisteraars. Op basis van een casestudy over elektronische dansmuziek (EDM) ontwerpen we een aanpak om hogere gebeurtenissen in muziek te kunnen detecteren. Deze gebeurtenissen in onze casestudy hebben een bepaalde impact op luisteraars die ervoor zorgt dat zij hierover reacties achterlaten op het SoundCloud-platform. Door middel van succesvolle experimenten tonen we aan hoe deze hogere gebeurtenissen efficiënt kunnen worden gedetecteerd door gebruik te maken van de vrijelijk beschikbare, doch met ruis gevulde gebruikersreacties. De resultaten van deze aanpak inspireerden ons om verder onderzoek te plegen naar andere taken die ons meer inzicht kunnen geven in hoe muziek de luisteraar beïnvloedt.

Het tweede grote deel van dit proefschrift behandelt het identificeren van muziek dat algemene activiteiten, waaronder werken, studeren, ontspannen, sporten, enz., kan ondersteunen. Bepaalde typen muziek zijn geschikt om gebruikers in staat te stellen om bepaalde taken uit te voeren. We onderzoeken eerst via een datagedreven experiment op YouTube wat voor activiteiten door muziekluisteraars belangrijk worden gevonden en waarvoor ook muziek wordt gezocht. Nadat we hebben aangetoond dat reeds bestaande

muziekmetadata zoals genre en gebruikte instrumenten tekortschieten, stellen we een methode voor die succesvol muziek kan classificeren op basis van de eerder gevonden activiteitscategorieën. Een belangrijk inzicht verkregen middels onze experimenten is dat het opknippen van een muzieknummer in korte frames niet een effectieve manier is om kenmerken te extraheren voor de classificatietaak in kwestie, maar dat deze op activiteiten gebaseerde muziekclassificatietaak juist een groter tijdvenster vereist. Bovendien kan de aanwezigheid van hogere gebeurtenissen in muzieknummers zoals drops de classificatienauwkeurigheid beïnvloeden.

Na het succesvol valideren van ons idee voor activiteiten gebaseerde muziekclassificatie onderzochten we wat luisteraars mogelijk kan afleiden terwijl zij bezig zijn met een taak. Hiervoor verzamelden we waardevolle input van gebruikers van Amazon Mechanical Turk (AMT) over welke karakteristieke kenmerken in muziek hun van een taak kunnen afleiden. Op basis van deze input bouwden we een systeem dat automatisch een 'ontspoormoment' in een gegeven muzieknummer, d.w.z. een moment waarop een luisteraar mogelijk kan worden afgeleid of ontspoord kan raken, kan detecteren. Hoewel deze taak aannemelijk een subjectieve component lijkt te bevatten, tonen we aan dat er ook universele aspecten aan hangen. Met behulp van een literatuurstudie en computersimulaties tonen we aan dat het mogelijk is om zo'n ontspoormoment automatisch te kunnen detecteren.

Door heel het proefschrift heen benadrukken we ook steeds hoe belangrijk crowdsourcing platforms als AMT en sociale mediadeelplatformen als SoundCloud en YouTube zijn in het begrijpen van gebruikersbehoeften en het verzamelen van data. We geloven dat onze voorgestelde methoden en bijbehorende resultaten toekomstige onderzoekers zullen aanmoedigen om zich te concentreren op het type MIR-taken waarin de focus ligt op hoe muziek de luisteraar raakt. We hopen ook dat de in dit proefschrift verkregen inzichten ontwerpers en ontwikkelaars zullen inspireren om vernieuwende gebruikersinterfaces te bouwen die effectieve toegang tot muziek mogelijk zullen maken.

1

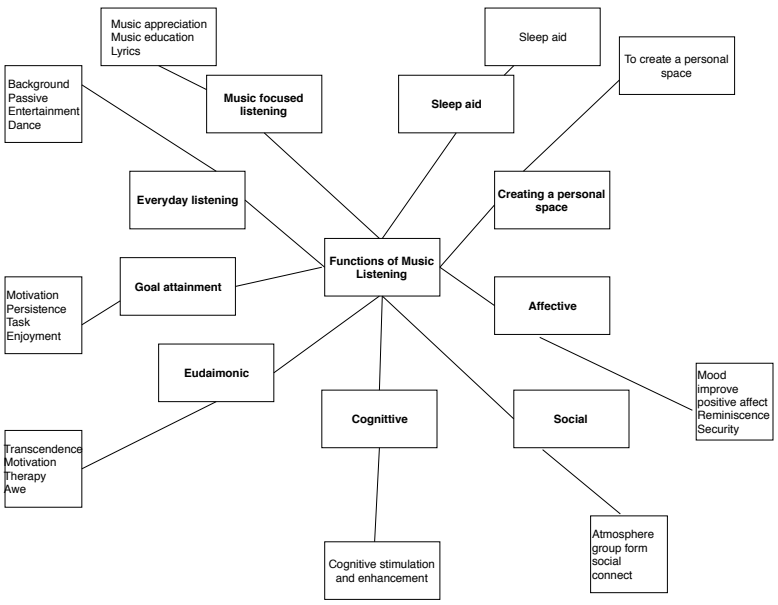
INTRODUCTION

1

1.1. THE MANY VALUES OF MUSIC

The ease of availability of music through various portable devices and online music streaming services (e.g., Spotify, Pandora, YouTube) has led to an increase in music consumption. A recent survey [1] indicates that music is the second most consumed type of media after video. Consumption of music has also increasingly broadened in scope, addressing a wide range of applications and contexts [2] and revealing a plethora of functions of music listening [3], as illustrated in Figure 1.1.

Figure 1.1: Various functions of music listening [3]



Compared to the past where music-focused listening and everyday listening dominated as the ways of consuming music, more and more insight has been gained regarding other values music could bring to a listener. For instance, music can help a person achieve a goal while performing an activity. An example of this is to help a person focus while studying, or improve and maintain the motivation while working out. Hence the "Music in Use" title of this thesis; in this thesis, we will illustrate how we can incorporate this information in expanding Music Information Retrieval (MIR) research.

In parallel with the increasing awareness of broader value that music could have for listeners, the research on the tools for automatically analysing music has gained tremendous momentum over the past decade. This led to a rapid development of the interdisciplinary MIR research field. Observing the research in MIR in general [4], one can say that most of the research has been focusing on extracting and understanding the information from a music signal and investigating a variety of ways to interpret a music signal. The tasks mostly addressed by this research are semantic filtering (e.g. event detection), annotation (e.g., auto-tagging [5][6][7][8][9], structure segmentation [10][11][12][13]), clas-

sification (e.g., genre [14][15]/instrument [16][17]) and search (e.g., "give me more music that is similar to this music track" [18][19]). From the perspective of the scheme in Figure 1.1, this research has been instrumental in facilitating mainly the traditional music-focused listening and everyday listening as modes of music consumption.

The main question underlying the research reported in this thesis is how we can incorporate the other functions of music that can broaden its usefulness in terms of the effect it has on the listeners. This perspective has garnered little attention from the MIR research community so far, with only a few researchers in the music emotion recognition field focusing on it [20] [21] and an exploratory study on music usage [22].

In order to provide an answer to the above question, we investigated how we can expand MIR research to address other functions in Figure 1.1 than the two mentioned before. In this investigation, we relied on insights from the fields of psychology and neuroscience to build machine learning algorithms operating on music signals, that can make music more useful to people. We note that the work reported in this thesis is not a validation of the various psychological theories on the functions of music listening; instead, in our work, we use long-established theories as a motivation to design and build our algorithms.

1.2. MOVING FROM WHAT MUSIC IS TO WHAT MUSIC DOES

In the context of this thesis, we group the tasks in MIR as illustrated in Figure 1.2. The first two columns illustrate the tasks focusing on extracting low- or semantic-level information from music signals. The third and fourth column respectively, address the tasks that we refer to as "Affective" and "Music as Technology" tasks. Looking at the columns, as we move from left to right in Figure 1.2, we see a gradual transition from "what music is" to "what music does" to a listener. As indicated above and in [23], the previous research in MIR has focused mainly on the first two columns, and lately also on the third one. The least addressed is the "Music as Technology" column, which also defines the scope of the research reported in this thesis.

In order to provide better understanding of what this fourth column stands for, we note that event detection in music had so far been restricted to low-level events like note onsets. There is little to no research in terms of detecting events at a higher abstraction level. There is substantial research in identifying structural boundaries in a music track [24] [25] [26], but it is still not exactly event detection. We attempt to detect such high-level events that are recognisable by the listeners (Chapter 2.1 and 2.2).

Additionally, we also note that many of the online music streaming services offer playlists to cater to different situations. Here are a few examples from YouTube (often rated as the most used music streaming service [27]):

1. Music to help listeners to concentrate on their work on YouTube [28].

2. Music to help listeners to workout [29].

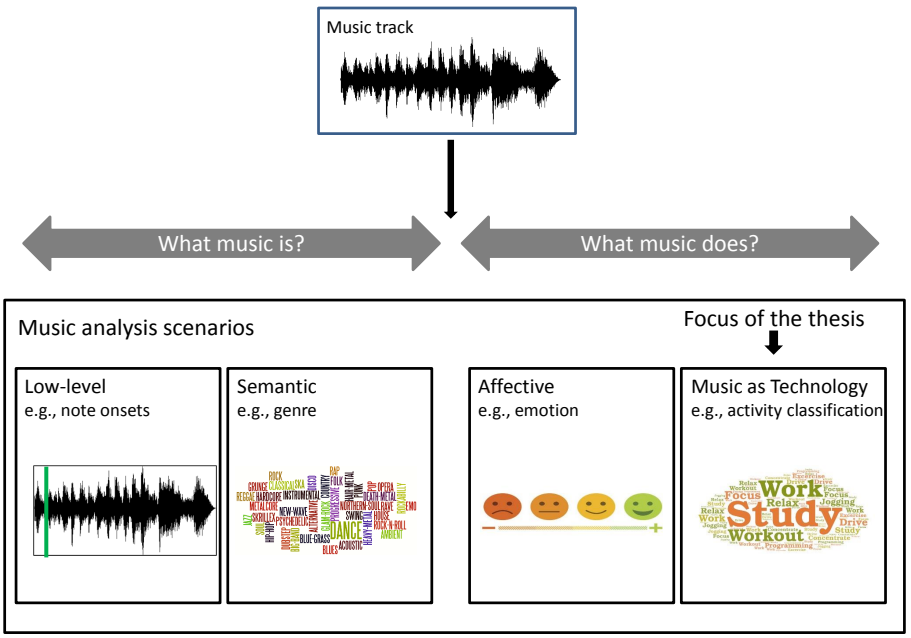
3. Music that can be played in a restaurant while people are having their dinner [30].

Such mixes/playlists are available on multiple platforms (e.g., Spotify, Pandora, Google Play, Focus@will, Brain.fm) and attract a lot of attention. Increasing availability and pop-

ularity of these options is an indication that listeners indeed use music as a tool ("music as technology") to accomplish another activity.

The research reported in this thesis was inspired by the many drawbacks to the existing "music as technology" services/playlists (Chapters 3 & 4). One of these disadvantages is that most of the playlists are human-curated or the music is electronically remastered. Studies in music listening behaviour indicate that people prefer self-created music playlists to the ones automatically created or generated by another human, and would like to have more control on the playlist creating process [2]. We therefore focused on addressing these drawbacks and investigating new methods and algorithms that can help listeners in their search for music to use as a tool in a given situation.

Figure 1.2: Classification of Music Information Retrieval tasks



1.3. CONTRIBUTIONS OF THIS THESIS

In view of the information provided above, we can now reformulate the main question underlying the research of this thesis as follows: *How can we extend the MIR research from analysing "what music is" and develop tools for automatically discovering "what music does", thereby increasing the value of music to people?*

We searched for answers to this question by developing MIR methods and algorithms that could potentially help listeners who use music as a tool to accomplish another ac-

tivity. Specifically, we focused on the scenario in which people use music to get them through common daily activities like relaxing, studying, working and workout. As a case study to start the investigation on music as technology, we worked in Chapters 2.1 and 2.2 on detecting socially significant events in Electronic Dance Music (EDM). These events are at a higher abstraction level than those typically targeted in the MIR literature and serve to increase the (emotional) effect of EDM on the audience. We refer to these events as "socially significant" because they are popular in social media circles, implying that they are readily identifiable and contribute to a large extent to how listeners experience a certain music track or music genre. In addition to being popular, these events affect the listeners by eliciting explicit emotional reactions on social media. In our investigation, we identified three events of particular interest in our Socially Significant Music Events dataset: Drop, Build, and Break. These events can be considered to form the basic set of events used by EDM producers [31]. What makes the detection of these events difficult, is their strongly varying temporal structure and complexity. Our initial work on music event detection played a significant role in developing our subsequent research directions.

In the spirit of using music as a tool to accomplish another activity, we then investigated in Chapter 3 the possibility of classifying music in the categories suitable for different activities. Unlike the common practice and previous work [32], we did not predefine the activity classes ourselves. Instead, we resorted to the most commonly used music streaming service, YouTube [27], to tell us what activities are the most common for which music is sought. Through a data-driven approach, we identified the three most common activity categories: Relax, Study, and Workout. Once we identified the activity categories, we then looked at the possibility of classifying music for each of them using existing metadata like genre, instrument, and artist. Our empirical results indicate that this metadata is not sufficient for classifying music for different activities. We then moved onto exploring the content-based classification of music using low-level and high-level features.

It often happens that one is listening to a particular music track while working on a task and it starts out fine. Suddenly, something happens in the music and one needs to skip/change the track in order to continue working. We call the moment at which a track becomes unsuitable for working a "derail" moment. Inspired by an end-user application, which can automatically skip to the next song when there is an approaching derail moment in the current track, we investigated in Chapter 4 the possibility for building a derail moment detector. Additional inspiration comes from social media sharing platforms like YouTube, where users can leave comments about the tracks. As an example, people leave comments about music tracks titled as being instrumental, but in which they encounter vocals they found to be disturbing for studying [33]. The biggest challenge in detecting derail moments in music is to discover the what constitutes such a moment. In order to get more insight into this, we relied on literature from psychology and neuroscience, but also on the information acquired from a large number of users via the Amazon Mechanical Turk (AMT), an online crowdsourcing platform. Building upon the insights from AMT and literature, we developed a method to automatically detect a derail moment in a music track.

Another important contribution of this thesis is that we have made the datasets, used

in our research, publicly available. We strongly believe and hope that releasing the annotated data would encourage researchers to build upon our research and develop innovative user-oriented applications. For event detection, we provide the IDs of music tracks from SoundCloud and the corresponding timed comments mentioning the events (Music events dataset). For our subsequent work on identifying music for common activities, we collected a lot of YouTube mixes for the following three activities: studying, relaxing, and working out. We have made the unique IDs of these YouTube tracks available online (Music for activities dataset). A significant element of this dataset is that the mixes are long and from a variety of genres, providing a wealth of information for researchers to carry out varied experiments. Similarly, we also released the dataset we used for evaluating our method to automatically detect a derail moment in music tracks (Derail moment dataset). As a part of this dataset, we released the IDs of YouTube music tracks and the corresponding annotations provided by workers on Amazon Mechanical Turk.

1.4. HOW TO READ THIS THESIS

For the technical part of the thesis, original publications have been adopted as chapters 2.1, 2.2, 3, and 4. The references to the corresponding publications are given in the footnote at the beginning of each chapter. Since some of the papers have appeared in conferences and some in scientific journals, the length and depth of the chapters also varies accordingly. Since we retained the original form of the publications, there may be variation in the notation and terminology across the chapters. Also, if chapters address the same general topic, there may be similarity in the motivation, argumentation and some of the material (e.g., sections on related work) they cover.

REFERENCES

[1] *Millennials' media consumption habits: Tv, music, social media, and ads,* `https://www.marketingprofs.com/chirp/2017/31633/millennials-media-consumption-habits-tv-music-social-media-and-ads-infographic` (2017).

[2] M. Kamalzadeh, D. Baur, and T. Möller, *A survey on music listening and management behaviours,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2012).

[3] T. Schaefer, P. Sedlmeier, C. Städtler, and D. Huron, *The psychological functions of music listening,* in *Frontiers in Psychology* (2013).

[4] J. Kepler, *Music information retrieval : Recent developments and applications,* (2014).

[5] M. Kaminskas, F. Ricci, and M. Schedl, *Location-aware music recommendation using auto-tagging and hybrid matching,* in *Proceedings of the ACM Conference on Recommender Systems* (2013).

[6] S. Wang, J. Wang, Y. Yang, and H. Wang, *Towards time-varying music auto-tagging*

based on cal500 expansion, in *Proceedings of the IEEE International Conference on Multimedia and Expo* (2014).

[7] K. Ellis, E. Coviello, A. B. Chan, and G. Lanckriet, *A bag of systems representation for music auto-tagging,* in *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 21 (2013) pp. 2554–2569.

[8] L. Barrington, D. O'Malley, D. Turnbull, and G. Lanckriet, *User-centered design of a social game to tag music,* in *Proceedings of the ACM SIGKDD Workshop on Human Computation* (2009).

[9] P. Lamere, *Social tagging and music information retrieval,* in *Journal of New Music Research*, Vol. 37 (Routledge, 2008) pp. 101–114.

[10] M. Levy and M. Sandler, *Structural segmentation of musical audio by constrained clustering,* in *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 16 (2008) pp. 318–326.

[11] R. J. Weiss and J. P. Bello, *Unsupervised discovery of temporal structure in music,* in *IEEE Journal of Selected Topics in Signal Processing*, Vol. 5 (2011) pp. 1240–1251.

[12] O. Nieto and T. Jehan, *Convex non-negative matrix factorization for automatic music structure identification,* in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (2013).

[13] R. B. Dannenberg and M. Goto, *Music structure analysis from acoustic signals,* in *Handbook of Signal Processing in Acoustics*, Vol. 1 (2008) pp. 305–331.

[14] A. Rosner and B. Kostek, *Automatic music genre classification based on musical instrument track separation,* in *Journal of Intelligent Information Systems*, Vol. 50 (2018) pp. 363–384.

[15] N. Scaringella, G. Zoia, and D. Mlynek, *Automatic genre classification of music content: a survey,* in *IEEE Signal Processing Magazine*, Vol. 23 (2006) pp. 133–141.

[16] Y. Han, J. Kim, K. Lee, Y. Han, J. Kim, and K. Lee, *Deep convolutional neural networks for predominant instrument recognition in polyphonic music,* in *IEEE/ACM Transactions on Audio, Speech and Language Processing*, Vol. 25 (2017) pp. 208–221.

[17] O. Slizovskaia, E. Gómez, and G. Haro, *Musical instrument recognition in user-generated videos using a multimodal convolutional neural network architecture,* in *Proceedings of the ACM International Conference on Multimedia Retrieval* (2017).

[18] D. F. Silva, C. M. Yeh, Y. Zhu, G. E. A. P. A. Batista, and E. Keogh, *Fast similarity matrix profile for music analysis and exploration,* in *IEEE Transactions on Multimedia*, Vol. 21 (2019) pp. 29–38.

[19] P. Knees and M. Schedl, *Music retrieval and recommendation: A tutorial overview,* in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval* (2015).

[20] A. Aljanaki, F. Wiering, and R. C. Veltkamp, *Emotion based segmentation of musical audio,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2015).

[21] Y. E. Kim, E. M. Schmidt, and L. Emelle, *Moodswings: A collaborative game for music mood label collection,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2008).

[22] X. Hu, J. Downie, and A. Ehmann, *Exploiting recommended usage metadata: Exploratory analyses,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2006).

[23] A. Demetriou, M. Larson, and C. C. S. Liem, *Go with the flow: When listeners use music as technology,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2016).

[24] M. Müller, *Fundamentals of music processing: Audio, analysis, algorithms, applications,* (Springer International Publishing, Cham, 2015) pp. 167–236.

[25] E. Quinton, K. O'Hanlon, S. Dixon, and M. Sandler, *Tracking metrical structure changes with sparse-nmf,* in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (2017).

[26] G. Sargent, F. Bimbot, and E. Vincent, *Estimating the structural segmentation of popular music pieces under regularity constraints,* in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 25 (2017) pp. 344–358.

[27] *Music consumer insight report 2016,* https://www.ifpi.org/downloads/Music-Consumer-Insight-Report-2016.pdf (2016).

[28] *Concentration music- concentration music for working fast- concentration and background music,* https://www.youtube.com/watch?v=7HeVWOnju-Y (2015).

[29] *Motivation music workout motivation music 2018,* https://www.youtube.com/watch?v=S9Dg0yFWhBc (2018).

[30] *Dinner music and dinner music playlist: Best 2 hours of dinner music instrumental,* https://www.youtube.com/watch?v=CHjJ3vGmoyY (2016).

[31] B. M.J., *Unlocking the Groove: Rhythm, Meter, and Musical Design in Electronic Dance Music*, 1st ed. (Indiana University Press, 2006).

[32] X. Wang, D. S. Rosenblum, and Y. Wang, *Context-aware mobile music recommendation for daily activities,* in *Proceedings of the ACM International Conference on Multimedia* (2012).

[33] *Relaxing instrumental house music for studying 2015,* https://www.youtube.com/watch?v=NPyiLkNf_0c (2015).

2.1

DETECTING SOCIALLY SIGNIFICANT MUSIC EVENTS USING EXPERT ANNOTATIONS

In this chapter, we focus on event detection over the timeline of a music track. Such technology is motivated by the need for innovative applications such as searching, non-linear access and recommendation. Event detection over the timeline requires time-code level labels in order to train machine learning models. We focus on three events, which are socially significant*, and play a key role in a music track's unfolding and are popular in social media circles. These events are interesting for detection, and here we leverage the annotations provided by experts manually listening to all the tracks. The conclusions we draw during this study provide useful insights that motivates further research in the field of event detection.*

This chapter is part of a paper published in the IEEE Transactions on Multimedia [1].

2.1

2.1.1. INTRODUCTION

Event detection in multimedia is an important field of research and has many applications, especially with the fast growing popularity of multimedia on the web. It has been extensively studied in the context of videos, where currently a broad set of event categories at various levels of semantic complexity can be detected [2]. Research on event detection in music has, however, so far focused mainly on topics like onset detection [3], music structure segmentation [4] and auto-tagging [5].

In this chapter, we look at the problem of event detection in music from a different perspective, guided by two fundamental questions:

1. What events are most interesting to detect?

2. How to detect these events effectively?

Answering these questions can be approached guided by the following consideration. A machine learning approach to event detection typically requires a large number of labels in order to train machine learning models [6]. In this chapter, we focus on providing the necessary background information on event detection and then propose a method to automatically detect these events on the timeline of a music track.

We focus on the domain of electronic dance music (EDM) as a testbed for developing and evaluating our approach. This domain is interesting for investigation due to a number of socially significant event categories, as elaborated in more detail in Section 2.1.2. We discuss the related work in Section 2.1.3, and then proceed towards explaining our approach and its methodological steps in Section 2.1.4. We present an analysis of our dataset in Section 2.1.5, while the experimental setup and results of the method are described in Section 2.1.6.

2.1.2. CASE-STUDY: EVENTS IN EDM

Electronic Dance Music (EDM) is an umbrella term for different genres of electronic music, like Techno, Dubstep, House, Electro. Producers of EDM tracks use different musical elements, like beat, tempo, sound energy or loudness, to shape the music tracks and the events occurring in them. For the purpose of this chapter, we use the following set of events: *Break*, *Drop* and *Build*. They are defined as follows [7]:

- Break: A section in an EDM track with a significantly thinner texture, usually marked by the removal of the bass drum.

- Drop: A point in the EDM track, where the full bassline is re-introduced and generally follows a recognisable build section.

- Build: A section in the EDM track, where the intensity continuously increases and generally climaxes towards a drop.

These events can be considered to form the basic set of events used by the EDM producers [7]. They have a certain temporal structure internal to themselves, which can be of varying complexity. Their social significance is apparent from the presence of a large number of timed comments, related to these events, on SoundCloud. Listeners

react to these events after they occur, or anticipate these events and react to them even before they occur. As an example of the latter case, the timed comment in this track[1] with the text "*Here comes the drop*" comes at the timestamp 00:50, while the actual drop happens at 01:00.

2.1.3. RELATED WORK

In this section, we provide an overview of the previous work related to audio event detection. Here, we explain to which extent we rely on the state-of-the-art, and what is new in our approach.

2.1.3.1. AUDIO EVENT DETECTION

Research related to audio event detection can broadly be divided into three categories: environmental sound recognition, music event detection and music structure analysis. Environmental sounds that can be detected in a given audio stream include, for example, bell ringing, applause, footsteps or rain. Various features and learning methods have been proposed to model the typically non-stationary characteristics of the environmental sounds [8]. We mention here as an example the usage of image processing techniques on a spectrogram image, as proposed in [9], for this purpose. These events typically come from a different acoustic source other than the background audio, while in our case, the musical events in question are part of the continuous music stream. In this chapter, we use the same spectrogram image to extract features. In addition to the spectrogram image, we also explore other image representations: self-similarity matrix, auto-correlation matrix. Some other methods look specifically for the presence of speech in a given audio stream [10]. Given an audio stream, such methods also try to locate segments that contain speech and also identify attributes of speech like fricatives or non-fricatives [11], [12]. Speech related event detection in audio supports automatic speech recognition.

Event detection in music has generally focused on detecting low-level events like onsets [3]. Music onset detection is a well-studied problem in music information retrieval (MIR) and it serves as a task in the MIREX benchmark evaluation every year. Another way of approaching music event detection is music auto-tagging [5], which assigns descriptive tags to short segments of music. It is also addressed by a task in MIREX, under the name *Audio Tag Classification*[2], where descriptive tags needs to be associated with 10-second music segments. These tags generally fall into three categories: musical instruments (guitar, drums, etc.), musical genres (pop, electronic, etc.) and mood based tags (serene, intense, etc.).

In music structure analysis [13], the objective is to divide a given piece of music into its various sections and later group them based on their acoustic similarity. It is an important task since structural elements give to a piece of music its identity. For example, in popular music tracks these structural elements could be the intro, the chorus, and the verse sections. Different aspects of musical expression have been deployed for analysing the musical structure, such as homogeneity (e.g., in instrumentation), repeating patterns

[1] Link active if viewed online.

[2] http://www.music-ir.org/mirex/wiki/2015:Audio_Tag_Classification

2.1

(e.g., in rhythm or melody) and novelty (e.g., through a change in tempo or tonality).

Regarding temporal analysis of the music track and event modelling using audio-visual features, in our approach we largely build on the state-of-the-art methods discussed above, as explained in more detail in Section 2.1.4.3. Specifically, we deploy existing structure segmentation methods that give us an indication of the probable position of events and we use this information to distinguish between event and non-event segments. For feature extraction and event modelling, we build on spectrogram-based signal representation and on a number of proven audio features.

2.1.4. PROPOSED FRAMEWORK FOR EVENT DETECTION

We propose a machine learning algorithm that learns a model per event category, which will later be used to detect the event in a new track. We apply this algorithm to our three events of interest: drop, break and build. In addition to predicting whether an event occurs in a music segment, we also locate the start point of the event.

Figure 2.1.1 illustrates our approach and its main methodological steps. The stage of "Filters" in the highlighted part of Figure 2.1.1 is to filter the noisy timed comments and pass only the selected timed comments to the training stage. In this chapter, we concentrate on building a method relying only on expert annotations and the method using the timed comments is explained in detail in the next chapter (Chapter 2.2).

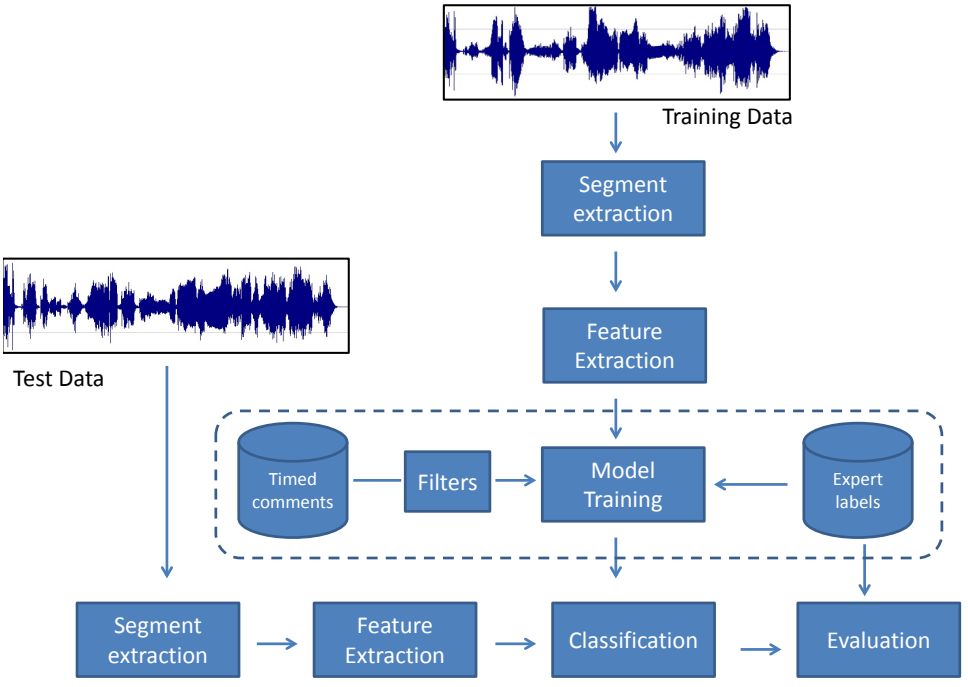


Figure 2.1.1: A schematic view of the different steps in our approach. Note the two different sources of labels: timed comments and expert labels. Changes occur within the part of the model enclosed by the dashed line depending on the source of training labels used.

2.1.4.1. SEGMENT EXTRACTION

In this step, we use two different strategies used to to obtain a unit of classification: Music structure segmentation (MSS) and Fixed-length segmentation (FLS). For MSS, we perform music structure segmentation on the music track and then extract fixed length classification windows centred at the segment boundaries. These windows are the unit that is used further for feature extraction, training, and prediction. The motivation behind choosing to perform structure segmentation is that the structural boundaries in a track can potentially give us start point of the events. For example, a break is a part of an EDM track where the texture is considerably thinner compared to the rest of the track. We hypothesise that the point where the texture becomes thin will be associated with a structural boundary, and for this reason we take our unit of classification to be a window around this boundary. This hypothesis that music events occur at or near boundaries is validated later with an analysis of the dataset in Section 2.1.5.1. Exploratory experiments indicated that the music structure segmentation method proposed in [4] gives a good first approximation of the event positions in an EDM track, when compared to other segmentation methods proposed in [14] and [15]. For this reason, we use the method of [4] for MSS.

For FLS, we divide the track into fixed length segments of duration t seconds with an overlap of $t/2$ seconds between successive segments. Here, we use the full segment of t seconds as the classification unit, unlike MSS where we extract a classification window after segmentation. For this strategy, we do not have the prior knowledge provided by MSS, which means that when we use it our event detection approach becomes comparable to music auto-tagging. Figure 2.1.2 illustrates the two different segmentation strategies.

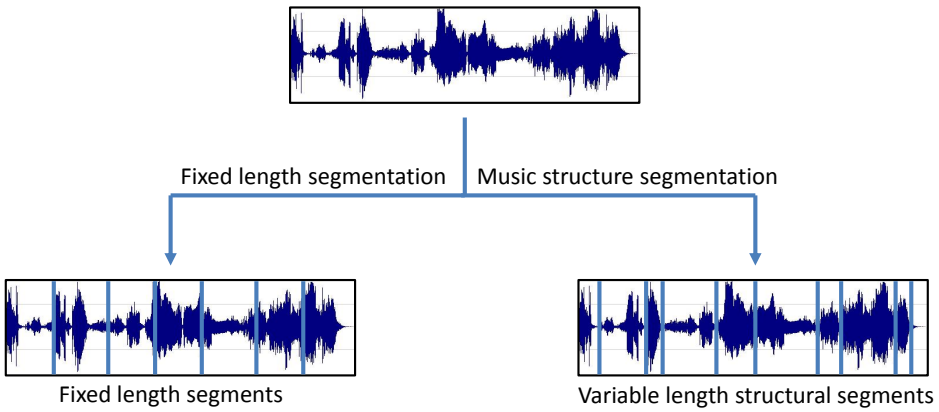


Figure 2.1.2: Two different strategies for segmenting a music track: structure segmentation and fixed-length segmentation.

2.1.4.2. STRATEGIES FOR DEPLOYING TRAINING LABELS

We have the timestamps of our three events of interest from two different sources: experts and timed comments (the procedure to acquire these labels is explained in detail in Section 2.1.5). Each segment coming from the segment extraction algorithm is given two labels depending on whether the timestamp given by an expert or a timed comment falls within the segment. We use four different strategies to obtain a trained model: training using expert labels (EL), training using timed comments (TC), training after combining expert labels with timed comments (CELTC) and training after combining expert labels with filtered timed comments (CELFTC). Expert labels are gold standard labels that can be relied upon and timed comments serve as weak labels. The part of Figure 2.1.1 enclosed by the dashed line changes based on which of the above strategies we use for training.

In the EL strategy, we label a segment as a positive example for an event if an expert label falls within the segment, while the other segments are taken as negative examples. Recall that segments here refer to the classification window extracted around the structural boundary for MSS and the whole segment of t seconds for FLS. We consider this strategy (EL) to be the best possible scenario because we have labels given by experts and the model trained on these labels should be able to make a reliable prediction. We take the performance of this strategy as an upper limit and refer to the EL strategy as the baseline event detector (Section 2.1.6.3). Other strategies (TC, CELTC and CELFTC) are deemed successful if their performance is close to the performance of the baseline event detector. These strategies are explained in detail in Chapter 2.2.

2.1.4.3. FEATURE EXTRACTION

The input to the feature extraction module is a fixed-length music segment (obtained from the following two strategies: MSS and FLS) and the output is a feature vector, which is then used for training a model. We explored image and audio information to choose what features to extract. Here, we provide details about the features from different sources and their corresponding dimensionality.

IMAGE FEATURES

The time-frequency representation of the music signal (spectrogram) has been used in sound event recognition [16]. Figure 2.1.3 shows the pattern representing a drop in the spectrogram. Observing Figure 2.1.3, we can see a sweeping structure indicating the buildup of intensity followed by a sudden drop (red vertical line). We are interested in capturing such patterns, which are unique for certain events in the music. We are not looking for specific frequency values, but rather for patterns that can help us distinguish between music segments containing the event and segments not containing the event. In addition to the spectrogram, we also explore other image representations of an audio signal: auto-correlation and the self-similarity matrix, visualised as images.

In order to calculate image features, we divide each image into rectangular cells of equal size and extract second- and third-order statistical moments from these cells. We divide an image of size 738×927 into 9×9 rectangular cells of size 82×103 to compute the features. We compute the second and third order moments for all three channels: red, green and blue. Moments from cells of each channel are then concatenated to construct

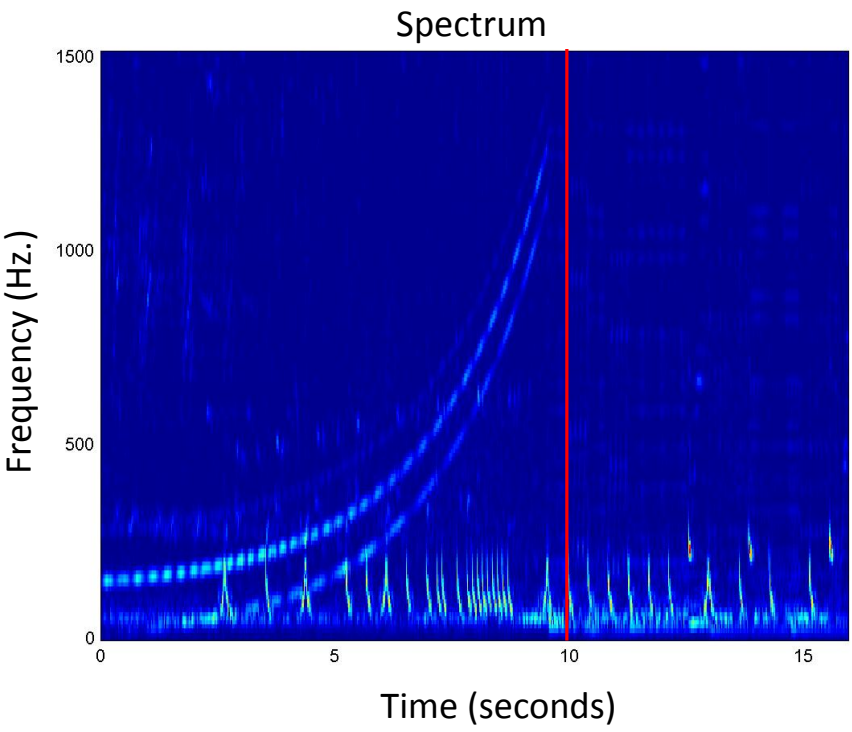


Figure 2.1.3: Spectrogram of a segment containing a *drop*. One can observe a sweep-like structure on the left side of the figure. The red vertical line indicates the position of the drop.

a feature vector with a dimensionality of 486 ($9 \times 9 \times 2 \times 3$), which is further used to train a model. The central moment of order k (m_k) of a distribution is defined as follows: $m_k = E(x-\mu)^k$.

We use the following sets of features with the specified dimensionality: second and third central moments with rectangular cells on spectrogram (486), second and third central moments with rectangular cells on auto-correlation (486), second and third central moments with rectangular cells on self-similarity matrix from spectrogram (486), second and third central moments with rectangular cells on self-similarity matrix from auto-correlation (486).

AUDIO FEATURES

When choosing a set of audio features that will help in distinguishing a segment containing an event and a segment not containing the event, we consider the general characteristics of an audio event and focus on rhythm, timbre and dynamics as feature categories. We use the following features to capture the component of rhythm as explained in [17]: rhythm patterns (RP), rhythm histogram (RH), temporal rhythm histogram (TRH) and

statistical spectrum descriptors (SSD)[3]. In addition to these, we also use other features: tempo (measured in beats per minute), number of beats in a segment, average and standard deviation of the difference between the locations of successive beats[4]. In order to capture the timbral variations, we compute the statistics from the frame-wise MFCC and frame-wise zero-crossing rate (ZCR). The dynamics of the signal change over the course of the build-up towards the drop. To capture these dynamics, we use the statistics (mean, std, var, average of first order derivative, average of second order derivative) computed from the frame-wise RMS energy.

In summary, we use the following set of features with the corresponding dimensionality: RMS energy (5), MFCC (65), ZCR (5), RP (1440), RH (60), TRH (168) and SSD (420).

2.1.4.4. FEATURE SELECTION AND TRAINING

As observed in the previous section, the dimensionality of the features is high and this in-turn could lead to problems like over-fitting or longer training times. In order to avoid such problems, we perform feature selection on the combined features from each of the two modalities (audio and image). We use a feature ranking method, where a score is computed for each dimension of the feature vector and the features are ranked based on this score. We compute the score by measuring the statistical dependency (SD) of the feature values on the corresponding class labels as done in [18]. SD is a measure that quantifies whether the feature values are dependent on the class labels or they co-occur by chance. Since we obtain a ranking of the features using this method, we need to determine which of the top-k features need to be included and we use cross-validation to make this choice.

Another important choice to make is the type of model to use. We choose a Support Vector Machine with a Radial Basis Function kernel because of its discriminative nature, simplicity and wide applicability. Here, we say a few words about why Hidden Markov Models, a common model used for time series data, are inappropriate for our problem. Hidden Markov Models work well for tasks like speech recognition and phonetic segmentation [19]. The strength of HMMs for these tasks is twofold: their ability to predict in the face of the uncertainty of event boundaries (word and phone boundaries) in the speech signal and their ability to model sequence information. In contrast, for our music event detection task, we have a high degree of certainty that an event will be located around a structural boundary. The challenge we face is uncertainty with respect to identification, rather than with respect to segmentation. In our problem, the amount of sequential information is limited to the fact that non-events alternate with events. This information is well captured by our segmentation approach, which also enforces constraints with respect to how closely two detected events can occur to each other. Although HMM architectures can be designed to capture long-distance dependencies, such designs, would come at the cost of an explosion in the number of parameters. Apriori we can anticipate such architectures to be ineffective since they ignore the constraints inherent to the structure of our problem.

With an RBF kernel, there are two parameters, which need to be optimised in an SVM: C and γ. The cost parameter C controls the trade-off between complexity of the

[3] http://www.ifs.tuwien.ac.at/mir/musicbricks/index.html#RPextract

[4] https://acousticbrainz.org/static/download/essentia-extractor-v2.1_beta2-1-ge3940c0-win-i686.zip

decision rule and the frequency of error, while γ is the Gaussian kernel parameter [20]. We perform a grid-search for these parameters using cross-validation and obtain the parameters that give the best performance. We use the cross-validation data set (80% of the data) for this experiment. We carry out a nested cross-validation, which first determines the k to use for selecting the top-k features, and the determines C and γ.

1. Compute SD score for each feature dimension.

2. Pick $k = 50, 100, 150, 200, 250, 300, 350, 400$, where k indicates how many of the top-k ranked features are to be picked for training.

3. For each value of k, follow these steps:

   - Pick the top-k features.
   - Randomly split the cross-validation data into two sets: X_{train} (90%) and X_{val} (10%).
   - Take X_{train} as the new training set and perform cross-validation (grid-search for C and γ) to obtain the best performing model. Use this model to predict labels in X_{val}.
   - Repeat these steps ten times to obtain average validation performance.

4. Choose the k with the best average validation performance.

5. Select the top-k features and perform 10-fold cross-validation on the cross-validation data to obtain the best parameters: C and γ. Now train an SVM on the actual training set using these parameters, which is further used for evaluation.

This procedure is followed while training a model for the four different strategies (EL, TC, CELTC, CELFTC), as explained earlier.

2.1.4.5. CLASSIFICATION

While testing, we follow the same procedure: we first create classification units (using FLS and MSS), which yields a set of segments. We then extract features, and represent each segment using the k features that were obtained while training the model. Using the trained model, we predict labels for the segments. Since we have three events of interest: drop, break, and a build we use three binary classifiers, one for each event. The choice of having three binary classifiers, rather than a single classifier which can predict three classes of events, was made so that we can investigate the utility of timed comments as training labels for each event individually. We train models with four different strategies as explained in Section 2.1.4.4, and predict labels for each test segment. For the models that use MSS, we predict the location of the event to be the mid-point of the segment, which corresponds to a structural boundary in the original segmentation. As we will see in Table 2.1.2, majority of the events start at a segment boundary and hence we use the segment boundary as the start point of the event.

Timestamp	Comment
00:32	That vocal is great.. give everyone goosebump
01:01	Amazing melody
01:28	loved the drop

Table 2.1.1: Examples of timed comments on SoundCloud: text and timestamp.

2.1.5. DATASET AND ANALYSIS

Traditional music tagging datasets like MajorMiner[5] use short music clips and collect labels through crowdsourcing/gamification, while other datasets, like the million song dataset [21], consist of whole tracks and tags collected in the wild on social networks. The focus of this chapter is to build a machine learning model that can localise events on the timeline and we want to achieve this goal while minimising the labelling effort. In contrast to the existing auto-tagging datasets (mentioned above), we need data that provides time-code level labels generated by listeners through social participation. In our work, we therefore rely on SoundCloud as a source of music and the corresponding social data in the form of timed comments. SoundCloud is an online social music sharing platform that allows users to upload, record and share their self-created music. Our goal is to exploit timed comments, which refer to a particular time-point in the track, and could contain useful information about the presence of events. Specific examples of comments from SoundCloud that refer to musical phenomena are given in Table 2.1.1. Using timed comments on SoundCloud as a source also provides an additional advantage over independent labelling of segments: the user has more context to listen to before they react to certain parts of the music track.

We deploy the SoundCloud API[6] to collect our data. Via the search functionality we search for tracks during the year 2014 that have a Creative Commons license, which results in a list of tracks with unique identification numbers. We search the timed comments of these tracks for the keywords: *drop*, *break* and *build*. We keep the tracks whose timed comments contain a reference to these keywords and discard the other tracks.

We use the resulting 500 music tracks to evaluate our proposed method. Most commonly occurring genres in our dataset are the following: dubstep, electro and progressive house. We have a total of 640 drops, 760 builds and 550 breaks in our dataset. These numbers indicate the actual number of events in our dataset i.e., the events are counted based on the expert labels (procedure to obtain expert labels explained later in this section). Associated with the dataset , there are 720 comments with the word “drop”, 750 comments with the word “build” and 600 comments with the word “break”. Note that the statistics indicate the number of timed comments that have a reference to the specific events, meaning that there could be multiple timed comments for a single event posted by different users. We use the timestamps of these timed comments, containing reference to our events of interest, as training labels in the following strategies: TC, CELTC, and CELFTC.

[5] http://majorminer.org/info/intro

[6] https://developers.soundcloud.com/docs/api/guide

Event	0 sec	1 sec	2 sec	3 sec	4 sec	5 sec	6 sec
Drop	80%	1%	0%	1%	1%	0%	1%
Build	56%	4%	6%	2%	2%	3%	10%
Break	60%	10%	5%	2%	4%	6%	2%

Table 2.1.2: Percentage of different events that are $t = 0,1,2,3,4,5,6$ seconds close to structure segment boundaries.

To create the expert labels, we ask a panel of 3 experts to listen to the tracks in the dataset and mark our three events of interest on the timeline of the music track. Each expert marks the events on the timeline of a subset of the music tracks individually. In order to make sure that all the experts have a common understanding of the events and the annotation procedure, we gave them a set of 20 music tracks that are not part of this dataset, but are from the same source (SoundCloud). We ask the experts to mark the events for these 20 tracks and we find that the three experts agree on more than 90% of the annotations. After this check we then ask the experts to mark the timestamps of the events on the timeline of the music tracks. After this process, we have timestamps from two different sources: experts and timed comments, which we employ in our experiments. The dataset, containing the mp3 files, timestamps of the events (both expert labels and timed comments), is hosted on the Open Science Framework and can be accessed here: https://osf.io/eydxk/.

2.1.5.1. STRUCTURE SEGMENTATION

As indicated earlier, we hypothesise that the events would happen in the vicinity of the structural boundaries. In order to validate our hypothesis, we look at the distance between the timestamps of the boundaries and the events in our training set. The training set constitutes 60% of the whole dataset and contains 411 drops, 567 builds and 345 breaks. We perform MSS on the tracks in the training set and obtain the timestamps of the boundaries. On an average, there are 13.6 segments per track in our training set.

The segment boundaries can exactly coincide with the event or can occur in the vicinity of the event. In order to have an estimate of the distance between the event and the segment boundary, we count the number of events at a fixed distance of s seconds, where $s = \{0,1,2,3,4,5,6\}$ and report our observations in Table 2.1.2. For example, if $s = 0$ seconds then we count the number of events which coincide with the segment boundaries. Similarly, if $s = 3$ seconds we count the number of events that are 3 seconds away from a segment boundary. Examining Table 2.1.2, we see that a large portion of the events ($\geq 80\%$) are within a distance of 6 seconds from segment boundaries. It is also interesting that 80% of the drops actually coincide with segment boundaries. These statistics support our hypothesis that the events occur within striking distance (≤ 6 seconds) of the structural boundaries.

2.1.6. EXPERIMENTAL SETUP AND BASELINE

In this section, we explain the experimental setup and report the results of our baseline event detector. Recall that the baseline event detector is trained on expert labels and

serves as a comparison for other proposed strategies (Section 2.1.4.2). We first explain how we split our dataset for the different experiments. We then explain how we tune different parameters in our approach. We also explain our choice of evaluation metrics in this section.

We split our data at the track level into three sets: 60% training data (already mentioned), 20% development data and 20% test data. We do it this way in order to ensure that we do not draw the training and testing material from the same track. This split is used for most experiments.

2.1.6.1. PARAMETERS

In this sub-section, we look at how we choose values for different parameters in our method. We have two different strategies: MSS and FLS. For MSS, we first segment the track and then extract a classification window centred at the segment boundary for feature extraction. The parameter that must be set for MSS is the size of the classification window. We explore the following values: 5, 10, 15, and 20 seconds for the size of the classification window. For each value, we follow the procedure of feature selection and training as explained in Section 2.1.4.4. Using this trained model, we predict the events for tracks in development set and compute the f-scores. By following this procedure, we obtain an optimal performance with 15 seconds as the size of the classification window. For FLS, we divide the track into fixed length segments of duration t seconds and use the entire segment as the classification window. We follow a similar procedure, as discussed for MSS, and obtain an optimal performance on the development data at $t = 15$ seconds.

For the audio features, we use the standard configuration provided by the tools we use for feature extraction. For the image features, we extract the spectrogram for a 15-second music segment by dividing it into 50 ms frames with no overlap. We cap the frequency at 1500 Hz, since we find a clear visible pattern for our musical events below this frequency level. Using MIRToolbox [22], we compute the spectrogram with the above-mentioned parameters and save the result as an RGB image that is further used for feature extraction. Please recall that we divide the image into 9×9 rectangular cells [9], with a cell size of 82×103 and ignore the border pixels on all 4 sides (Section 2.1.4.3). We compute the second and third order moments from the RGB pixel values of each cell and concatenate them to obtain a single feature vector, which is further used in the classification procedure.

2.1.6.2. EVALUATION METRICS

We use different evaluation metrics to understand various aspects of the proposed approach. As indicated earlier, we use two different scenarios: the traditional classification and a use case (non-linear access). For the traditional classification, we use f-score for the positive (fs_+) and negative class (fs_-) as well as the average f-score (fs_{avg}). Since we are also marking the events on the timeline, we assess jump-in points by measuring the distance between start point of the actual event and the predicted event. For this we use two different distance measures: 1. Absolute distance (abs_dist), measured as the difference in timestamps of predicted position and ground-truth; 2. Event anticipation distance (ea_dist), measured as the difference in timestamps of ground truth and the most recent preceding prediction. The distance metric, ea_dist, indicates the useful-

ness of our method in applications like non-linear access, where the user would like to skip to the next event. If there is no previously predicted event, ea_dist chooses the beginning of the track. However, because of the length of EDM tracks and the distribution of events, this situation does not occur in practice. The other distance metric, abs_dist, is only used for the purpose of comparison across the different strategies. Visualisation of the event anticipation distance is illustrated in Figure 2.1.4

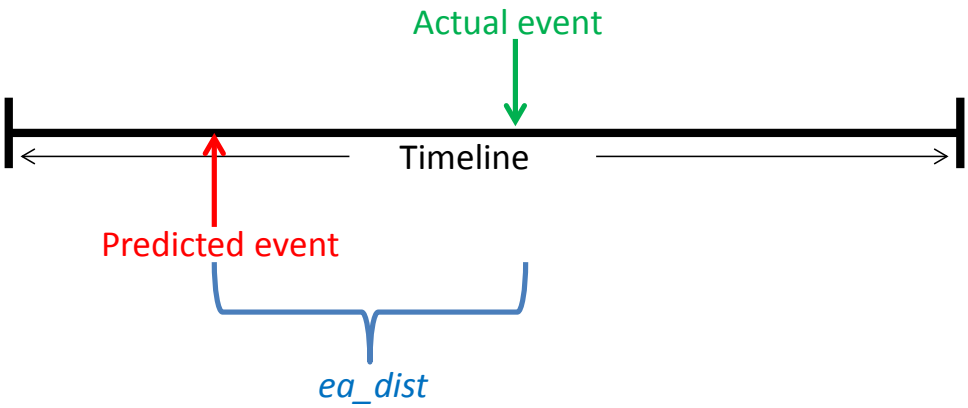


Figure 2.1.4: Visualisation of the event anticipation distance (ea_dist) metric useful to evaluate jump-in points provided to the listener in a non-linear access scenario.

2.1.6.3. BASELINE EVENT DETECTOR

We now report the results of our baseline event detector that uses only expert labels for the entire dataset. Tables 2.1.4 and 2.1.5 report the f-scores: fs_{+}, fs_{-}, fs_{avg}. Similar results are also reported for MSS in tables 2.1.6 and 2.1.7. Observing the scores, we can say that the features extracted from the three image representations (Table 2.1.4 and 2.1.6) perform better than the audio features (Table 2.1.5 and 2.1.7). Of all three events, the scores for detecting the build are lower, which is understandable because it is quite difficult, even for human listeners, to locate the start point of a build.

Here, we also report the number of features that were selected for each event. Table 2.1.3 lists the number of features selected and the top features. We observe that the rhythm-related features dominate the audio features while spectrogram and similarity matrices dominate the image features.

In addition to the f-scores, we also report two other metrics, abs_dist and ea_dist (Tables 2.1.6 and 2.1.7). We report these metrics only for MSS and not for FLS, because the 15-second segments in FLS do not hold any specific meaning while the structural segments in MSS are hypothesised to be the start points of our events of interest (due to Table 2.1.2). Here, it is important to note that ea_dist considers predictions that precede the actual events on the timeline i.e., the predicted start point of the event comes before the actual start point. After manual inspection, we observe that a majority of the detected events precede the actual events. We use the ea_dist metric in order to quantify how close the detection is to the actual event. The values of ea_dist and the above

2.1

Event	*Image features*	*Audio features*
Drop	150, Auto-correlation, Spectrogram, Similarity matrix from spectrogram	200, RP, ZCR, RMS, SSD, MFCC
Break	100, Spectrogram, Similarity matrix from spectrogram	150, MFCC, SSD, RMS, RP
Build	200, Similarity matrices from auto-correlation and spectrogram, Spectrogram	200, SSD, RP, BPM,

Table 2.1.3: Number of selected features and the top selected features.

	fs_+	fs_-	fs_{avg}
Drop	70.3	96.1	83.2
Break	71.6	94.2	82.9
Build	69.8	89.9	79.8

Table 2.1.4: F-scores for the baseline event detector EL: FLS using image features.

findings suggest that we can direct the listener to a few seconds before the actual event is heard. Further analysis and discussion on the significance of ea_dist is presented in Section 2.2.7 (Chapter 2.2).

In this chapter, we discussed our music event detection approach. Our proposed approach was guided by two fundamental questions: What events are interesting to detect? and How can we detect them effectively? In order to scope our research, we considered EDM as our testbed and investigated the interesting events in this genre. We resorted to social music sharing platform - SoundCloud for deciding what events are interesting to detect. By analysing the timed comments on SoundCloud, we found that the following events are interesting: Drop, Build, and Break. We then proposed and evaluated a machine learning algorithm to automatically detect these events in a given music track. One of the important building blocks of our algorithm was identified as the music structure segmentation algorithm and we utilised audio as well as image features for event detection. We trained our models on manually acquired labels from experts and consider this as an ideal situation. We take the results presented in this chapter as a baseline to investigate the utility of timed comments as training labels. In the subsequent chapter, we will present our findings on using timed comments to detect these high-level events of interest (Chapter 2.2).

REFERENCES

[1] K. Yadati, M. Larson, C. C. S. Liem, and A. Hanjalic, *Detecting socially significant music events using temporally noisy labels,* in *IEEE Transactions on Multimedia*, Vol. 20 (2018) pp. 2526–2540.

[2] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah, *High-level event recogni-*

	fs_+	fs_-	fs_{avg}
Drop	68.2	92.3	80.2
Break	69.8	93.1	81.4
Build	67.9	92.4	80.1

Table 2.1.5: F-scores for the baseline event detector EL: FLS using audio features.

	fs_+	fs_-	fs_{avg}	abs_dist	ea_dist
Drop	73.7	97.4	85.5	2.8	2.6
Break	74.4	96.5	85.4	3.1	2.9
Build	70.2	93.1	81.6	3.4	2.9

Table 2.1.6: F-scores and distance metrics for the baseline event detector EL: MSS using image features.

tion in unconstrained videos, in *International Journal of Multimedia Information Retrieval*, Vol. 2 (2013) pp. 73–101.

[3] J. Schlüter and S. Böck, *Improved musical onset detection with convolutional neural networks,* in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (2014).

[4] J. Serrà, M. Müller, P. Grosche, and J. L. Arcos., *Unsupervised music structure annotation by time series structure features and segment similarity,* in *IEEE Transactions on Multimedia*, Vol. 16 (2014) pp. 1229–1240.

[5] H.-Y. Lo, J.-C. Wang, H.-M. Wang, and S.-D. Lin, *Cost-sensitive multi-label learning for audio tag annotation and retrieval,* in *IEEE Transactions on Multimedia*, Vol. 13 (2011) pp. 518–529.

[6] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, *A survey of audio-based music classification and annotation,* in *IEEE Transactions on Multimedia*, Vol. 13 (2011) pp. 303–319.

[7] B. M.J., *Unlocking the Groove: Rhythm, Meter, and Musical Design in Electronic Dance Music*, 1st ed. (Indiana University Press, 2006).

[8] S. Chachada and C. C. J. Kuo, *Environmental sound recognition: A survey,* in *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference* (2013).

[9] J. W. Dennis, *Sound event recognition in unstructured environments using spectrogram image processing*, Ph.D. thesis, Nanyang Technological University, Singapore (2014).

[10] A. Brutti, M. Ravanelli, P. Svaizer, and M. Omologo, *A speech event detection and localization task for multiroom environments,* in *Proceedings of the Joint Workshop on Hands-free Speech Communication and Microphone Arrays* (2014).

[11] J. Li and C.-H. Lee, *On designing and evaluating speech event detectors,* in *Proceedings of Interspeech* (2005).

2.1

	fs_+	fs_-	fs_{avg}	abs_dist	ea_dist
Drop	71.3	94.6	82.9	4.1	3.0
Break	71.1	95	83	4.8	3.9
Build	69.8	87.1	78.4	4.5	3.7

Table 2.1.7: F-scores and distance metrics for the baseline event detector EL: MSS using audio features.

[12] S. Ziegler, B. Ludusan, and G. Gravier, *Towards a new speech event detection approach for landmark-based speech recognition,* in *Proceedings of the IEEE Spoken Language Technology Workshop* (2012).

[13] M. Müller, *Fundamentals of music processing: Audio, analysis, algorithms, applications,* (Springer International Publishing, Cham, 2015) pp. 167–236.

[14] L. Barrington, A. Chan, and G. Lanckriet, *Modeling music as a dynamic texture,* in *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 18 (2010) pp. 602–612.

[15] R. J. Weiss and J. P. Bello, *Identifying repeated patterns in music using sparse convolutive non-negative matrix factorization,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2010).

[16] J. Dennis, H. D. Tran, and H. Li, *Spectrogram image feature for sound event classification in mismatched conditions,* in *IEEE Signal Processing Letters*, Vol. 18 (2011) pp. 130–133.

[17] T. Lidy and A. Rauber, *Evaluation of feature extractors and psycho-acoustic transformations for music genre classification,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2005).

[18] J. Pohjalainen, O. Räsänen, and S. Kadioglu, *Feature Selection Methods and Their Combinations in High-Dimensional Classification of Speaker Likability, Intelligibility and Personality Traits,* in *Computer Speech & Language* (2013).

[19] D. T. Toledano, L. A. H. Gomez, and L. V. Grande, *Automatic phonetic segmentation,* in *IEEE Transactions on Speech and Audio Processing*, Vol. 11 (2003) pp. 617–625.

[20] C. Cortes and V. Vapnik, *Support-vector networks,* in *Machine Learning*, Vol. 20 (1995) pp. 273–297.

[21] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, *The million song dataset,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2011).

[22] O. Lartillot and P. Toiviainen, *MIR in Matlab (II): A Toolbox for Musical Feature Extraction from Audio,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2007).

2.2

Detecting Socially Significant Music Events combining Expert Annotations and Timed Comments

In the previous chapter, we discussed the generic event detection approach using expert annotations. In this chapter, we focus on how we can use timed comments *for detecting the three events of interest. We use* timed comments *from SoundCloud, a modern social music sharing platform, to obtain these labels. While in this way the need for tedious and time-consuming manual labelling can be reduced, the challenge is that timed comments are subject to additional temporal noise, as they are in the temporal neighbourhood of the actual events. We investigate the utility of such noisy timed comments as training labels through a case study, in which we investigate three events of interest in EDM:* drop, build *and* break. *These events are interesting for detection, and here we leverage the timed comments generated in the course of the online social activity around them. In the experiments reported in this chapter, we focus in particular on investigating to which extent noisy timed comments can replace manually acquired expert labels. The conclusions we draw during this study provide useful insights that motivates further research in the field of event detection.*

This chapter is part of a paper published in the IEEE Transactions on Multimedia [1].

2.2.1. INTRODUCTION

Event detection in multimedia is an important field of research and has many applications, especially with the fast growing popularity of multimedia on the web. It has been extensively studied in the context of videos, where currently a broad set of event categories at various levels of semantic complexity can be detected [2]. Research on event detection in music has, however, so far focused mainly on topics like onset detection [3], music structure segmentation [4] and auto-tagging [5].

In the previous chapter, we identified three events of interest in EDM. We then proposed an automatic method to detect these three events in a music track and evaluated the method using a dataset from SoundCloud. To propose an automatic method, we stressed on the importance of structure segmentation [6] and also identified a list of content-based features that can be extracted from the audio signal. The method utilised the expert annotations as training labels to build a machine learning model in the previous chapter. We now turn our focus on how we can make use of the "timed comments" that accompany a music track on SoundCloud.

A machine learning approach to event detection typically requires a large number of labels in order to train machine learning models [7]. Acquiring these labels is expensive and time consuming process, as observed in the previous chapter (2.1). We can, however, benefit from the increasing contextualisation of music in online social communities in order to address this problem. Users listen to music on different social music sharing platforms, such as SoundCloud or YouTube, which allow them to express their opinions/reactions to the music in the form of comments. SoundCloud, for example, offers the possibility to its users to insert *timed comments* while listening to a music track. These comments are similar to usual user comments, however, with an associated timestamp so that they refer to a particular part of the music track. Not only could such timed comments serve as training labels, reducing the need for dedicated manual annotation, but they also allow us to identify the types of events that are interesting for detection in the first place. We refer to such events as being *socially significant*: as a consequence of their recognisably, popularity and anticipation. Listeners talk frequently about them in their comments. In this chapter, we choose to focus on detecting these socially significant events using the "timed comments". As in the previous chapter, we focus on detecting these three events: *build*, *drop*, and a *break* in EDM. For detecting these events, we choose to deploy timed comments as training labels in order to improve the training efficiency.

Usage of timed comments as training labels, however, comes with its own challenges, in particular, the noisy nature of these comments: temporal noise. The timed comment (referring to an event) can occur precisely at the location of the actual event, in the temporal neighbourhood, or far away from the location of the actual event. Figure 2.2.1 illustrates a few possibilities of the distances between the actual event and the corresponding timed comment. Because of their noisy nature, we consider timed comments to be weak labels.

Considering the above-mentioned challenges, we propose an approach using timed comments independently as well as in combination with manually acquired expert labels to build robust machine learning models for detecting socially significant events. Specifically, we aim to answer the following research questions:

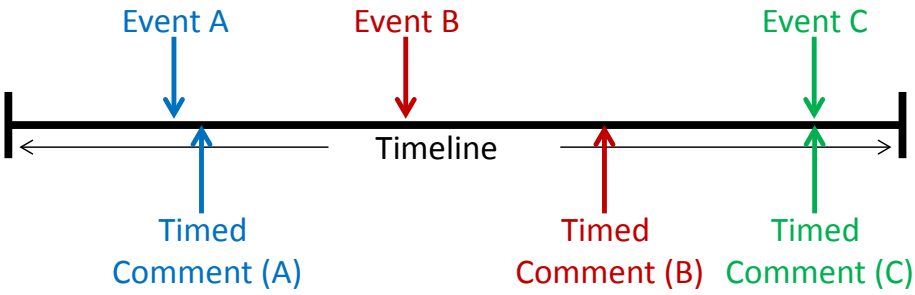


Figure 2.2.1: Timed comments can have temporal noise. A timed comment can be in the temporal neighbourhood of the actual event or precisely at the location of the actual event. Event/timed-comment pairs are in the same colour.

1. (RQ1) Are timed comments helpful in detecting socially significant events?

2. (RQ2) How helpful are timed comments in reducing the number of expert labels needed to train detectors?

To the best of our knowledge, our work is one of the first to use timed comments as a source of training labels for event detection in music. We explain our contributions in Section 2.2.2 and the methodology is explained Section 2.1.4 in Chapter 2.1. The experimental setup and results for the baseline method are described in Section 2.1.6 and Section 2.2.5 presents the overall results. We then explain how the model generalised in Section 2.2.6 and evaluate our method from the perspective of a user application in Section 2.2.7. Finally, we summarise our findings and provide an outlook for further research in Section 2.2.8.

2.2.2. CONTRIBUTION

As reflected by our research questions in Section 2.2.1, the main goal of this chapter is to investigate the usefulness of timed comments as labels for training event detection models in the music audio domain. In order to provide answers to these questions, a framework is needed in which a music track is analysed for the presence of events for which timed comments are available. There, we first identify candidate start points and then select a candidate as the predicted start point of the event using a machine learning step that is trained with noisy timed comments independently. We also combine the timed comments with expert labels. The framework uses music structure segmentation [6]. We build our framework by drawing on previous work where possible and proposing innovations where needed. The link between the previous work and the realisation of our event detection framework is explained in Section 2.2.3.

The framework serves as a vehicle for obtaining insight on the helpfulness of timed comments for event detection. Our findings are communicated in the analysis and discussion of our experimental results in Sections 2.1.6 and 2.2.5. The framework design choices, such as filtering social data based on expert labels, described in Section 2.2.4.1, are made in order to make it possible to answer our research questions.

2.2

In this chapter, we consider the helpfulness of timed comments from two different perspectives, which correspond to two different evaluation scenarios. The first is the signal perspective and this is represented by the conventional performance metric: f-score. We analyse changes in f-score to determine whether we have improved the ability of our approach to detect and exactly localise an event. The second is a user perspective and this reflects the ability of an event detector to support user-facing applications. We choose the application of non-linear access to represent this perspective. A non-linear access system places markers for predicted events on a timeline, which allows a user to jump into the content at a particular time point. The key quantity impacting the user perception of the helpfulness of the event detection is the amount of time a user, who clicks on the marker, must wait in order to encounter an occurrence of the event. We refer to this distance as the event anticipation distance (ea_dist) and use it as an evaluation metric reflecting how users would experience the predicted start points (Figure 2.1.4). Section 2.2.7 further discusses how timed comments and a few expert labels can enable non-linear access.

2.2.3. RELATED WORK

In the previous chapter, we looked at audio event detection (Section 2.1.3.1) and now we look at machine learning with noisy labels, usage of timed comments.

2.2.3.1. MACHINE LEARNING WITH NOISY LABELS

Finding effective ways of dealing with noisy labels is a critical aspect of our machine learning approach. As already mentioned, a segment containing a timed comment referring to an event might not actually coincide with the actual occurrence of that event. Consequences of this temporal noisiness of the labels could be diverse. Noisy labels could decrease classification performance, increase the complexity of the learning models or cause difficulties in identifying relevant features. A detailed survey of different techniques to address the challenge of developing machine learning algorithms in the presence of noisy labels is provided in [8]. We address the issue of noisy labels in two ways. We use different sources of features and also propose strategies to filter the noisy labels.

2.2.3.2. USAGE OF TIMED COMMENTS

Timed comments have been explored in [9] to obtain shot-level tagging of videos. In this work, a topic model is built that can link the audiovisual content of a video shot to the topic of a timed comment. The main difference with our method is that we investigate the association between the timed comments and the signal, while the authors of [9] only analyse the timed comments to achieve video shot-level tagging. A thorough investigation was conducted on timed tags used on an online video platforms in [10], where the authors investigate the differences between timed and timeless tags.

YouTube allows users to mention a timestamp in a comment, which is then converted into a link to that particular part of the video. These comments are called deep–link comments and have been exploited to provide non-linear access to videos [11]. To the best of our knowledge, however, these comments have not yet been deployed for

video event detection. The first attempt to do so in the music domain, which used the timed comments on the SoundCloud platform, was reported in our previous work [12] for the case study of drop event detection. The method presented in this chapter, explained in detail in Section 2.2.4, is an extended and improved version of the work presented in [12]. We note that it was observed in [10] that timed tags for videos are characterised by a phenomenon of temporal noise, which can considered to be comparable to the temporal noise of the timed comments in our music dataset (Figure 2.2.1).

2.2.4. PROPOSED FRAMEWORK FOR EVENT DETECTION

We propose a machine learning algorithm that learns a model per event category, which will later be used to detect the event in a new track. We apply this algorithm to our three events of interest: drop, break and build. In addition to predicting whether an event occurs in a music segment, we also locate the start point of the event.

Figure 2.1.1, in Section 2.1.4, illustrated our approach and its main methodological steps. The stage of "Filters" in the highlighted part of Figure 2.1.1 is to filter the noisy timed comments and pass only the selected timed comments to the training stage.

2.2.4.1. STRATEGIES FOR DEPLOYING TRAINING LABELS

We have the timestamps of our three events of interest from two different sources: experts and timed comments (the procedure to acquire these labels is explained in detail in Section 2.1.5). Each segment coming from the segment extraction algorithm is given two labels depending on whether the timestamp given by an expert or a timed comment falls within the segment. We use four different strategies to obtain a trained model: training using expert labels (EL), training using timed comments (TC), training after combining expert labels with timed comments (CELTC) and training after combining expert labels with filtered timed comments (CELFTC). Expert labels are gold standard labels that can be relied upon and timed comments serve as weak labels. The part of Figure 2.1.1 enclosed by the dashed line changes based on which of the above strategies we use for training.

In the EL strategy, we label a segment as a positive example for an event if an expert label falls within the segment, while the other segments are taken as negative examples. Other strategies (TC, CELTC and CELFTC) are deemed successful if their performance is close to the performance of the baseline event detector. In the second strategy (TC), we label a segment as a positive example for an event if a timed comment referring to that event falls within the segment and the other segments are taken as negative examples. In the other two strategies, we divide the training data into two subsets of m and $N-m$ tracks, where N is the total number of tracks in the training set and $m = p \times N$ represents a proportion of N for $p = \{20\%, 40\%, 60\%, 80\%\}$. For example, if $p = 20\%$ then $m = 0.2 \times N$ and $N - m = 0.8 \times N$ represents a portion of the training data. We use expert labels for the m tracks and use timed comments as labels for the remaining $N-m$ tracks. In CELTC, we directly combine expert labels for the m tracks and timed comments for the $N-m$ tracks to train a model. For CELFTC we use a different approach that includes a step of filtering the noisy timed comments (Figure 2.2.2). More specifically, we train a model using expert labels for m tracks and test if the timed comments from the $N-m$

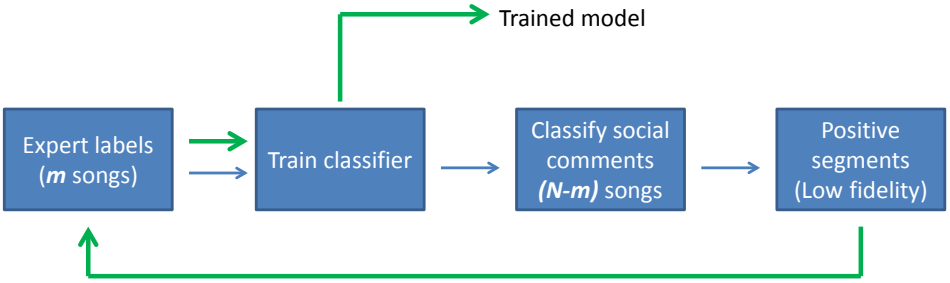


Figure 2.2.2: CELFTC: Pipeline for combining expert labels with timed comments. This strategy involves the step of verifying the timed comments before adding them to the training data. The thicker, green arrows refer to the training after filtering the timed comments.

tracks actually refer to the event. We then take positively classified examples from the $N-m$ tracks and add them to the existing training data labelled with expert labels i.e., m tracks. The training procedure applied to all four strategies using the corresponding sets of training labels is explained in Section 2.1.4.4. In all the four proposed strategies: EL, TC, CELTC, and CELFTC, we use all the positive and negative examples for training i.e., we do not take an equal number of positive and negative examples for training.

2.2.5. EXPERIMENTAL RESULTS

In this section, we report the results of the experiments that help us in addressing the two research questions as introduced in Section 2.2.1. We also introduce a naive event detector that randomly picks segment boundaries as start points of our events of interest.

2.2.5.1. NAIVE DETECTOR

In this sub-section, we describe a naive detector which picks x number of events from each tracks where x is the average number of events in the training set. In our training set, we have 1.4 drops, 1.6 builds and 1.5 breaks per track, on average. We follow these steps for the naive classifier:

- Perform MSS on each track. Recall that there are 13.6 segments, on an average, per track (Section 2.1.5.1).
- Randomly pick x number of segment boundaries as the start points of our three events of interest, where x is as explained above for each event.
- Repeat the above step 10 times to reduce the effect of biases.
- Compute all the evaluation metrics as explained in 2.1.6.2

The performance of the naive detector is reported in Table 2.2.1 and we observe that the average f-scores are very low. We consider the performance of this naive detector as the lower bound and that of the baseline event detector (Section 2.1.6.3) as the upper bound for comparing the proposed strategies (TC, CELTC, and CELFTC).

	fs_+	fs_-	fs_{avg}	abs_dist	ea_dist
Drop	5.9	71.4	38.6	29.1	32.6
Build	4.9	61.4	37.6	28.7	33.4
Break	6.5	68.7	37.6	31.4	34.9

Table 2.2.1: F-scores and distance metrics for the naive classifier: randomly pick x number of events from each track.

	fs_+	fs_-	fs_{avg}
Drop	29.4	60.1	44.7
Break	34.2	59.4	46.8
Build	27.9	58.6	43.2

Table 2.2.2: F-scores for the strategy TC: Timed comments as training labels and FLS using image features.

2.2.5.2. USING TIMED COMMENTS AS TRAINING DATA

We now investigate the utility of timed comments as training labels, which helps us in addressing the first research question (RQ1 from Section 2.2.1). We follow the same procedure as in the baseline event detector, except for the source of labels. We use timed comments instead of expert labels for training our models. Tables 2.2.2, 2.2.3, 2.2.4, and 2.2.5 report the results. Observing the tables, we can say that the timed comments perform very well in comparison to the naive classifier (Table 2.2.1), but not so well when compared to the baseline event detector (Tables 2.1.4, 2.1.5, 2.1.6, 2.1.7). We observe a significant improvement in fs_+, abs_dist, and ea_dist, when compared to the naive classifier. However, we see a decline in f-scores for the negative class. The classifier struggles to identify non-events, which probably have less regularity than events. We surmise that the noisy nature of timed comments makes it even harder to learn non-events. In order to ensure that the classifier is not over hypothesising, we count the number of events that the classifier hypothesises per track. From Section 2.1.5.1, we know that there are 13.6 segments, on average, per track in our training set. Consider the drop event detector, we use a classifier trained on timed comments alone to count the number of segment boundaries that are classified as a drop, in each track of the test set. Then we take an average of the number of drops across all the tracks in the test set. By repeating this process for the other two events, we observe that the classifier hypothesises 3.1 drops, 3.6 builds and 2.6 breaks per track on an average. These numbers are not overly high compared to the actual average number of events per track: 1.3 drops, 1.5 builds and 1.1 breaks. In an application scenario in which the average number of events expected per track is highly stable, the prior information that is used here by our naive classifier could also be integrated into our event detection models. However, here, we will continue to assume a use scenario in which that information is not available, and not add it to our models. We can see that the timed comments are indeed useful in detecting socially significant events and thus we have an answer for RQ1. Now, we will explore the combination of timed comments and expert labels to address the next research question, where we investigate whether the presence of timed comments can reduce the number of expert labels needed to detect socially significant events.

	fs_+	fs_-	fs_{avg}
Drop	27.2	61.5	44.3
Break	30.8	56.4	43.6
Build	29	58.4	43.7

Table 2.2.3: F-scores for the strategy TC: Timed comments as training labels and FLS using audio features.

	fs_+	fs_-	fs_{avg}	abs_dist	ea_dist
Drop	28.1	66.3	47.2	21.5	18.1
Break	33.2	52.1	42.6	24.3	21.2
Build	28.4	59.1	43.7	26.6	22.3

Table 2.2.4: F-scores and distance metrics for the strategy TC: Timed comments as training labels and using MSS using image features.

2.2.5.3. COMBINING EXPERT LABELS AND TIMED COMMENTS

The main contribution of this chapter, as presented in Section 2.2.2, is the investigation of the utility of timed comments as training labels. In the previous sub-section, we saw that using timed comments alone as training labels yielded lower scores because of the noisy nature of timed comments. Here, we investigate how the addition of timed comments used as labels can reduce the number of expert labels needed for detecting socially significant events. We investigate this by performing a series of experiments focusing on the strategies: CELTC and CELFTC, introduced in Section 2.2.4.1. In these strategies, we divide the training data into two subsets of m tracks and $N-m$ tracks, N being the total number of tracks in the training set and $m = p\% \times N$. We use the following values for $p = \{20\%, 40\%, 60\%, 80\%\}$, which controls the proportion of the training data (N) that is used. In CELTC, we directly combine the expert labels for the m tracks and timed comments for the $N-m$ tracks to train our model.

In CELFTC, we train a model using the expert labels on m tracks and use the model to filter the timed comments on the $N-m$ tracks. It is important to note that CELFTC requires more training time than the other strategies because it involves a two-step process of first filtering the timed comments and then re-training the model using the additional data from the filtering step. Since we use the top-k features computed in the first step of the algorithm (Section 2.1.4.4), the additional training time in the second step is not very high. For example, when $p = 60\%$, the overall training time of CELFTC is a mere 6% more than that of CELTC. After filtering the timed comments, we add the positively labelled examples from the $N-m$ tracks to the actual training set of m tracks to build the final model (illustrated in Figure 2.2.2). For each value of m, we repeat the experiment 10 times and report the average results in order to minimise the chance of interference of incidental characteristics of the data.

In order to provide a further basis for comparison, we report the results of training with m tracks (EL@p) i.e., we use only a part of the training data with expert labels corresponding to the value of $p = 20\%, 40\%, 60\%, 80\%$. For example, if p=40%, then we use 40% of the training data with expert labels to train the model. This model then predicts the positions of the events in the test set and we compute the f-scores as usual.

	fs_+	fs_-	fs_{avg}	abs_dist	ea_dist
Drop	23.1	61.2	42.2	29.4	24.6
Break	24.1	59.1	41.6	25.2	20.3
Build	31.1	56.1	43.6	31.2	29.4

Table 2.2.5: F-scores and distance metrics for the strategy TC: Timed comments as training labels and using MSS using audio features.

Tables 2.2.6, 2.2.7, 2.2.9 and 2.2.8 report the average f-scores (fs_{avg}) for each of the strategies (CELTC, CELFTC and EL@p) at different values of p. Similarly, Tables 2.2.10 and 2.2.11 report the distance metrics for each strategy. Observing the tables, we can say that image features are more effective than audio features. Filtered timed comments (CELFTC) perform better than the unfiltered timed comments (CELTC) when combined with the expert labels. This can be observed in the results for CELFTC and CELTC, where the f-scores for CELFTC are higher than those for CELTC. When the CELFTC's performance is greater than that of EL@p, results are highlighted in bold.

Filtering the timed comments (CELFTC) seems to improve the performance beyond just using the expert labels (EL@p) at certain proportions of the training data. For example, the average f-score for detecting a drop using CELFTC, at p = 60% and p = 80%, is greater than that of EL@60 and EL@80% respectively (Table 2.2.6). Similar observations can be made for the break at 60% and 80% of the training data. For the event build, the average f-scores of CELFTC come very close to the f-scores of EL at 80% of the training data. The distance metrics abs_dist and ea_dist reported in Tables 2.2.10 and 2.2.11 indicate that the scores for CELFTC at 60% are very close those for EL at 60%.

2.2

Event	20%			40%			60%			80%		
	CELTC	CELFTC	EL@20	CELTC	CELFTC	EL@40	CELTC	CELFTC	EL@60	CELTC	CELFTC	EL@80
Drop	43.6	45.3	50.2	56.1	61.1	64.2	65.5	**76.1**	72.4	71.6	**81**	78.1
Break	44.2	47.2	58.7	61.7	65.8	69.5	72	**80**	77.8	73.6	**82.6**	81
Build	43.2	43.8	49.3	55.8	58.7	61.3	63.7	**74.1**	73.4	71	**78.2**	77.8

Table 2.2.6: Average F-scores for training using different proportions of expert labels for the three different strategies: CELTC, CELFTC and EL@p. Results are for FLS using image features.

Event	20%			40%			60%			80%		
	CELTC	CELFTC	EL@20	CELTC	CELFTC	EL@40	CELTC	CELFTC	EL@60	CELTC	CELFTC	EL@80
Drop	44.4	45.5	49.1	53.5	56.9	58.15	66.6	**72.6**	70	75.1	**78.9**	76.3
Break	47.2	48.3	52.2	59.3	59.7	61.5	70.3	**77.8**	76.4	76.3	**80**	79
Build	43.8	43.4	46.5	54.9	57.5	59.2	65.2	73.1	74	73	76.5	76.8

Table 2.2.7: Average F-scores for training using different proportions of expert labels for the three different strategies: CELTC, CELFTC and EL@p. Results are for FLS using audio features.

Event	20%			40%			60%			80%		
	CELTC	CELFTC	EL@20	CELTC	CELFTC	EL@40	CELTC	CELFTC	EL@60	CELTC	CELFTC	EL@80
Drop	47.6	48	52.2	62.5	64.9	65.9	73	**75.9**	73.8	81	**83.4**	81.6
Break	49.5	50.1	53.7	69.8	72.1	72.9	78.7	**83.1**	79.7	81.3	**84**	83
Build	44.3	44.6	49.4	59.6	63.8	65.5	70.6	72.8	74.5	75.1	**81**	80.1

Table 2.2.8: Average F-scores for training using different proportions of expert labels for the three different strategies: CELTC, CELFTC and EL@p. Results are for MSS using image features.

Event	20%			40%			60%			80%		
	CELTC	CELFTC	EL@20	CELTC	CELFTC	EL@40	CELTC	CELFTC	EL@60	CELTC	CELFTC	EL@80
Drop	44.2	46.4	51	54	56.3	59.5	65	**74.7**	73	75	**81.4**	78.6
Break	52.2	52.4	54	60.1	61.6	62.5	69.9	**79.4**	78	74.6	**81.7**	79
Build	44.1	44	48.5	56.3	60.2	62.5	63.5	**72.4**	72	71.2	**77.4**	76

Table 2.2.9: Average F-scores for training using different proportions of expert labels for the three different strategies: CELTC, CELFTC and EL@p. Results are for MSS using audio features.

2.2

Event	20%			40%			60%			80%		
	CELTC	CELFTC	EL@20	CELTC	CELFTC	EL@40	CELTC	CELFTC	EL@60	CELTC	CELFTC	EL@80
Drop	19.1,16.2	19.2,15.9	17.4,14.3	14,12	13.7,12	13.4,12.1	14.2,12.2	11.5,9.6	11.1,9.1	10.4,8.3	**8.0,6.1**	8.3,6.4
Break	17.3,15.1	17.1,15.8	15.4,14.6	13,10.2	11.7,10	11.2,10.5	11.6,9.3	9.6,8.7	9.4,7.9	8.6,6.8	7.3,6.4	7.4,6.1
Build	16.1,15.5	16.3,14.6	17.8,16.3	15,13.3	14.4,13.8	13.6,11.8	14.5,12.6	13.6,11.5	11.4,9.5	11.5,9.3	10.6,8.7	8.6,6.3

Table 2.2.10: Distance metrics (*abs_dist* and *ea_dist*) for training using different proportions of expert labels for the three different strategies: CELTC, CELFTC and EL. Results are for MSS using audio features.

Event	20%			40%			60%			80%		
	CELTC	CELFTC	EL@20	CELTC	CELFTC	EL@40	CELTC	CELFTC	EL@60	CELTC	CELFTC	EL@80
Drop	17.4,15.3	16.8,14.9	15.3,12.4	15.6,12.9	13.7,11.1	12.4,9.9	10,8.5	**8.1,7.9**	9.5,8.6	6.3,5.1	**5.9,4.9**	6,5
Break	16.4,14.2	15.9,13.9	14.2,13.1	12.4,10.2	11.8,10.6	10.4,8.7	9,7.6	**7.4,6.9**	8.1,7.8	5.4,4.6	**4.8,3.9**	5.2,4.1
Build	18,15.2	17.8,15.6	16.4,12.8	15,12	14.2,11.9	11.4,10.6	12.4,10.6	10.8,7.1	9.6,6.8	10,8.2	**7.9,6.0**	7.6,6.4

Table 2.2.11: Distance metrics (*abs_dist* and *ea_dist*) for training using different proportions of expert labels for the three different strategies: CELTC, CELFTC and EL. Results are for MSS using image features.

Next, we further investigate the performance of CELFTC, at different proportions of expert labels, by comparing its performance with that of the baseline event detector, which represents an ideal situation. Recall that the baseline event detector was trained with expert labels on the entire training set (Section 2.1.6.3). For the baseline event detector, we choose the following combination for all the events as it was shown to result in the best performance: MSS and Image features. For the same combination, we report the results of CELFTC and also add results for EL@p at different proportions of expert labels. The results are depicted in Figures 2.2.3 (drop), 2.2.4 (build), and 2.2.5 (break). The blue horizontal line in the figures represents the performance of the baseline event detector (Table 2.1.6). Observing the figures, we can see that with 60% of the training data labelled with expert labels we already achieve a performance very close to the baseline event detector, especially for break. For example, observing Figure 2.2.5 at 60%, the performance of CELFTC and the performance of the baseline break event detector are almost the same. For the other two events: drop and build, we observe that with $p = 80\%$, we get a performance equal to that when $p = 100\%$. This is a bit lower when compared to the performance of break detection, but at the same time we should note that drop and build are difficult events to detect. From this result, we can conclude that if we have a training set labelled with expert labels, then, it will improve our classifier to add additional training data labelled with filtered timed comments, so long as we have a minimum amount of expert-labelled data. On this basis of this conclusion, we can say that the timed comments are helping in reducing the number of required expert labels, which represents a positive answer to RQ2.

2.2.6. GENERALISATION OF THE MODEL

2.2.6.1. CROSS-VALIDATION

A 5-fold cross-validation was performed on the cross-validation data (80% of the entire dataset) and the average f-scores and standard deviation are reported in Table 2.2.12. One of the reasons to perform a cross-validation experiment is that the dataset is relatively small and we want to investigate whether the trained model overfits. Results of the cross-validation are good but lower when compared to the ones reported in Tables 2.1.4, 2.1.5, 2.1.6 and 2.1.7.

2.2

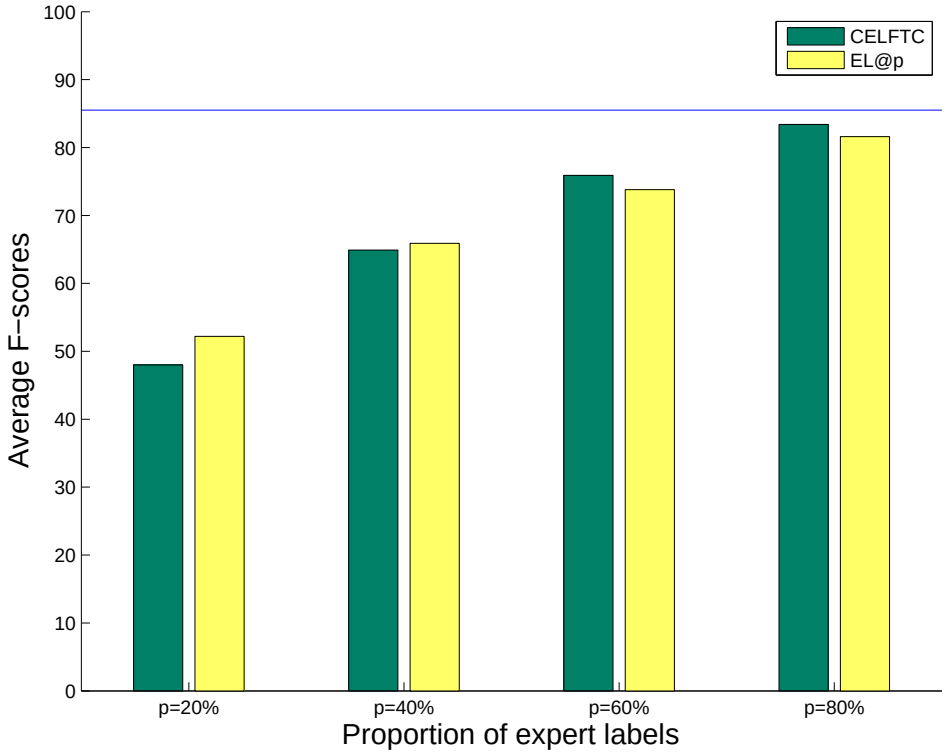


Figure 2.2.3: Average f-scores (fs_{avg}) for detecting a drop for CELFTC: FLS and image features at different proportions of expert labels. The horizontal blue line indicates the performance of the baseline event detector with 100% expert labels.

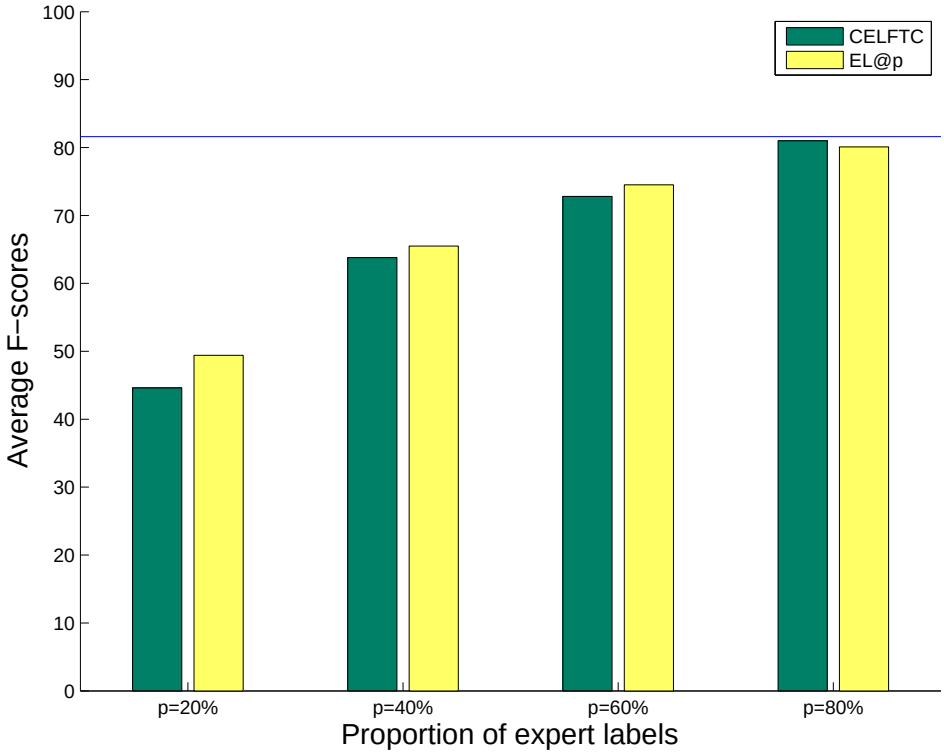


Figure 2.2.4: Average f-scores (fs_{avg}) for detecting a build for CELFTC: MSS and audio features at different proportions of expert labels. The horizontal blue line indicates the performance of the baseline event detector with 100% expert labels.

2.2

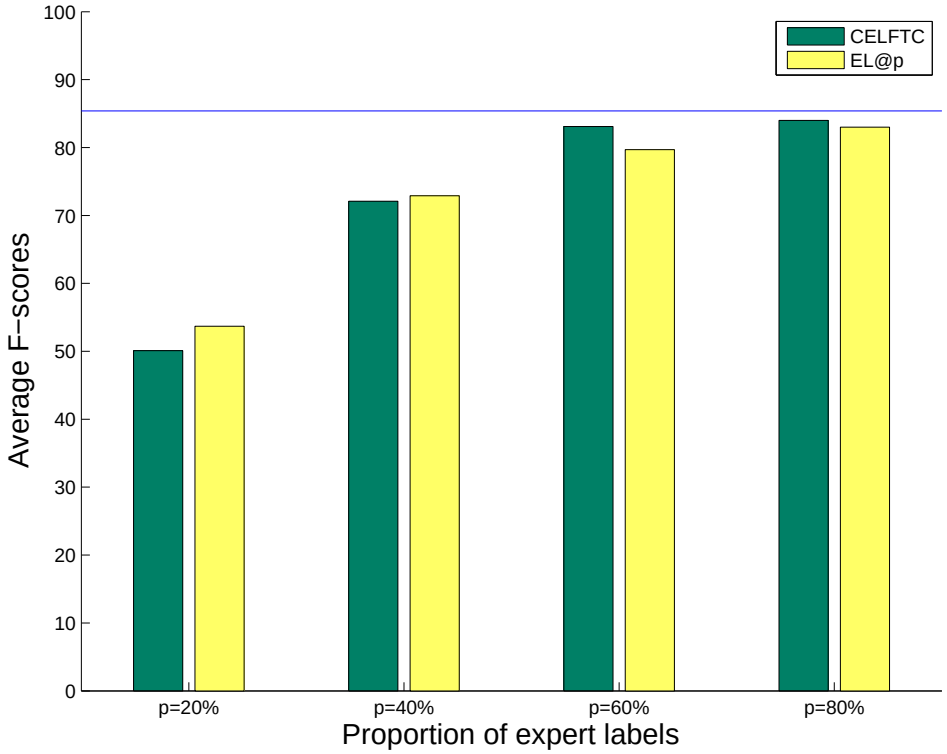


Figure 2.2.5: Average f-scores (fs_{avg}) for detecting a break for CELFTC: MSS and image features at different proportions of expert labels. The horizontal blue line indicates the performance of the baseline event detector with 100% expert labels.

	fs_{avg}(IM,FLS)	fs_{avg}(AU,FLS)	fs_{avg}(IM,MSS)	fs_{avg}(AU,MSS)	abs_dist(IM)	ea_dist(IM)	abs_dist(AU)	ea_dist(AU)
Drop	73.3 (±4.1)	72.2 (±3.2)	77 (±5.3)	74.4 (±4.2)	7.1 (±1.1)	5.2 (±1.2)	6.9 (±1.7)	5.4 (±3.2)
Break	73.2 (±3.1)	71.4 (±4.2)	76 (±4.1)	75.3 (±5.6)	7.2 (±2.8)	5.5 (±2.9)	7.1 (±2.1)	5.6 (±1.3)
Build	71.3 (±5.3)	72.7 (±3.6)	76.2 (±3.2)	74.4 (±5.7)	7.8 (±2.1)	5.7 (±3.0)	7.1 (±1.4)	5.8 (±4.2)

Table 2.2.12: Cross-validation results for our three events. im: image features; au: audio features.

Event	F-score for 60% expert labels	F-score for 100% expert labels
Drop	73.2	76.4
Break	74.9	77.1
Build	71.4	73.5

Table 2.2.13: Average F-scores for CELFTC on data from a new source (YouTube) for different proportions of expert labels.

This effect can be related to our sampling method. For the purpose of cross-validation, the folds are created at the track level, and not at the event level. This is necessary in order to ensure that it is never the case that training and testing material is drawn from the same track. However, the track-level sampling makes the folds sensitive to the presence of one or two tracks with a style of event that is overall more "difficult" (applies in particular to short events). For this reason, the variance between the folds is higher than expected and the average is lower. The lower average raises a question on the generalisation capability of the model and in order to answer this question, we turn to another dataset. Specifically, we next report the results of the experiment on an unseen dataset that provide an insight into the generalisability of the model.

2.2.6.2. PERFORMANCE ON DATA FROM A NEW SOURCE

In order to check for the generalisability of the model, we conduct another experiment where we take the test set from another source. YouTube contains many EDM tracks and can be used as another source of music data. We download 70 tracks from YouTube and manually marked the positions of our three events in the tracks. We use this as the test set and the corresponding ground-truth in order to evaluate the performance of the detector. We chose our best model in order to predict the events on the new test set. We use MSS and image features for evaluation. We use two different trained models that use 60% and 100% expert labels respectively. Table 2.2.13 presents the results of the event detection on the YouTube test set. Please note that we use the same model trained for CELFTC at 60% expert labels (Section 2.2.5.3) and EL with 100% expert labels (Section 2.1.6.3) for the two columns in Table 2.2.13.

Observing the scores, we can see that the performance of the event detector is reasonable and similar trends can be found when compared to the performance on the test set from SoundCloud. For example, the f-scores for both 60% and 100% expert labels are very close together.

2.2.7. EVALUATION WITH USER-PERSPECTIVE METRICS

In this section, we turn to a deeper discussion of the implication of our results for a real-world application. Specifically, we consider a non-linear access system, i.e., a system that would allow a listener to browse through the events in a track. Such a system would involve a play bar in which music events are marked, making it possible for listeners to listen specifically to certain events, without having to listen to the track entirely. For example, such a system would be useful to a DJ who is interested in quickly reviewing all the drops in a particular EDM track.

In order to understand the usefulness of our music event detection approach to users of a non-linear access system, we make use of the metric event anticipation distance, *ea_dist*, introduced in Section 2.1.2, where it is illustrated in Fig. 2.1.4. Recall, that *ea_dist* is the time that a listener would need to wait before jumping into a music stream, and hearing the event that is marked on the play bar. For comparison, we also discuss the absolute distance, *abs_dist*. Note that we do not consider *abs_dist* to be a user-perspective metric, since it has the same value whether the listener is dropped into the stream before or after the event. A music event that occurs *before* a user jumps into a stream will be missed, and can, for this reason, be considered useless in a non-linear access application scenario.

When we consider this application scenario, and *ea_dist*, the full potential of timed comments becomes clear in a way not directly reflected by the f-score that has been the focus of the previous sections. We would like to draw attention to the condition in which the music event detector is trained only with timed comments as training labels and in which MSS with image features is used. This condition was presented in Table 2.2.4 (Section 2.2.5.2). From Table 2.2.4 we see that using timed comments only, we can provide a jump-in point, on an average, 18.1 seconds before the actual drop. We point out that an error of 18.1 seconds may not be substantial enough to impact user experience significantly. Statistics calculated on our dataset as a whole reveals that a typical build-drop combination can last somewhere between 6 and 20 seconds. If we can direct the user to 18.1 seconds before the drop, there is a good chance that the build will have already started and it will be obvious to listeners that they are moving towards the drop. An interesting future research would be to conduct a user study with DJs if this result can already help them in finding these events in a given EDM track.

In the rest of this section, we make some other observations about our results from the perspective of our distance-based evaluation metrics *abs_dist* and *ea_dist*. These results are reported in Tables 2.2.4 and 2.2.5 (training on timed comments only) and Tables 2.2.10 and 2.2.11 (mixing expert labels and timed comments.) Note that in Tables 2.2.10 and 2.2.11 results are given in the order *abs_dist, ea_dist*, separated by a comma. Overall, the image features are more effective than the audio features. This observation is consistent with the observations that we have made using the average f-score in previous sections. Further, we note that *ea_dist* is systematically smaller than *abs_dist*. This observation is interesting, since it means that our approach to music event detection tends to detect an event before it occurs, rather than after it occurs. In other words, it shows a tendency away from the sort of error that would be most detrimental to the user experience.

Finally, we make another observation about Tables 2.2.10 and 2.2.11. We see that in general, if expert labels are available, it is most advisable to train with expert labels. Adding examples labelled with timed comments to the expert-labelled training data can add another performance boost, or at least will not hurt the performance substantially. It is interesting to consider the implications of the performance that can be achieved with a relatively limited number of expert labels. For example, using 60% expert labels we see that ea_dist for the build reaches a value of 8.6 seconds for image features (Table 2.2.11). This value is very close to the minimum length of a build-drop combination, again as estimated by statistics calculated on our dataset as a whole. This example sug-

gestions that listeners might not notice further improvement of *ea_dist*. It also suggests that careful attention should be paid to whether further improvements of *ea_dist* actually hurt the user experience by cutting off context that users need to fully recognise and appreciate certain music events.

2.2.8. CONCLUSION AND OUTLOOK

This chapter has demonstrated the utility of timed comments as a source of labels to train models to detect socially significant music events. Through experiments, we show how timed comments can be utilised as training labels independently as well as in combination with expert labels. The important conclusions of our chapter are summarised here:

- Timed comments, on their own, can potentially be used as training labels to detect socially significant events. A model trained on timed comments alone performs better than a random baseline in terms of f-scores. In applications like non-linear access, where the listener wants to jump through different events in the music track without listening to it in its entirety, timed comments can already get you to around 20 seconds before the event.

- Adding expert labels improves the performance. Our experiments demonstrate that with a combination of 60% expert labels and 40% timed comments, we can potentially obtain a performance very close to the performance when we have 100% expert labels for training data. Again, this can be viewed from two perspectives. In terms of f-scores, we observe that a break event detector performs well at a combination of 60%-40%, while drop and build detectors perform well at a combination of 80%-20%. More importantly, in terms of the user based distance metrics, all the three event detectors perform well at a combination of 60%-40%.

- The performance of the event detection is not dependent on the source of data, as we obtain a good performance on an unseen test set, from YouTube, by using a model trained on SoundCloud data.

In this chapter, we have presented an extended case-study on using timed comments to detect events in the music signal. We looked at three specific events that are important in the EDM community: drop, build and a break. A cursory glance at other kinds of comments listeners mention on tracks reveals that these timed comments have a great potential for other tasks in MIR as well. For example, there are many comments that mention different structural parts of a track: “intro”, “bridge”, “instrument solo” and these can be used in generating more training data for music structure segmentation. Moreover, these timed comments can be used in music auto-tagging and emotion recognition at the temporal level and they could be a source of more natural training labels because of the vocabulary used by the listeners. For example, we find this comment “groovy tune” on a lot of music tracks, which could be a more useful tag while searching for music rather than genre/instrument/artist.

Our work is one of the first to utilise timed comments as training labels to develop an event detector. We hope that our results would encourage researchers to explore the

usefulness of timed comments for other media. We do not claim that the exact segment-based approach that we take here will transfer directly to videos. However, we would like to point out that our work has demonstrated that the impact of temporal noise can be overcome and that the contribution of timed comments to video event detection is worth investigating further.

REFERENCES

[1] K. Yadati, M. Larson, C. C. S. Liem, and A. Hanjalic, *Detecting socially significant music events using temporally noisy labels,* in *IEEE Transactions on Multimedia*, Vol. 20 (2018) pp. 2526–2540.

[2] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah, *High-level event recognition in unconstrained videos,* in *International Journal of Multimedia Information Retrieval*, Vol. 2 (2013) pp. 73–101.

[3] J. Schlüter and S. Böck, *Improved musical onset detection with convolutional neural networks,* in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (2014).

[4] J. Serrà, M. Müller, P. Grosche, and J. L. Arcos., *Unsupervised music structure annotation by time series structure features and segment similarity,* in *IEEE Transactions on Multimedia*, Vol. 16 (2014) pp. 1229–1240.

[5] H.-Y. Lo, J.-C. Wang, H.-M. Wang, and S.-D. Lin, *Cost-sensitive multi-label learning for audio tag annotation and retrieval,* in *IEEE Transactions on Multimedia*, Vol. 13 (2011) pp. 518–529.

[6] M. Müller, *Fundamentals of music processing: Audio, analysis, algorithms, applications,* (Springer International Publishing, Cham, 2015) pp. 167–236.

[7] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, *A survey of audio-based music classification and annotation,* in *IEEE Transactions on Multimedia*, Vol. 13 (2011) pp. 303–319.

[8] B. Frénay and M. Verleysen, *Classification in the Presence of Label Noise: A Survey,* in *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 25 (2014) pp. 845–869.

[9] B. Wu, E. Zhong, B. Tan, A. Horner, and Q. Yang, *Crowdsourced time-sync video tagging using temporal and personalized topic modeling,* in *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining* (2014).

[10] P. Xu and M. Larson, *Users tagging visual moments: Timed tags in social video,* in *Proceedings of the ACM Workshop on Crowdsourcing for Multimedia* (2014).

[11] R. Vliegendhart, B. Loni, M. Larson, and A. Hanjalic, *How do we deep-link?: Leveraging user-contributed time-links for non-linear video access,* in *Proceedings of the ACM International Conference on Multimedia* (2013) pp. 517–520.

[12] K. Yadati, M. Larson, C. C. S. Liem, and A. Hanjalic, *Detecting drops in electronic dance music: Content based approaches to a socially significant music event,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2014).

3

On the Automatic Identification of Music for Common Activities

In this chapter, we address the challenge of identifying music suitable to accompany typical daily activities. We first derive a list of common activities by analysing social media data. Then, an automatic approach is proposed to find music for these activities. Our approach is inspired by our experimentally acquired findings (a) that genre and instrument information, i.e., as appearing in the textual metadata, are not sufficient to distinguish music appropriate for different types of activities, and (b) that existing content-based approaches in the music information retrieval community do not overcome this insufficiency. The main contributions of our work are (a) our analysis of the properties of activity-related music that inspire our use of novel high-level features, e.g., drop-like events, and (b) our approach's novel method of extracting and combining low-level features, and, in particular, the joint optimisation of the time window for feature aggregation and the number of features to be used. The effectiveness of the approach method is demonstrated in a comprehensive experimental study including failure analysis.

Parts of this chapter have been published in the International Conference on Multimedia Retrieval, 2017 [1].

3.1. INTRODUCTION

In addition to "just" listening to music as a part of our leisure, we can also use music to facilitate our daily activities. For example, listening to appropriate music while studying can help us focus better [2]. While there exist online music services specialised in this direction (e.g., Focus at will and Brain FM), the mechanisms underlying their offerings are either automatic music generation or fully manual curation. Moreover, they only cover a limited scope of activities (e.g., focus only). A recent study on using "music as a technology" [3] to accomplish a goal highlights how little research has been done on developing music recommender systems for daily activities. This chapter delves into this area and investigates the main challenges in automating the process of identifying existing music for different daily activities.

The first challenge we face is deriving a list of daily activities for which music is sought. Instead of simply predefining this list, as is typically done in the existing work (e.g., [4]), we mine a social media sharing platform (YouTube) to derive a list of popular activities i.e., those activities that are found to have frequent mention when searching for activity-related music. Our mining approach is similar to that taken by recent work [5] on identifying common user-intent categories in online video search.

The second challenge we address in this chapter is to find an appropriate approach to automatically recognise activity-related music categories. We start by looking at the available textual metadata that we typically find associated with the music tracks that are posted on YouTube and promoted as being suitable for a particular activity. Specifically, the fact that the titles of these music tracks in many cases contain genre/instrument information, leads us to investigate the usability of this specific information first.

Since our findings indicate that relying on music genre or presence of particular instruments is not a reliable approach to link a music track to a particular activity, we proceed by investigating how to develop an approach that works well for the posed problem. For this purpose, we look into the existing content-based approaches in the field of music information retrieval (MIR) and encounter several issues that need to be addressed. The first issue is the temporal variation of musical content in a given track. We hypothesise that the time resolution for feature extraction in standard MIR tasks is not appropriate to capture the variations for activity-based music classification. Traditional MIR tasks typically extract features at a resolution of 10-100 ms and then aggregate them over the entire music track (e.g., for emotion detection [6]) or a segment sampled from the track (e.g., for genre recognition [6]). In this chapter, we propose to aggregate features over windows of different time resolution and identify the temporal resolution that can give optimal classification performance.

A related issue we address in this chapter is how to represent a music track in the feature space in order to enable effective activity-based classification. We take as our starting point a standard set of low-level features that can be extracted from the music signal. Additionally, we also consider some other sources of information that we, either intuitively or through exploratory experiments, found relevant for the task. Specifically, we consider different dimensions of affect (arousal, valence and tension [7]) and the presence of events like onsets and drops [8]. We encode this information in an additional set of high-level features. Finally, we design a classifier with which we investigate the possibility of identifying different activity-related music categories, and the usefulness

of different low- and high-level features for the task. Specifically, in parallel with optimising the time window for feature aggregation as explained above, we also optimise the number of low-level features to be used.

In summary, the main contribution of this chapter consists of the answers to the following research questions:

1. *RQ1: Which activity categories are popular?* We mine music on YouTube to derive common categories of activities. By analysing the textual metadata related to the activity-related music tracks, we identify the top-3 activity categories to focus on (Section 3.3).

2. *RQ2: Is genre or instrument information helpful in predicting an activity-related music category?* This research question is addressed in Section 3.4 by using the textual metadata of the music tracks, and in particular the presence of genre/instrument related keywords.

3. *RQ3: How to automatically identify music for a specific activity?* In Section 3.5, we investigate two aspects to dealing with this question, viz. the temporal resolution at which we should aggregate features and the types of features that would be helpful for the task.

The contributions listed above are presented after an analysis of the existing work on automatically associating music with daily activities, as well as different feature extraction strategies in Section 3.2. Experimental results assessing the performance of our proposed classification method and a failure analysis are presented and discussed in Section 3.6. Section 3.7 concludes the chapter.

3.2. RELATED WORK

In this section, we look at two different aspects dealt in this chapter: associating music with activities and different feature extraction strategies used in the MIR literature.

3.2.1. ASSOCIATING MUSIC WITH ACTIVITIES

Wang et al. proposed a method that associates music with specific activities [4]. The authors use a predefined list of activities: running, walking, sleeping, working, studying and shopping, for which they recommend music. Sensors on the mobile phone are used to infer whether the user is in the middle of one of these activities, and then suitable music is recommended based on an analysis of low-level features extracted from the signal. To train the recommender system, playlists for specific activities are collected from an online music sharing platform. Next, a subset of 1200 songs is picked from these playlists and manually labelled with one or more activities as tags. A classification problem is then set up where a model is trained for each activity based on the mean and standard deviation of low-level features extracted from a 512 sample frame extracted every 30 seconds of the song. Wang et al. use the following features for classification: Zero crossing rate, Centroid, Rolloff, Flux, Mel-Frequency Cepstral Coefficients (MFCC), Chroma, Spectral Flatness Measure (SFM) and Tempo. The trained model then predicts activity-based tags for new songs. Similar work is reported by Dias et al. [9], where the system "Improvise" is designed to associate music with daily activities mentioned above.

In our approach, we focus on the categories of activities derived from social media data and base the classification process on a novel feature extraction approach that, as we will explain and demonstrate experimentally, is more suitable for the task.

3.2.2. FEATURE EXTRACTION

Typical MIR tasks, like genre recognition, mood classification or instrument recognition, have been addressed frequently in the past [6]. Characteristic for these tasks is the way of extracting audio features, namely at the frame (time interval) level and with typically rather small frames, e.g., 10-100 ms for timbre features. In order to extract temporal features, like rhythm, a larger time window with a couple of seconds in length is used. Recently, the research community working on extracting emotion from music argued for using longer time windows and tracking emotions over "emotionally stable" segments [10], [11]. We take this discussion a step further by investigating segments of differing lengths while aggregating features for music-to-activity mapping. Additionally, we enrich the set of common audio features by new high-level features that we find especially useful for the task.

3.3. WHICH ACTIVITIES ARE POPULAR?

In this section, we address RQ1, i.e., we identify types of activities during which users commonly listen to music. Many daily activities are potential occasions for listening to music. A priori, examples include commuting, taking a shower, cooking, cleaning the house, studying or working out. However, compiling an exhaustive list of music-accompanied activities would require difficult-to-acquire behavioural information. For this reason, we focus on activities that are publicly mentioned, and can be assumed to be important to a substantial portion of general population. We turn to social media platforms as an information sources. Specifically, we analyse textual metadata on YouTube for common mentions of activities, which we take as providing indication of their popularity and wide-spread importance to users seeking music online.

When listeners are searching music for specific activities, we assume that search queries could take on various common forms, e.g., "Music for *", where the wildcard * could refer to a specific activity (e.g., studying, workout or jogging). Our metadata analysis is based on the observation that this query consists of a conjunction or a preposition connecting the other two words. In order to construct queries that would allow us to identify common activities, we looked at all possible prepositions[1] and conjunctions[2] that can follow the word "music". In this way, we arrived at five different word pairs: "music for", "music to ", "music when ", "music while" and "music during". By enclosing the word pairs in quotations, we created a query that could be matched with track metadata (i.e., title and description).

YouTube is a rich source of music, and offers a wealth of music intended for different activities, e.g., Study[3] or Workout[4]. In general, such music takes form of long tracks with duration typically exceeding 30 minutes. We use the queries just discussed to identify

[1] https://www.englishclub.com/grammar/prepositions-list.htm
[2] http://www.english-grammar-revolution.com/list-of-conjunctions.html
[3] https://www.youtube.com/user/StudyMusicProject
[4] https://www.youtube.com/user/WorkoutMusicService

Figure 3.1: Cloud from *Relax* track titles

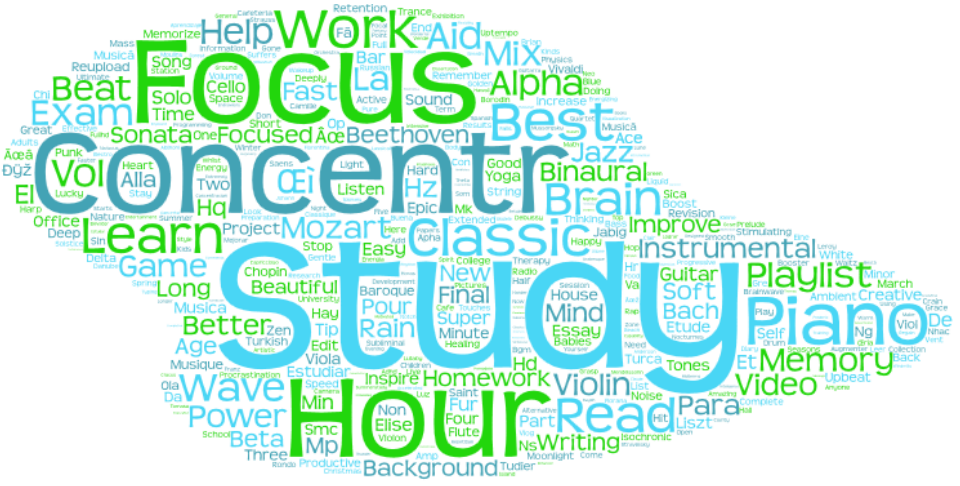


Figure 3.2: Cloud from *Study* track titles

Figure 3.3: Cloud from *Workout* track titles

Figure 3.4: Cloud from *Other* track titles

these tracks on YouTube. For each of the 5 queries, we follow these steps to collect the tracks and the corresponding metadata:

1. Using a web crawler, go through all the pages returned by YouTube for a given query and collect the unique identifiers of the videos as well as the titles.
2. Download the mp3 audio of the videos and the corresponding metadata, e.g., title, description and likes.
3. Remove duplicates in the search results and also remove the results that are not music tracks.

We accumulated a total of 2589 music tracks from YouTube and their textual metadata using our search queries. We used the titles of the collected music tracks to identify the most frequently occurring activities. We rely on the title of a track because it appeared to be the most informative about the music-type of the track. For example, the title "Workout Music - Best Workout Rock Music 2016 for GYM and Fitness" indicates that this track can be used while working out in the gym and it contains rock music released in the year 2016. We pre-process the titles by changing them into lower case, converting the -ing forms to their root words (e.g., studying is changed into study) and removing unicode characters, the standard English stop words, genre-related keywords, and numbers (e.g., years).

After this pre-processing, we counted the most frequently occurring terms in the titles, and arrived at the following top-3 activity-related keywords: "relax", "study" and "workout". Note that these keywords can be seen more as activity categories rather than single activities. Examples of single activities, e.g., for study music, include keywords like "work" and "office". Similarly, we find keywords like "run" and "exercise" in the titles of workout music. Our response to RQ1, is the top-3 activity categories, which we will focus on in the remainder of this chapter. To provide an impression of these categories, we provide a list of associated keywords:

- *Relax*: relax, calm, soothe, peaceful, chill, meditation, stress relief, sleep
- *Study*: study, focus, concentration, office, work
- *Workout*: workout, training, exercise, gym, run

Our final dataset contains a total of 1272 (49%) *Relax* tracks, 567 (22%) *Study* tracks and 450 (17%) *Workout* tracks. The remaining 300 (12%) tracks were found not to belong to any of the above three categories of activity-related music, despite the presence of the relevant keywords. Although this set of tracks is not used as a classification target, we keep it as *Others* and use it for analysis later in this chapter. In order to check for bias towards a particular Internet source, we also inspected the names of the channels from which the tracks were collected. We observed that the *Relax*, *Study* and *Workout* tracks were collected from 12, 10 and 9 different channels respectively, which gives a reasonable diversity of sources.

As a supplement to the keyword information above, Figures 3.1 - 3.3 show word clouds, which visualise the term clusters corresponding to our activity categories, which

were generated using the titles of the tracks, as described above. Common stop words, numbers, urls have been removed and stemming has been applied. The word clouds allow us to observe the difference in terms that characterise each of the three main activity categories we found. The word cloud for relaxation music contains keywords like "relax", "calm", "sleep", "heal", "meditate", "calm", "zen", "relief" and "lullaby". Similarly, "workout", "training", "gym", "fit" and "running" are the most frequently used keywords in the titles of workout music (Figure 3.3). Additionally, we observe the word cloud for the tracks not belonging to any of the above three categories labelled as "Others" in Figure 3.4. Observing Figure 3.4, we can say that there is a lot of music for babies, playing games, pets (Dogs) etc.

3.4. IS GENRE OR INSTRUMENT INFORMATION ENOUGH?

In this section, we address RQ2 and investigate whether genre or instrument information is helpful for predicting music for the top-3 activity categories identified in the previous section. For this investigation we do not develop nor implement any existing genre or instrument detection method. Rather we rely on the textual metadata carrying information about the music genre or instruments present in the titles and descriptions of the music tracks we crawled from YouTube. Our hypothesis is that if the link between the genre- or instrument-related textual metadata and a particular activity category is unambiguous, then it is meaningful to focus on the development, implementation and optimisation of the corresponding content-based methods and algorithms as the means to solve the activity-related music classification problem.

The next question to answer is whether the specific genre- and instrument-related terms found in the term clusters are also distinctive per activity category. In order to answer this question, we pick the genre- and instrument-related keywords from the titles of tracks in each of the four term clusters and arrange them in Table 3.1. Please note that there is no particular order in which the genres or instruments are laid out in the table. Since the dataset contains both electronic and acoustic music, we list the instruments found only in acoustic music. Observing the table, we can say that investigating genre or instrument is not enough to associate music to activities. We can see that genres like classical music, electronic music and ambient music are present in three of the four clusters. In particular, house music is present in all the four clusters, thus also in the *Others* cluster consisting of the tracks for a wide range of activities other than the three targeted in this chapter. Similarly, piano, guitar and violin are present in three of the four clusters.

We now take a look at example music tracks for genres and instruments that are common between different activity categories. First, we compare genres in *Relax* and *Study* categories and pick one of the common genres present in both the categories: *Trance*. Listening to the examples of *Trance* music in *Relax* and *Study*, one can immediately identify a difference in texture where a *Study* music example has a slightly higher density than a *Relax* music example. Another difference is the presence of drop-like events [8] in a *Study* music example and a complete absence of such events in the *Relax* example. We refer to drop-like events as those that follow similar acoustic and rhythm patterns as drop events that are typically associated with electronic dance music (EDM). Drop events generally occur as combinations of two different events, viz. *drop* and *build*, defined as follows:

1. Drop: A point in the EDM track, where the full bassline is re-introduced and generally follows a recognisable build section.

2. Build: A section in the EDM track, where the intensity continuously increases and generally climaxes towards a drop.

These events are associated with the increasing intensity of the music and reaching a climax before the beat returns. We investigate these events because they represent the building up of intensity and changing of rhythm, which could be important for music for study (to not let the listener zone out) or workout (to push the listener to intensify the workout). Finally, the *Study* music examples were completely devoid of vocals, unlike the *Relax* music examples that have vocals at certain points in the track. This analysis revealed that even within one genre (*Trance*), some musical properties could make one track of that genre suitable for the *Relax* and the other track for the *Study* category.

Music for *Relax* is bound to be very different from the music for *Workout* as they are activities at the extremes of physical exertion with *Relax* requiring least physical activity and *Workout* requiring high level of physical activity. Even though we expect the two categories to be related to completely different music, they still have some genres in common, like *dubstep* and *hip hop*. Listening to an example for both the *Relax* and *Workout* category in the *dubstep* genre, one can clearly observe a difference in terms of tempo and texture. As expected, the tempo is higher and the music is more dense for the *Workout* example as compared to the *Relax* example. Listening to the *Workout* example, one can observe the prominence of the bassline, which is at times "naked" without melodic layers. In contrast, in the *Relax* example the bass is much less prominent. As indicated earlier, *Relax* music does not contain drops but there are many drops in the *Workout* example. Our conclusion here is therefore the same as above. Though the tracks belong to the same genre, there are significant variations that make it challenging to rely on genre information alone to distinguish between music for *Relax* and *Workout*.

Finally, we look at two examples from the same genre (*progressive house*), but from different activity categories: *Study* and *Workout*, and notice the presence of vocals in the *Workout* track. Another key difference is the presence of many drop-like events in the *Workout* track and limited number of such events in the *Study* track.

Based on the analysis reported above, we can conclude that genre- and instrument-related information alone is not sufficient to predict suitability of a music track for an activity. Observing the individual examples, we see the main reason for this is the local properties of a music track, i.e., localised variations in low-level features. In the subsequent section, we therefore propose a method which segments the tracks into windows of different time resolutions in order to investigate how to optimally capture these local variations for the posed classification task. Furthermore, the insights presented above motivate our decision to consider the presence/absence of drop-like events as one of the features in the design of activity-related music classification framework, as stated in Section 3.1.

Type of music	Genres	Instruments
Relax	classical, binaural, jazz, ambient, house, trance, dubstep, chillstep, hip hop, trap, rock, country, folk	piano, guitar, flute, saxophone, violin, drums
Study	classical, binaural, jazz, ambient, trance, Drum & bass, electronic, deep/electro/progressive house	piano, violin, guitar, viola
Workout	electro, dubstep, progressive house, rock, rap, EDM, hip hop, electronic, techno	piano, drum
Others	jazz, classical, binaural, house, ambient, rock, folk, dubstep, EDM, electronic, electro, trance	piano, saxophone, guitar, drums

Table 3.1: Genres and musical instruments present in each activity category.

3.4.1. ADDITIONAL EXPERIMENTS ON UTILITY OF EXISTING METADATA

Previously, we qualitatively analysed how existing metadata like genre and instrument is not enough for activity based music classification. Table 3.1 summarised our insights where we find that the same genre/instrument labels occur across activity categories. In this sub-section, we provide a more quantitative evidence to support our hypothesis. We take advantage of the recent advances in deep learning to learn a feature descriptor that encodes genre and instrument information from a music track. We fine-tune the model trained by Choi et al. for music auto-tagging [12] using another auto-tagging dataset MagnaTagaTune [13]. We take the top-50 most popular tags from the MagnaTagATune dataset that includes various genres and instrument related tags[5]. After fine-tuning the model, we do a forward pass using our data and combine the features of the all layers of the model to obtain a 256-dimensional feature vector [14]. From each of our tracks, we extract a 29 second snippet from the middle of track. We train a k-NN classifier, and predict the activity category of the music tracks that are part of the test set (explained in further experiments). We obtain the following f-scores for each of the activity category: 0.44 (Relax), 0.49 (Study), 0.58 (Workout), and 0.50 (Average). Comparing with the results that are reported in subsequent experiments (Table 3.3), we observe that the f-scores are lower with this set of features. This provides additional results that support our hypothesis that genre/instrument information is not sufficient for activity based music classification.

3.5. HOW TO IDENTIFY MUSIC FOR ACTIVITY CATEGORIES

In this section, we describe our approach to developing an automatic classification method for activity-to-music mapping. Since the information on genre or instruments is not helpful in detecting music for a given activity, classification based on other and more relevant information needs to be developed.

We start off by noting that recent advances in deep learning, such as [15], may enable unsupervised extraction of relevant features. However, we would like the features that we identify as contributing towards identifying music for activities to be explainable, and we would also like to carry out an assessment of the temporal resolution that is appropriate for feature extraction. Explainability of deep learning pipelines for music currently still is in a pioneering phase [16]. For these reasons, we choose to investigate features and models that are already well understood and reflect different musical characteristics. More specifically, we take as input a basic set of low- and mid-level (rhythm and tonality related) features known from the MIR field. These features and their corresponding dimensionality (in parenthesis) are listed in Table 3.2.

[5] https://github.com/emarkou/Audio-auto-tagging

Type of features	Features
Low-level features	Avg. loudness (1), dynamic complexity (1), pitch salience (2), spectral centroid (2), spectral complexity (2), spectral decrease (2), spectral energy (2), spectral entropy (2), spectral flux (2), spectral kurtosis (2), spectral rms (2), spectral rolloff (2), spectral skewness (2), spectral spread (2), zero crossing rate (2), MFCC (13), Image moments (162) [8]
Rhythm features	Number of beats (1), tempo (1), danceability (1), onset rate (1), statistical spectrum descriptor (168) [17], rhythm histogram (60) [17]
Tonal features	chromagram (12), key strength (1), pitch class profile (pcp) (36)
High-level features	Number of events (1) [8], Affect (3)

Table 3.2: Low-level/mid-level/high-level features we use for distinguishing between music for different activities.

In addition to the problem of understanding what features are suitable for the task in the first place, the main open issue related to how the features are extracted is the selection of the time window t to optimally aggregate the feature values in order to capture the above mentioned informative local signal variations in the best possible way. In order to discover the best value for t, we devise an algorithm that we run on our training data set and that uses repeated random sub-sampling validation [18] to test different values of t. In the same algorithm, we also embed the search for the best value of another parameter d, which stands for the number of most discriminative features used for classification. We use a simple k-nearest neighbour (k-NN) classifier and repeat the algorithm to identify the combination of d and t that gives the best classification performance. The proposed algorithm is defined as follows:

1. Consider the range of $t = 0.5, 1, 5, 10, 15, 20, 25, 30$ seconds
2. For each value of t, follow these steps:
   (a) Extract features for each segment and combine them into a single feature vector.
   (b) Randomly divide the training data into X_{train} (90%) and X_{val} (10%).
   (c) Select a value of d from the set 10, 11, 12 ... to 50 features.
   (d) Use X_{train} for feature selection and pick the top-d most discriminating features. Before feature selection, we normalise each feature.
   (e) Use X_{train} with selected features to build a training model.
   (f) Use this model to predict labels in X_{val}.
   (g) Aggregate the segment-level labels using a majority vote to obtain a single label for a track and then compute the f-score.
   (h) Repeat steps 2 (d) – (g) for the whole range of d.
   (i) Repeat the whole process ten times for different X_{train} and X_{val} each time to obtain average validation performance.
3. Choose the t with the best average validation performance.

In this chapter, we aim to understand the phenomena underlying the activity-related music classification and *not* to optimise the classification itself. This is the reason for which we chose a simple and standard k-NN classifier, which has minimal number of parameters to be tuned. Regarding the range we considered for t, we also investigated the window sizes beyond 30 seconds (up to 60 sec) and found that the performance does not improve. For feature selection, we use a method that deploys mutual information and that is available in the feature selection toolbox [19]. Once we identify the best values of d and t, we evaluate the performance on the test set to predict the links between the music tracks and the activity categories.

The other features we introduce are based on intuition, informed by the analysis in Section 3.4. Here, we consider three affect dimensions, namely arousal, valence and tension, and assess their impact to activity-related music classification experimentally,

using conventional scores [7]. We do the same with the feature encoding the number of drop-like events [8] found in a music track: while the consideration of this feature initially was also based on intuition, the potential of this feature has been strengthened by the analysis reported in the previous section. The affect scores are extracted over non-overlapping segments of duration t seconds (result of the algorithm described above) and for the events feature, we count the number of drop-like events in the entire music track.

Window size(sec.)	F-score(relax)	F-score(study)	F-score(workout)	Overall F-score	No. of features
0.5	0.41	0.23	0.56	0.4	46
1	0.43	0.24	0.55	0.41	45
5	0.41	0.28	0.57	0.41	48
10	0.59	0.58	0.79	0.65	40
15	0.59	0.63	0.86	0.69	38
20	0.58	0.61	0.83	0.67	26
25	**0.60**	**0.69**	**0.89**	**0.73**	**25**
30	0.60	0.59	0.89	0.69	35

Table 3.3: F-scores for different classes and number of discriminative features across different window sizes.

Feature	F-score(relax)	F-score(study)	F-score(workout)	Overall F-score
Events (E)	0.79	0.64	0.72	0.72
Affect (A)	0.63	0.51	0.73	0.62
A + E	0.65	0.51	0.71	0.62

Table 3.4: F-scores using high-level features at a windows size of 25 sec.

3.6. EXPERIMENTAL EVALUATION

In this section, we evaluate our proposed method and also compare its performance to a number of existing methods and approaches that we found related to the problem addressed in this chapter. Additionally, we report which features are the most discriminative for our classification task. Finally, we summarise the insights we gained from a failure analysis, based on which we propose topics for future research in this direction.

3.6.1. EXPERIMENTAL DESIGN AND RESULTS

For the experiment, we have a training set, containing 300 relax tracks, 250 study tracks and 250 workout tracks. For the test set, we use 50 each of relax, study and workout tracks. We focus on tracks that are specific for a single activity category, as reflected in our labels. Music suitable for multiple categories is an interesting topic for future work, but we do not look at it here.

For finding t and d according to the algorithm described in the previous section, we set k = 10 for the k-NN classifier after evaluating different values of k on a smaller development set (not included in the train and test set). For extracting low-level, tonal features and most of the rhythm related features, we use the Essentia framework [20]. For statistical spectrum descriptors and rhythm histogram, we use the source code provided by Lidy et al. [17]. For extracting the low-level features, we use a non-overlapping frame size of 100 ms. Regarding the high-level features, we use the method proposed in [8] to detect the drop-like events in a given music track. We rely on the dataset released by Yadati et al. [21] to train models and predict the presence of events in our dataset. Finally, for computing the affect scores, we use the MIRtoolbox [7] that gives us a 3-dimensional feature vector with one score per dimension.

Table 3.3 shows the f-scores per activity category obtained while executing the algorithm for optimising the values of t and d, as introduced in the previous section. We note again that the classification here is performed using the low- and mid-level features only. It can be observed that the best classification performance was obtained at a window size of 25 seconds. Examining the f-scores obtained at this window size, we can say that the simple classifier performs reasonably well in distinguishing between music for the three different categories.

As indicated in the last column, for the window size of 25 seconds, the best number d of discriminative features to use is 25. Here, we list the features (and their dimensionality) that are found to be most discriminative in this case: tempo (1), dynamic complexity (1), danceability (1), onset rate (1), spectral centroid (1), spectral flux (1), image moments (6), PCP (4), rhythm pattern (4), rhythm histogram (3) and MFCC (2). This is a mix of rhythm features, low-level features and tonal features, with a majority of them being rhythm-related and with PCP being the only representative of the tonal features. A key observation here is that most of the selected features (tempo, danceability, rhythm pattern etc.) generally need longer time segments to be computed. We therefore believe that the flexibility we allowed in the selection of the time window t was critical for pushing these features forward as being most informative for classification and therefore also critical for getting the most out of the signal and achieving the best possible classification performance.

We also performed the classification based on the high-level features, first separately

and then integrated with low- and mid-level features. We computed the affect features in the time interval corresponding to the optimal value of t, namely 25 seconds. Experiments using other window sizes showed, however, that this parameter is not critical, resulting in relatively constant classification performance. Computation of the event feature is not dependent on the time window as this is solely the number of drop-like events found in a music track. The classification results for different features and their combinations are presented in Table 3.4. We observe that the high-level features generally perform worse than low- and mid-level features. An interesting exception is the result obtained for the events feature and *Relax* category. The detector of the drop-like events that we adopted from [8] is namely known for its high precision and low recall. This is beneficial for the *Relax* music tracks having no drops and less beneficial for the tracks from other two categories where drop-like events are present, but because of the detector deployed, not well detectable. This result shows the potential of this feature to improve the overall classification performance upon the one obtained by using low- and mid-level features, however, under the condition that the detector of drop-like events performs well. We discuss this further in the next section.

So far, we looked at the performance of our method in isolation. We now compare our method with existing methods which classify music tracks into activities. Specifically, we compare it with the method proposed by Wang et al. [4] and also with two other methods that we devised as being representative of common approaches deployed in standard MIR classification tasks. The four methods entering a comparative analysis are:

1. *Our method*: As a representative of our proposed approach we choose the method variant deploying low- and mid-level features with the best performance in Table 3.3, namely for the time window of 25 seconds and 25 features.

2. *Full track*: We aggregate the low-level features, extracted from 50ms frames, over the entire track by computing the mean and variance. We then combine these features with other rhythm and tonal features extracted from the whole track. We perform feature selection and select the most discriminative features (51 in this case). Using a k-nearest neighbour classifier, we predict the labels of the music tracks in the test set and compute the f-scores for the three categories. Such a method is inspired from the field of static emotion recognition [6], which aggregates the features over the entire music track in order to give an affect score for a track.

3. *One segment*: We select one 25-second segment from each track in the training data and extract the features as before. We then perform a feature selection and obtain the most discriminative features (49 in this case). We divide each music track in the test set into 25-second segments and select these 49 most discriminative features. Using a k-nearest neighbour classifier, we predict the labels of each 25-second segment in the test set and use a majority vote to get a single label for a track. We then compute the f-scores for the three categories. This method is inspired by existing MIR approaches, especially genre recognition [6], where a short segment taken from the track is used for feature extraction and classification.

4. *Wang et al.*: Wang et al. extract features from a 512-sample frame every 30 seconds and compute the mean and variance of these features over the entire track. Then, they use an adaboost classifier to predict the labels of music tracks in the test set. We follow this procedure on our dataset and compute the f-scores for the individual categories.

Figure 3.5 summarises the results of all four methods for the three target activity categories. We can clearly see that our method outperforms other methods. We further observe that the *Full track* method that aggregates features over the entire track performs better than the *One segment* method and the *Wang et al.* method at least in two categories: *Relax* and *Study*. From Table 3.3 and Figure 3.5, we can conclude that aggregating over a longer window size helps in classifying the music track into one of the three activity categories and, based on the experiments on our dataset, the best window size is 25 seconds.

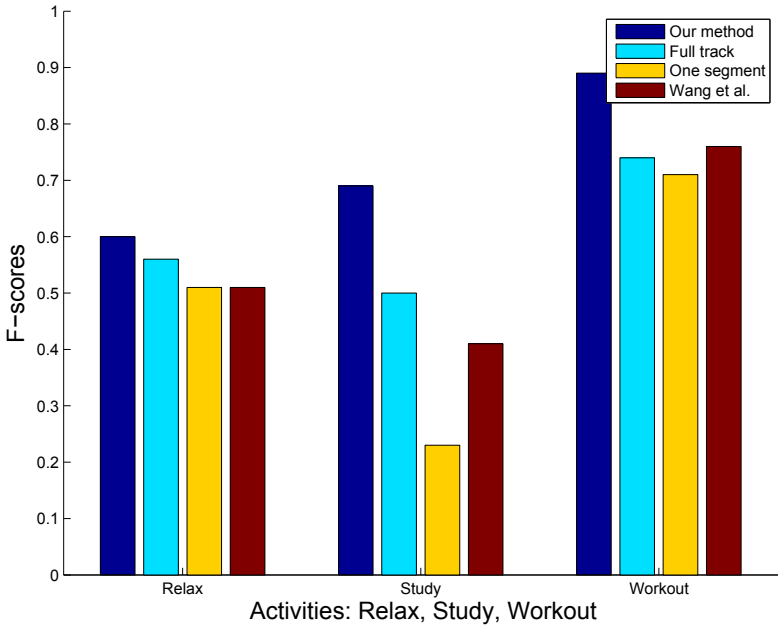


Figure 3.5: Comparison of performance for three activity categories across different methods.

3.6.2. FAILURE ANALYSIS AND OUTLOOK

Through different experiments, we have shown that we can distinguish between music for different activities and that our method performs better than the related existing methods. However, even the best results are not perfect. In this section, we focus at these imperfections by analysing the failure cases where music tracks belonging to a certain activity category are assigned a wrong label.

Figure 3.6 illustrates the confusion matrix for the predicted labels on the test set and the numbers in the boxes indicate the number of correctly/wrongly classified samples.

The first observation we make is that there is considerable confusion between the *Relax* and *Study* categories. Here, we take a look at individual examples and try to find patterns that lead to this confusion between *Relax* and *Study*. Using a majority vote to aggregate the labels seems to be the reason behind some of the misclassifications. One of the tracks in the relax category, which had 72 segments in total, is misclassified as a *Study* track because 33 segments are classified as *Relax*, 34 segments are classified as *Study* and the remaining segments are classified as *Workout*. A majority vote clearly finds that the track is *Study*, but the competition between *Relax* and *Study* categories was close. We also found many other examples where the difference between the number of segments classified as *Relax* and those classified as *Study* is low. There was even an example in the *Study* category that contained equal number of segments classified as *Relax* and *Study*, but the max operator chose the category of the track as *Relax*. In order to investigate this phenomenon further, we measured the mean and standard deviation of the difference between the number of segments in the top two categories for each example. For correctly classified examples, the mean is 48.6 while the standard deviation is 16.2. For incorrectly classified examples, the mean is 26 while the standard deviation is 6.1. We clearly see lower values for incorrectly classified examples, indicating that there is a closer competition between categories for incorrectly classified examples. Clearly, deploying majority vote has drawbacks as it does not reflect how strong the majority is. This calls for investigating different aggregation strategies that can combine the labels of individual segments into a single label for the track in a more robust fashion. Alternatively, we would also like to explore whether we could choose the segments in a smart way (analogous to feature selection) that are most discriminative for an activity category, which removes the need for an aggregation step. Another possible direction could be to consider the labels for the segments as a sequence instead of considering them as a bag of segments (current method). The temporal ordering could provide additional information that could reduce the misclassfication rate.

We initially hypothesised that drop-like events are completely absent in *Relax* music, while they are present in the other two categories. This is confirmed by the classification results reported in Table 3.4, in particular by a high f-score for *Relax* music when events feature is used. As explained in the previous section, this effect is additionally emphasised due to a strong bias of the event detector used towards high precision. However, there is more to it. Some of the *Study* music tracks also do not contain drop-like events and this resulted in a confusion between *Relax* and *Study* categories. Furthermore, there are similar numbers of drop-like events in some *Study* music and *Workout* music tracks, which results in lower f-scores for these two categories. Another reason for failure is that there are more subtle drop-like events in *Study* music while *Workout* music has more pronounced events and the drop detector missed detecting some of the subtle events.

Mapping between low-level features and affect is a difficult proposition and we have used an off-the-shelf toolbox to compute the affect scores for the music tracks. Observing the results reported in Table 3.4, the affect based classifier performs reasonably well, but there is a scope for improvement. We could look at different strategies to compute affect scores in the future and investigate its impact on the classification performance.

An aspect of activity-based music that needs further attention is the presence of distractors, which are musical characteristics that might distract the user from his/her ac-

tivity. For instance, one of the observations in Section 3.4 was that *Study* music did not have any vocals while the other two types of music could contain vocals. In the future, one could investigate to which extent the presence/absence of vocals is informative as a feature for this classification task. In general, one could search for additional sources of information, e.g., user comments, that can help identify the distractors for different activities. Here are some examples of user comments that can be used to identify if the track is really useful for an activity:

- Comment on a relax music track: "There is a jarring piano sound in the middle!"
- Comment on a study music track: "This track contains vocals and distracting while working"

The biggest challenge we see when relying on user comments is to spot the comment with the relevant information among plenty of (largely noisy) comments posted by the users.

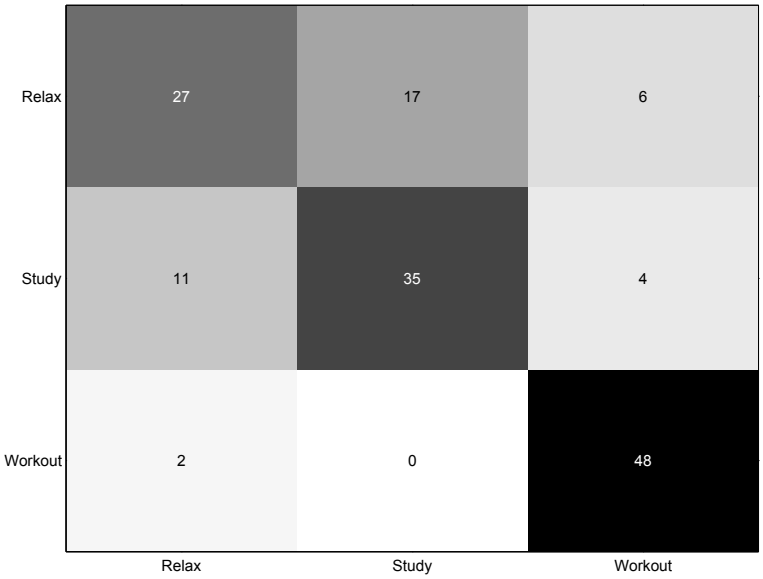


Figure 3.6: Confusion matrix for the best performing case of $t = 25$ seconds

3.7. CONCLUSION AND OUTLOOK

In this chapter, we have addressed the challenge of identifying appropriate music for common daily activities. In this way, we made a critical step towards developing a music recommender system that takes into consideration both the aspects of what music is and what it can do for a listener. We have focused on the three activity categories that we found to be be common via a study of textual metadata on YouTube: *Relax*, *Study* and *Workout*.

One of our key findings is that the presence of standard musical metadata, like genre, instrument, do not seem to be enough for addressing the problem of activity-based music classification. We demonstrated this through multiple experiments: using genre/instrument information to classify and using the feature extraction strategy inspired by these tasks. However, we have used a standard techniques: k-NN classifier, standard features and feature selection in our multi-class classification problem as our focus was not on optimising for performance but validating our idea of activity based music classification. Future work could involve experimenting with more advanced classifiers/techniques. Another important finding is that this task requires more timeline information (25 seconds) for feature extraction from an audio track, i.e., the window size must be longer than what is currently conventional in the MIR literature.

Based on these findings we have developed a method that identifies the time resolution at which the low-level features should be aggregated and also the best number of discriminative features to be used. Using the features extracted at the identified temporal resolution, our classifier could successfully distinguish between music for the three different activity categories and also outperform existing methods.

This chapter opens interesting perspectives for future work. From the musical content perspective, we plan to investigate additional information to improve the identification of music for activities. Here, we have taken a bag-of-segments approach. Moving forward, however, we anticipated that incorporation of the temporal order of the segments could, as mentioned above, provide further insight. Further, also as mentioned above, users post comments on YouTube for different music tracks. Some of these touch on the suitability of a music track is for a specific activity. These comments are a promising source of information. Additionally, high-level features, e.g., presence/absence of vocals, could also improve classification.

Our work here is based on the insight that there are general characteristics of music which have a similar reception across a broad population. In pursuit of these general characteristics, we focus on information about music tracks provided by uploaders. We adopt an assumption used recently in work on video uploader intent [22]: the fact that uploaders are publishing on a public platform, accessible to millions of users, makes it likely that they are taking the musical reception of the general population into account. The fact that we focus on here on broad consensus on which music is appropriate for which purposes, should not preclude future study of the role played by individual preferences in users' choices of music for different activities. Individual preferences should also be understood as preferences of groups of users who pattern together, such as introverts and extroverts, as studied by [23]. Moving forward, understanding where universal music preferences fall short of being useful will allow us to gain further insight into the performance of the classifier. Specifically, we would like to investigate the relatively large confusion between the music for the *Relax* and *Study* categories from a user's perspective. Such a user study would allow us to determine whether the classifier should be further improved, or whether the category labels must be refined to make it possible to cater for finer-grained preferences within the population.

REFERENCES

[1] K. Yadati, C. C. Liem, M. Larson, and A. Hanjalic, *On the automatic identification of music for common activities,* in *Proccedings of the ACM International Conference on Multimedia Retrieval* (2017).

[2] S. Bottiroli, A. Rosi, R. Russo, T. Vecchi, and E. Cavallini, *The cognitive effects of listening to background music on older adults: processing speed improves with upbeat music, while memory seems to benefit from both upbeat and downbeat music,* in *Frontiers in Aging Neuroscience*, Vol. 6 (2014) p. 284.

[3] A. Demetriou, M. Larson, and C. C. S. Liem, *Go with the flow: When listeners use music as technology,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2016).

[4] X. Wang, D. S. Rosenblum, and Y. Wang, *Context-aware mobile music recommendation for daily activities,* in *Proceedings of the ACM International Conference on Multimedia* (2012).

[5] A. Hanjalic, C. Kofler, and M. Larson, *Intent and its discontents: The user at the wheel of the online video search engine,* in *Proceedings of the ACM International Conference on Multimedia* (2012).

[6] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, *A survey of audio-based music classification and annotation,* in *IEEE Transactions on Multimedia*, Vol. 13 (2011) pp. 303–319.

[7] O. Lartillot and P. Toiviainen, *MIR in Matlab (II): A Toolbox for Musical Feature Extraction from Audio,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2007).

[8] K. Yadati, M. Larson, C. C. S. Liem, and A. Hanjalic, *Detecting drops in electronic dance music: Content based approaches to a socially significant music event,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2014).

[9] R. Dias, M. J. Fonseca, and R. Cunha, *A user-centered music recommendation approach for daily activities.* in *Proceedings of the ACM Workshop on Content based Recommender Systems* (2014).

[10] A. Aljanaki, F. Wiering, and R. C. Veltkamp, *Emotion based segmentation of musical audio,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2015).

[11] L. Lu, D. Liu, and H.-J. Zhang, *Automatic mood detection and tracking of music audio signals,* in *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 14 (2006) pp. 5–18.

[12] K. Choi, G. Fazekas, and M. B. Sandler, *Automatic tagging using deep convolutional neural networks,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2016).

[13] E. Law, K. West, M. Mandel, M. Bay, and J. S. Downie, *Evaluation of algorithms using games : The case of music tagging,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2009).

[14] K. Choi, G. Fazekas, M. B. Sandler, and K. Cho, *Transfer learning for music classification and regression tasks,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2017).

[15] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, *Unsupervised feature learning for audio classification using convolutional deep belief networks,* in *Proceedings of the International Conference on Neural Information Processing Systems* (2009).

[16] J. Pons, T. Lidy, and X. Serra, *Experimenting with musically motivated convolutional neural networks,* in *Proceedings of the International Conference on Content based Multimedia Indexing* (2016).

[17] T. Lidy and A. Rauber, *Evaluation of feature extractors and psycho-acoustic transformations for music genre classification,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2005).

[18] W. Dubitzky, *Fundamentals of Data Mining in Genomics and Proteomics* (Springer-Verlag, Berlin, Heidelberg, 2009).

[19] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján, *Conditional likelihood maximisation: A unifying framework for information theoretic feature selection,* in *Journal of Machine Learning Research*, Vol. 13 (2012) pp. 27–66.

[20] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. R. Zapata, and X. Serra, *Essentia: an audio analysis library for music information retrieval,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2013).

[21] M. Larson, K. Yadati, M. Soleymani, and P. S. N. Chandrasekaran Ayyanathan, *Mediaeval 2014 crowdsourcing task: Crowdsorting multimedia comments,* (2014).

[22] C. Kofler, S. Bhattacharya, M. Larson, T. Chen, A. Hanjalic, and S. F. Chang, *Uploader intent for online video: Typology, inference, and applications,* in *IEEE Transactions on Multimedia*, Vol. 17 (2015) pp. 1200–1212.

[23] G. Russell, *Preferred stimulation levels in introverts and extroverts: Effects on arousal and performance.* in *Journal of Personality and Social Psychology*, Vol. 46 (1984) pp. 1303–1312.

4

AUTOMATIC IDENTIFICATION OF DERAIL MOMENTS IN FOCUS MUSIC

Listening to music while engaging in another activity is a common phenomenon. Many times, however, the listener finds that a music track starts out fine for a task, but suddenly something happens in the music that completely distracts the listener from the task. We call such a moment in a music track a "derail moment". We investigate the different aspects of derail moments and propose a method to detect them. This work represents the first step towards a music player that could automatically detect an upcoming derail moment and smoothly blends into the next track before it arrives. Our work is motivated by evidence that derail moments are not purely personal experiences, but that they have elements of universality. Two sources provide evidence of the existence of such elements. First, a survey of neuro-, cognitive-, and social- psychology literature reveals potential causes of derailment. Second, an exploratory user study on Amazon Mechanical Turk (AMT) among people who listen to music while working reveals commonalities in the self-reported experience of distracting moments in music. Building on this evidence, the chapter then proposes an automatic method to detect derail moments. We create a dataset by collecting music tracks from real users who work on AMT while listening. We draw on their experience to train and evaluate a model than can automatically detect a derail moment. Through experiments, we demonstrate the effectiveness of our method. Analysis and discussion of our results sheds light on which aspects of derail moment detection remain particularly challenging.

This chapter is in preparation for submission to the Transacations of International Society for Music Information Retrieval [1].

4.1. INTRODUCTION

Listening to music is a daily activity for many, and a large amount of music is readily available via today's online platforms. Although listening to music can be considered an activity in itself, it can also support another activity that the listener is simultaneously engaged in. For example, you might be studying for an exam, working in your office, or commuting to your workplace, and you could potentially listen to music during any of these activities. This chapter opens a new perspective on what makes music appropriate for focusing on work. Specifically, we introduce the concept of a *derail moment*, which is a point at which a listener loses focus due to something that happens in the music signal. We propose an automatic approach that makes it possible to detect these moments in a music tracks, with the goal of helping listeners to avoid them.

The motivation for our work comes from two directions. First, we are motivated by research evidence that has revealed that people listen to music in the course of activities other than deliberate music listening. This observation was made by [2], who found that when people are listening to music by themselves, in about 20% of the cases they reported it was helping them listener to concentrate or think. Recently, [3] have argued that more music retrieval research should focus on situations in which listeners use music as a tool to achieve certain psychological effects. This work points out that people effectively use music "as technology" while performing another activity in order to help themselves achieve mental states in which their attention is focused on their task. We build on this work by diving more deeply into specific moments in music that cause listeners to lose focus on their task.

Second, we are motivated by research evidence that it is important for listeners to be able to control the music that they listen to. In particular, we point to a research study by [4], which concludes that users want to have the possibility to manipulate aspects of their experience, including how distracting tracks are. Although listeners prefer more control, they would also prefer, in general, to have a minimal amount of interaction with their music players. Further evidence of the importance of control in playlists and streams is covered by [3].

The experience of a "derail moment" is described with the following scenario. Many times, it happens that you start listening to a music track thinking that it is suitable for the task you are doing at that moment, and that the track will not distract you. The track starts out fine, but suddenly something occurs in the music that distracts you from the task you are doing, and you stop what you are doing to skip the track manually in order to continue working. The "derail" moment is the moment that the music causes your concentration to break. A track that contains a derail moment is not useful for listening while working because you cannot simply play the track, ignore it and focus on your task. This chapter investigates the nature of derail moments, their characteristics, and the automatic detection of a derail moments in music tracks.

The context of our work is a growing number of music services that provide music for focus (e.g., Focus@Will, Brain.fm) and relaxation (e.g., Brain.fm). Additionally, there are music playlists/mixes available on popular music streaming services like YouTube, Spotify, Google Play music. This context supports our position that music retrieval should help users to find focus music. However, to our knowledge, until now, research on focus music has addressed the track level, and has not looked at the effects of disruptive

moments with tracks. As mentioned above, control of music choice is known to be important for listeners. We believe that not only should listeners have a large choice of music tracks available, but they should also be able to choose to avoid certain moments in these tracks.

As a step towards building such tools, we envision a smart music player that can continuously play music without distracting listeners from their tasks. Listeners can create their playlists with music tracks of their choice, and the smart music player automatically detects the positions of any derail moments present in the music tracks. When the derail moment is approaching in the current track that is playing, the player smoothly blends into the next track so that the listener is not distracted by it. Such a smart music player would build on research in automatically creating playlists while considering different factors, such as the order of the tracks and different blending strategies, e.g., [5]. The primary focus of this chapter is to develop a method that can automatically detect derail moments in a music track, and, as such, we leave the development of the specific method for creating playlists to future work.

We start with a minimal assumption: a listener experiences a derail moment due to something that has shifted or is shifting within the music. On the basis of this assumption, we conceptualise a derail moment as the start of a new music event potentially associated with a number of different properties. The event may start gradually or may start suddenly. The duration of the event may be long or may be short. After the event, the music may go back to be suitable for focus, or it may evolve in ways not suitable for focus. The assumption that a derail moment is a start of an event allows us to consider derail moment detection as a type of event detection. We do not, however, make assumptions about which of the properties holds for any given derail moment.

Motivated by this reasoning, this chapter proposes an approach to derail moment detection that is based on approaches known to be effective for music event detection. Specifically, we adopt the insight of [6] that effective event detection can be carried out with a two step approach, which first segments the music track and then identifies segment boundaries as events. We adopt a segment-based approach to automatically detect a derail moment in a given music track, and explore features that are effective for derail moment detection.

Although we base our approach on event detection, it is important to clarify the ways in which derail moment detection is conceptually a different problem from standard event detection. Typically, an music event is conceptualised as occurring in the audio signal, and people creating or listening to music have a high level of agreement on the occurrence of events. A derail moment, on the other hand, is conceptualised as occurring in the listener's mind. It is triggered by a shift in the audio signal, but there is not necessarily a high level of agreement among creators and listeners on the occurrence of derail moments. Within a given music track there may be two moments that are identical from the point of view of the audio signal. Under an event detection viewpoint, a successful detector would need to detect both moments as accurately as possible. Under a derail moment detector viewpoint, a successful detector must only detect the first moment. The second moment is not actually a derail moment, since the user has already lost focus and switched to another track. Also, detection should be accurate. However, a derail moment detector is successful if it predicts the moment somewhat earlier in the

music stream than it occurs, but fails completely if it predicts it later than it occurs. This sort of asymmetry does not arise with conventional music event detection.

The main contribution of this chapter is to explicitly identify the concept of a *derail moment*, point out its importance for music listeners, and to demonstrate a productive way forward in creating automatic detectors of derail moments that would be useful for music recommendation systems. To our knowledge, we are the first to work studying derail moments, and to create a detector that can automatically predict their occurrence. Because we are entering new territory, there are a number special challenges that we must face in studying derail moments and in creating a detector that can automatically predict their occurrences. Here, we summarise these challenges, and briefly describe how we tackle them:

1. *Not all users will experience the same derail moments.* We expect that what causes listeners to lose focus will not be identical for all listeners. We address this challenge, by investigating whether there are any aspects of derail moments that are potentially shared among users. We find evidence, through a literature survey and a user study, that there are elements of universality to a derail moment. This evidence motivates us to continue to study derail moments, with the idea that a detector that focuses on this element of universality could be beneficial for a wide range of listeners.

2. *Not all derail moments are punctual, some are gradual.* Although we do expect that some derail moments will be triggered by sudden events in the signal (such as a sudden clash of cymbals), other derail moments will triggered by gradually evolving events (such as a crescendo). We address this challenge by asking the listeners who create the ground truth for our experiments to provide us with the first moment in the track at which they think that the music becomes distracting. In this way, we are able to circumvent variation in the exact moment at which the user decides to stop listening to a music track. We also ask our listeners to tell us whether they experienced the derail moment as a very clear point in the track or as the start in the change of a track. We then analyse the performance of our derail moment detector taking these differences into account.

3. *No existing dataset supports the study of derail moments.* Studying derail moments requires input for a large number of people on how they experience music. Since different tasks require different types of focus, creating a dataset for studying derail moments requires controlling for the type of tasks that people are doing while listening to music. We address this challenge, my turning to Amazon Mechanical Turk (AMT) in which we are able to request input from a large group of people who regularly listen to music while carrying out a specific type of data set.

It is our hope, that this work will inspire researchers in the future to work on the problem of derail moments. In order to make our work reproducible and support future research, we release the text of our user study questionnaires, the dataset that we created, and the code of our classifier on the Open Science Framework to promote further research (https://osf.io/n3zmp/).

This chapter is organised around a set of research questions, which our research strives to answer:

1. Are derail moments purely listener-specific or do they have an element of universality?

2. Can we automatically detect the position of a derail moment based on features extracted from the music signal?

3. Given a track, can we detect whether there is a derail moment in it?

The chapter is structured as follows. After having introduced our problem and the challenges that it presents in this section, we proceed to survey literature pertinent to the task of derail moment detection in Section 4.2. We then move on in Section 4.3 to answering the first research question on elements of universality of derail moments by carrying out a survey of existing neuro-, cognitive-, and social- psychology literature and also a user study among music listeners on AMT. We explain the procedure to collect our dataset and analyse it in Section 4.4. Next, we present our method to detect derail moments in a music track in Section 4.5. We describe the experimental setup in Section 4.6 and discuss our results in Section 4.7. We then provide further insights into the results in Section 4.8, before proceeding to conclude the chapter in Section 4.9.

4.2. RELATED WORK

In this section, we look at different aspects related to the challenge of detecting a derail moment in a given music track. Since we consider derail moments to be music events, we start off by reviewing the field of audio event detection. We then review literature in the field of music for activities.

4.2.1. AUDIO EVENT DETECTION

Research related to audio event detection can broadly be divided into three categories: environmental sound recognition, music event detection, and music structure analysis. Environmental sounds that can be detected in a given audio stream include, for example, bell ringing, applause, footsteps, or rain. Various features and learning methods have been proposed to model the typically non-stationary characteristics of the environmental sounds [7].

Event detection in music has typically focused on detecting low-level events, such as onsets [8]. Music onset detection is a well-studied problem in music information retrieval (MIR), and it is offered as a task in the MIREX benchmark evaluation every year. A slightly related task at MIREX is music auto-tagging [9], which assigns descriptive tags to short segments of music. These tags generally fall into three categories: musical instruments (e.g., guitar and drums), musical genres (e.g., pop and electronic) and mood-based tags (e.g., serene and intense).

Recent work on music event detection has extended from low-level events to high-level events [6], where the authors propose methods to detect and localise high-level events in a given music track. They use a case study on events in Electronic Dance Music (EDM): Drop, Break, and Build, to illustrate that these high-level events can be detected using temporally noisy labels. The presence of these high-level events may help us in identifying whether there is a derail moment in a given music track. An important part

of the high-level event detection pipeline is music structure segmentation [10], which divides the music track into its structural segments.

In music structure analysis [10], the objective is to divide a given piece of music into its various sections and later group them based on their acoustic similarity. Structural elements are a very important characteristic to the identity of a piece of music. For example, in popular music tracks, these structural elements could be the intro, the chorus, and the verse sections. Different aspects of musical expression have been deployed for analysing the musical structure, such as homogeneity (e.g., in instrumentation), repeating patterns (e.g., in rhythm or melody) and novelty (e.g., through a change in tempo or tonality).

In this chapter, we consider a derail moment to be an event that occurs in a music track and triggers a reaction from the listener. It is a higher level event than a drop, as a drop is only one kind of event that can distract the listener. Hence, we need to follow an approach that is different from low-level event detection (onsets). Unlike auto-tagging, we are dealing with the stream of a music track to detect a derail moment. We use the output of music structure segmentation to get an indication of the probable position of a derail moment, as we hypothesise that a derail moment could be closer to a structural segment boundary.

4.2.2. MUSIC FOR ACTIVITIES

Recently, interest has arisen in the research community for the challenge of automatically identifying music for various daily activities. Wang et al. proposed a method that associates music with specific activities [11]. The authors use a predefined list of activities: running, walking, sleeping, working, studying and shopping, for which they recommend music. Sensors on the mobile phone are used to infer whether the user is in the middle of one of these activities, and then suitable music is recommended based on an analysis of low-level features extracted from the signal. To train the recommender system, playlists for specific activities are collected from an online music sharing platform. Next, a subset of 1200 tracks is picked from these playlists and manually labelled with one or more activities as tags. A classification problem is then set up where a model is trained for each activity based on the mean and standard deviation of low-level features extracted from a 512 sample frame extracted every 30 seconds of the track. Wang et al. use the following features for classification: Zero-crossing rate, Centroid, Rolloff, Flux, Mel-Frequency Cepstral Coefficients (MFCC), Chroma, Spectral Flatness Measure (SFM) and Tempo. The trained model then predicts activity-based tags for new tracks. Dias et al. reports similar work. [12], where the system "Improvise" is designed to associate music with daily activities mentioned above.

Recently, Yadati et al. [13] provided an alternate perspective on automatic identification of music for common activities. Instead of defining a list of activities in advance, they proposed a data-driven approach that identifies the most common activity categories for which music is sought. They rely on a social media sharing platform (YouTube) to identify the following common activity categories: Relax, Study and Workout. The authors investigate a variety of low-level, high-level features in addition to the metadata provided by the media uploaders to categorise a music track into one of the three activity categories. One of the main findings of the authors was that the metadata like genre,

instrument, or artist does not provide a useful classification. However, presence of musical events had a significant role in classifying music into the three activity categories. The events considered by Yadati et al. [13] were drop-like events from Electronic Dance Music (EDM) that attracted a listener's attention towards the music. This attraction towards music is a distinguishing factor in identifying music for focus and workout. In focus music, you want to limit this attraction and in workout music you want to maximise this attraction. Another significant finding is that the window size needed to extract features should be longer (25 seconds) than what is state-of-the-art practice (100ms) in music information retrieval research.

Our investigation in this chapter is different from the approaches as mentioned above. The existing approaches investigate what kind of music is suitable for specific activities, while we study what musical events can distract a user from performing his/her task. Many times, it could be the case that a majority of a music track may be suitable for focusing on a specific task, but it has a particular segment or a moment in the track that can completely distract the listener. In this chapter, we are trying to find such events in a given music track and warn the user before he/she chooses a specific track for working on a task.

4.3. ON THE ELEMENTS OF UNIVERSALITY OF DERAIL MOMENTS

In this section, we address our first research question: Are derail moments purely listener-specific or do they have an element of universality? Upon first consideration, it might be tempting to assume that what causes a derailment of concentration is dependent only on the specific listener. In the introduction we have already mentioned the assumption that derail moments are subjective to be a factor that has held previous researchers back from looking at this topic. We were initially triggered to revisit this assumption by the observation that distraction has its roots in basic brain functions. If distraction has a dependence on the nature of our brains, it is to be expected that at least some elements contributing to what listeners identify as a derail moment must be universal, i.e., shared among all listeners. This section first surveys the neuro-, cognitive-, and social- psychology literature and develops an argument for rejecting the assumption that derail moments are completely listener-dependent. This argument motivates our guiding assumption that there is enough listener-independence to derail moments to make it worthwhile studying them from the perspective of their elements of universality. The survey also provides motivation for choosing the features to extract from the music signal in order to build machine learning models that can automatically detect derail moments. Next, the section turns to analyse an exploratory survey that we carried out among listeners who use music for a specific type of task. Here, again we find evidence that

4.3.1. NEUROSCIENCE PERSPECTIVE

Our survey on the neuro-, cognitive-, and social- psychology literature starts from the assumption that derailment is a form of distraction. For this reason, we focus on features of music tracks that might result in

1. attention being drawn to the music, and

2. an increase in physiological arousal.

To guide our hypotheses, we turn to the BRECVEMA model of musical emotions [14], which reviews literature concerning the different ways music has been shown to elicit emotion in a listener. We include emotions in our discussion, as lower order emotions are theorised to be automatic responses to stimuli that were relevant to survival in the ancestral environment. We draw on three of the lower order components of the BRECVEMA model: brain stem reflexes (B), rhythmic entrainment (R), and emotional contagion, specifically as regards the presence or absence of vocals in a musical piece (C). [14] hypothesise that all three components occur involuntarily and automatically, and are therefore free from cultural impact. These components, therefore, may guide the automatic extraction of features that universally affect the individuals listening to them.

MUSICAL EVENTS

We tie brain stem reflexes to the presence of events within a music track. It is hypothesised that loud, dissonant, low frequency, sudden onset or changing sound signal to the brain stem that something urgent may have occurred, and that this event may demand immediate attention and reaction [14]. In the ancestral environment, for example, one can imagine such sounds occurring as the result of the movement of something significant, or because of a dominant force (e.g., falling boulders, or the roaring of a large predator). While the effect from music is likely far less pronounced, pieces that contain "events" that meet these criteria may result in a) the attention of the listener being diverted to the music and b) an increase in arousal of the listener. We would thus hypothesise that the larger the number of such incidents in the piece, and the more frequent the presence of noisy and dissonant sounds, the better suited the piece might be for activities where arousal is desired, and the attention being paid to the activity is minimal (e.g., alleviating boredom during a morning commute, housework, exercise involving repetitive actions like running). However, such noisy music may be distracting when the person is attempting to focus on task. Moments where dramatic changes in the audio signal occur, such as "drops" in electronic music [6], might be expected to draw attention to the music, arousing the listener in the process, and causing distraction.

PULSE CLARITY

Music that has a clear pulse may be more likely to lead to entrainment, e.g. electronic dance music which is characterised by a conspicuous and steady bass drum. Entrainment is typically defined as the synchronisation of two systems caused by their interaction: musical pieces can be considered oscillating systems, as can various physiological processes in the human body (e.g., neuronal oscillations, cardiac activity, respiration etc. [15] posit that entrainment between these various physiological processes and a musical signal occurs at different levels. Perceptual entrainment, or the cognitive process underlying the recognition of the meter in a musical piece (i.e. recognising the "beat") can be expected to be greater when the pulse is clear (i.e., when the notes occur on downbeats, and when the last downbeat is accented) and when the meter is easy to recognise and predict (i.e., a standard, even, simple time signature as opposed to an unfamiliar, odd, complex time signature). Physiological entrainment is typically considered to be the tendency of respiratory and cardiac activity to adapt to the perceived pulse of the

music, although it may involve other biological systems that are less convenient to monitor. As a result, musical pieces with a salient rhythm result in a higher likelihood that perceptual and physiological entrainment will occur to the perceived tempo of the piece. Given that the brain stem is likely to react to a sudden onset, low-frequency sound [14], and sounds that appear consistently on the downbeat are likely to lead to perceptual and physiological entrainment [15], we focus on the frequency and salience of bass drum notes in the piece. We further expect faster tempos to result in increased arousal, and lower speeds to result in decreased arousal. However, when these beats are constant we would not expect them to distract a listener, even though the listener may use such music to keep themselves aroused. On the other hand, if the drum beats were to suddenly change or disappear entirely, this may result in a derail moment.

VOCALS

It has been shown that the brain reacts differently to music when there are vocals vs. when there are no vocals, and when the vocals contain lyrics vs. vocalisations without words [16]. Similarly, physiological arousal has been shown to be higher for music with vocals vs. music without vocals (e.g. [17]). Specifically, [17] showed that the presence of vocals correlated with pupil dilation, which in turn is associated with the locus coeruleus (LC). The LC is a region of the brain that has been shown to affect both attention and arousal (see [18]). As such, we would expect the presence of vocals in a musical piece to draw more attention and to be more arousing when compared to compositions that do not have vocals. Therefore when working on a task, it may be the case that music with vocals may be more distracting and likely to derail than music without vocals. Furthermore, the BRECVEMA model [14] suggests that the presence of vocals may also result in emotional contagion. Thus, if the vocals convey a relaxed emotional state it is suggested the listener will also feel more relaxed, e.g. as a child would when listening to the mother sing a lullaby. On the other hand, vocals that convey a high level of arousal, such as screaming, may also result in the listener being aroused. As such, music that contains vocals whose perceived emotions vary, or where the vocals appear suddenly, may be more likely to derail the listener.

Furthermore, it is possible that the presence of music with lyrics will interfere with concentration on tasks that require cognitive engagement, mathematics, or the processing of language, such as reading or writing (e.g., [19]). As such, the presence of lyrics may be distracting when the task at hand involves similar kinds of processing. Therefore, a song with lyrics may result in a difficulty being able to process what is being read, which may derail the listener.

In summary, experiments in neuro-, cognitive-, and social- psychology literature suggest the existence of basic brain behaviours that draw a listener's attention towards the music. Since these behaviours can be expected to be common for all listeners, their existence supports our position that it is worthwhile to assume that derail moments have a universal component that makes it worthwhile to investigate them in a user-independent manner. Additionally, we use these findings to compile a list of musical features that we can extract from the signal for building our machine learning models: presence/absence of drop-like events, pulse clarity, tempo, and the features representing the characteristics of vocals. We will use these features while we discuss our second research question in the subsequent sections.

4.3.2. USER STUDY AMONG MUSIC LISTENERS

In our quest to find whether there exists an element of universality to the phenomenon of derail moments in music tracks, we now turn to hear directly from people who use music as a tool to support their work. We carry out a user survey on Amazon Mechanical Turk (AMT), where we can find a large number of users performing similar tasks and listening to music while doing so.

Based on literature [20], [21] and our investigation on AMT, we found that text transcription HITs are ubiquitous and popular among AMT participants. In these HITs, the participants are given an image of a receipt or an invoice and are asked to transcribe the contents into a web form. These HITs are well-defined and this is important because literature provides evidence that listening to music while performing well-defined tasks makes them more enjoyable [22]. While designing our HIT to collect responses from participants, we wanted to control for the variability in what people consider appropriate focus music by looking at a single type of task, which is also an important type of task. We then designed our HIT asking participants to provide us insight into what kind of music they would listen to while doing a text transcription task. Another reason for choosing this particular task is because it is a well-defined task and has no ambiguity about it. We asked the user study participants the following questions to understand what kind of music they would listen to, and what type of music they would avoid listening to, while working on text transcription tasks.

1. Please describe the type of music that you think people generally listen to while doing transcription HITs. We are interested in how the music sounds (in other words, try to describe what you hear in the music, rather than giving artists or genres).

2. People sometimes have the experience that they are listening to a new music track, and it starts out being fine for working. But then something happens in the music making the track not good for working anymore. In your experience, please tell us what ruins a music track for work.

3. Finally, go to YouTube and provide us with a link to a video of a music track that you would like to have in a transcription HIT playlist.

4. Please explain why you chose this example.

In addition to these four main questions, we also asked the participants for an appropriate name for this moment in the music track where you get distracted from the task. We gave the following options to the participants: Track crash, Deal breaker, Derail moment, Track fail, others. A vast majority of participants (80%) responded that they would call the moment a derail moment and hence we decided to call it a “derail” moment.

The overall goal of the HIT was to understand the main reasons why a moment in a track ruins it for work. We performed a card-sorting process on the answers we got for the second question on distracting elements in a music track. We printed the responses on rectangular strips of paper and timed the card-sorting process. For card-sorting, we manually read through the responses from the participants and categorise them into different clusters. We start with the first response as the first cluster, and as we read through

Category	Examples
Signal-oriented	Songs that go from fast to slow or slow to fast can also disrupt rhythm when working. A bass drop would ruin the track. A sudden shift from the rest of the piece would also throw me off.
Artist-oriented	When the lyrics become too dark or violent. When the lyrics become too dark or violent
Listener-oriented	Any music that is going to make you think about things other than work at that moment will ruin music for work by doing to opposite of what you seek in helping you focus. Music that invokes too much emotion or thought ruins the work focus and motivations. Well, I was listening to Tom Petty quite a bit in my playlist as I love "Face in the crowd" and "Into the Great Wide Open". But, after he died this week, I find it hard to listen to those tracks because I begin thinking about his death and it makes me a little upset.

Table 4.1: Examples for the different categories of participant responses on AMT.

the answers, we either create a new cluster or merge the response into an existing cluster. Three of the authors participated in this process, and it took us approximately 3 hours to converge on a stable clustering of responses.

Our card-sorting process resulted in three major categories of responses to distracting elements in a music track: signal oriented characteristics, artist-oriented characteristics, and personal preferences. There were different sub-categories under each of the three major categories. For the signal oriented characteristics, responses included comments on loudness, sudden changes in tempo or volume or style, bass drops, repetitiveness, complexity. For the artist-oriented characteristics, people commented on the high pitch of the singer, screaming, foul language, wordiness, and in general presence of vocals. While discussing their personal preferences, participants talked about liking, familiarity, emotion, memory (good/bad). You can see some representative example responses in Table 4.1. For a full list of the responses and their corresponding categories, read the following document: participant responses. We also generated a visual representation of the various comments from the user study participants on AMT in the form of a word cloud (Figure 4.1[1]).

Observing the three main categories, we can say that personal preferences or subjectivity in a derail moment is only a part of the phenomenon. There are other more universal characteristics which define a derail moment, and these are identifiable by a general audience.

From neuro-, cognitive-, and social- psychology literature, we found suggestions that the following characteristics should be very important in selecting music that is not distracting: tempo, vocals, events, pulse clarity. If we carefully observe the responses from participants, we see confirmation of the suggested characteristics. Additionally, other characteristics emerge viz., loudness, dynamic complexity. Multiple participants mention these characteristics while responding to the question of distracting elements of

[1] https://wordart.com/

Figure 4.1: Word cloud of all the responses from the participants on AMT.

music while working on a specific kind of task. From the responses, we could filter out personal preferences and concentrate on the signal oriented characteristics. Our experimental findings (categorisation) on AMT and our neuro-, cognitive-, and social- psychology literature survey provides support for the expectation that there is an element of universality to the phenomenon of a derail moment. Once the methods to automatically identify these universal elements are developed, future research could progress to the addition of personal preferences

4.4. DATASET AND ANALYSIS

As indicated earlier, we turn to AMT to collect data on music tracks and the corresponding derail moments. We created a new HIT on AMT asking the participants to give us examples of music tracks with derail moments. The HIT gathered the following information from the participants:

1. YouTube URL for a music track
2. Timestamp of the derail moment in hh:mm:ss format
3. Explanation on what exactly distracted the participant,
4. If the derail moment was a point where the participant lost focus or if the music is starting to get distracting at a particular moment.

We conducted pilot HITs for the following regions: USA, India, and Europe, as these are the regions where most of the participants on AMT come from. Compared to India

and Europe, we got responses of higher quality from the USA as well as faster response time, and hence we chose to go with the USA for our primary data collection HIT. We published 1000 HITs for the USA region to collect our dataset.

We collected 1000 music tracks and their corresponding derail moments. Apart from the timestamp and reason for it being labelled a derail moment, we also asked the participants to tell us if the music immediately became so unbearable that he/she had to change the track, or if the music had slowly started to become distracting. This helps us in keeping the events happening in the music signal and the reaction event from the listener separate. By keeping these events separate, we obtain a stable set of training labels which can be used for training robust machine learning models for detecting a derail moment.

Another aspect we want to investigate in our dataset is the distribution of derail moments across the timeline of a music track. We divide each music track into 10 parts and label the part containing the derail moment. By counting the number of derail moments in each of the ten parts, we observe that a derail moment occurs during the initial 30% (std: ±4) of the track on an average. There are approximately 200 tracks in our dataset that have a derail moment in the first 10% of the timeline, while there are around 20 tracks that have a derail moment towards the final 10% of the track.

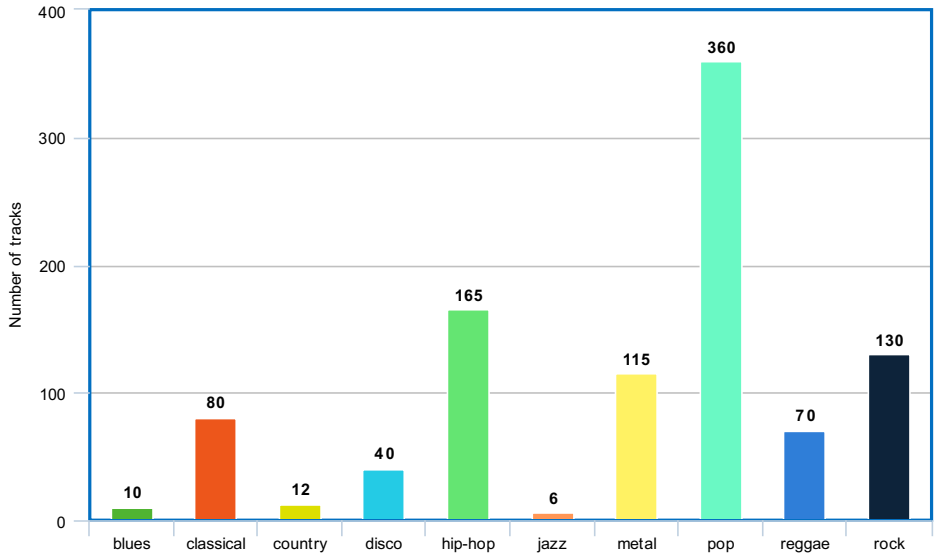
To gain a clearer picture of the variety of the music that we collected, we performed genre recognition on the tracks using a state-of-the-art deep learning model [23]. We used the model and the weights provided by the authors to recognise the genres of our music tracks. Figure 4.2 illustrates the genre distribution of our dataset. Although we can see that many tracks belong to the pop genre, there is variety in terms of other genres like hip-hop, metal, classical, rock etc. This suggests that derail moments are not genre specific.

In general, we hypothesise that the derail moments are closer to structural segment boundaries. To prove this, we perform structure segmentation on our development set and measure the distance between the segment boundary and the derail moment in seconds. We then compute the average distance, and we observe that derail moment is approximately 4 seconds away from the segment boundary. This experiment demonstrates that the derail moments are indeed close to the segment boundaries. We use this finding to identify the appropriate segmentation algorithm in the next sub-section.

4.4.1. SEGMENTATION

Structure segmentation plays a significant role in our method, as indicated in Section 4.1. For structure segmentation, we use the Music Structure Analysis Framework (MASF) [24]. The framework provides implementations of the various structure segmentation algorithms, and these algorithms are divided into two categories: boundary algorithms and labelling algorithms. Boundary algorithms identify the segment boundaries in the track, while labelling algorithms also label the segments according to their similarity with each other. For a list of the available algorithms, please visit the tutorial page of MSAF. Additionally, different features can be explored: Mel-Frequency Cepstral Coefficients (MFCC), Constant Q-Transform (CQT), Pitch Class Profiles (PCP), and Tempogram. These algorithms take the music track as an input and produce a sequence of segments (boundary algorithms) with similar segments being clustered together (labelling algorithms). Table

Figure 4.2: Genre distribution of the tracks in our dataset.



Start time	End time	Label
0	0.3	0
0.3	11.3	1
11.3	28.7	2
28.7	44.1	3
44.1	53.2	2
53.2	64.4	1

Table 4.2: Snapshot of the output of segmentation algorithm on a music track.

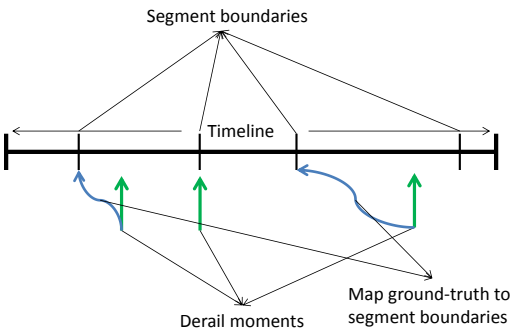
4.2 illustrates the segmentation algorithm on a music track. Each row represents a segment, giving the start time, end time and the label of the segment. Observing the table, we can say that segments having the same label are similar to each other (e.g., third and fifth segments are labelled as 2).

We explore all possible combinations of boundary algorithms, labelling algorithms, and features on the development set. For all possible combinations, we measure the average distance between the segment boundaries and the derail moments. Finally, we select the combination that results in the least average distance. For our dataset, we found that the following combinations give the least average distance: Structural features [25], Convolutional Non-Negative Matrix Factorization [26], and MFCC. The output of the segmentation algorithm is a list of segment boundaries and the corresponding labels. The labels correspond to the similarity of the segments i.e., segments having the same label are considered to be similar to each other.

One of the segmentation algorithms in MSAF is the novelty detection algorithm proposed by Foote [27]. From our experiments with MSAF, we found that the segments provided by the novelty detection algorithm do not give a good approximation of the derail moments. This provides evidence that derail moment detection requires analysis beyond novelty detection.

After segmentation, we also need to assign training labels to these segments so that we can pass them onto the machine learning model. Figure 4.3 illustrates how we map our ground-truth labels, which are provided in the form of timestamps by the participants of our user study, onto the segment boundaries. Observing the figure, we can say that we map the ground-truth timestamps of derail moments to a segment boundary that occurs prior to the derail moment. The reason for doing this is that we do not want to predict a derail moment after it has actually passed. To make sure that we predict a derail moment before the actual derail moment occurs, we follow this mapping procedure.

Figure 4.3: Mapping the ground-truth to segment boundaries.



4.5. AUTOMATIC DETECTION OF DERAIL MOMENTS

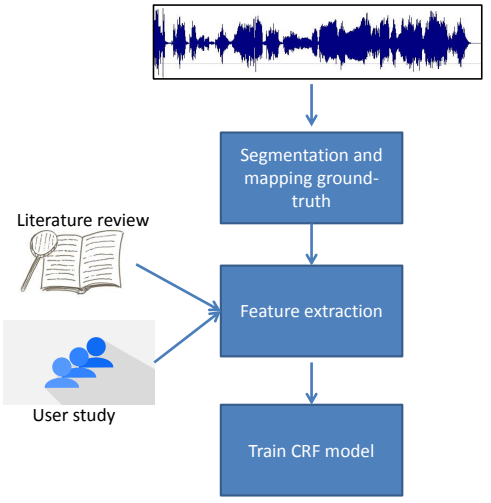
The previous section has motivated the existence of an element of universality to the derail moment, which motivates us to develop automatic methods to detect these in music tracks, and we also identified a list of features that can be used to build machine learning models to detect derail moments. We also took a look at our dataset collected from AMT and performed some analysis on it to judge the variety of tracks and how we map the ground-truth to structural segment boundaries. In this section, we will focus on developing a method that, given a music track, can identify a derail moment in it. Using the dataset collected from AMT participants, we report the performance of our method in identifying derail moments.

4.5.1. APPROACH

We consider a derail moment to be an event that happens in the music track and we design our pipeline accordingly. The method has several steps, performed in the following sequence: structure segmentation, feature extraction, training, testing, and evaluation.

Figure 4.4 illustrates the steps involved in training a model that can predict derail moments in a music track. We will briefly describe each of the steps mentioned above.

Figure 4.4: Block diagram illustrating the proposed approach with the following steps: Structure segmentation and mapping, feature extraction inspired by the neuro-, cognitive-, and social- psychology literature survey and user study, and training a model.



The first step in our method to detect a derail moment is structure segmentation. The intuition behind a segment-and-then-classify approach is that we hypothesise an underlying event structure in a music track and that some of these events distract the listener. Music structure segmentation provides an indication of the structure within a music track and we hypothesise that the derail moment would occur close a segment boundary. We make the assumption that segment boundaries represent either the beginning of the event, or the point at which it becomes unbearable. Following up on this observation, we segment the music track into its structural segments using a structure segmentation algorithm. In addition to segmentation, we also map the ground-truth timestamps to these segment boundaries, as illustrated in Figure 4.3.

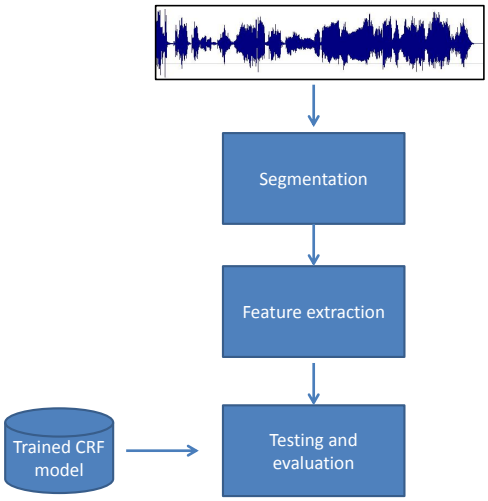
From each of the segments (from the structure segmentation algorithm), we perform feature extraction that will further be used for training a machine learning model. For feature extraction, we rely on the findings from our neuro-, cognitive-, and social-psychology literature survey and the AMT user survey. Some of the features were motivated from the neuro-, cognitive-, and social- psychology literature: pulse clarity, presence/absence of events (drops or drop-like events), vocals. Some other features were motivated from our findings on the AMT survey: tempo, dynamic complexity, loudness. Apart from the two features on detecting events and vocals, the other features can be obtained from standard feature extraction libraries like essentia [2].

To assess the presence/absence of drop-like events, we use the model trained on music data from SoundCloud [6]. Figure 4.6.1 illustrates the different steps in the test-

[2] http://essentia.upf.edu/documentation/introduction.html

ing/evaluation procedure. We provide the individual segments as input to this model and acquire a binary label indicating the presence/absence of a drop-like event in the segment. Similarly, for the other feature on vocals, we use the feature proposed by Lehner et al. [28].

Figure 4.5: Block diagram illustrating the evaluation procedure.



We model the task as a sequence-to-sequence labelling problem, where we algorithmically assign a label to each element of a sequence. The final output is a timepoint at which a derail moment has occurred. This is chosen to be the first derail moment in the sequence. We model the music track as a sequence of segments, where all segments until the derail moment are labelled as positive examples. For labelling the segments after the derail moment, we follow a slightly different strategy.

It is possible that the music after a derail moment can become appropriate for focus again. We take this aspect into consideration while labelling the segments for training and we describe the training procedure here. Considering this aspect, we utilise the output of structure segmentation algorithm (explained in Section 4.4.1) to assign training labels to the segments after the first occurrence of the derail moment. If any of the segments after the derail moment have a label that is the same as that of a segment before the derail moment, we consider it as a positive training example. Otherwise, we consider the segment as a negative training example.

We train a conditional random field (CRF) [29] for our task. Exploratory experiments demonstrated that classifying individual segments is not effective and hence we need to exploit the temporal information. Once we have a trained CRF model, we predict the labels of the sequence of segments in new tracks (test set).

4.6. EXPERIMENTAL SETUP

Until now, we discussed the approach for detecting a derail moment in a given music track. In this sub-section, we discuss the experimental setup in the form of evaluation strategy, vocal detection, and the baseline algorithm.

4.6.1. EVALUATION

We evaluate our proposed method in two different scenarios, addressing two research questions. For the second research question on detecting a derail moment in a given track, we need to evaluate how accurate the predicted derail moments are and how close they are to the actual derail moments (provided by AMT participants). We use two different evaluation metrics: accuracy and distance. For measuring accuracy, we consider the correct labelling as the one explained in the training procedure (Section 4.5.1). Accuracy measures the percentage of segment boundaries that are labelled correctly. Distance measures the gap between the predicted derail moment and the actual derail moment. The distance measure would be useful in judging the utility of our method in building a user-oriented application where the player automatically detects the derail moment and moves to the next track before the actual derail moments kicks in.

For the third research question on predicting whether there is a derail moment in a given music track (RQ3), we perform a qualitative experiment by feeding tracks without any derail moments ("clean" tracks) to the model and the above metrics would not be informative. To evaluate how well we answer RQ3, we manually check the number of tracks where our model predict or does not predict a derail moment. Additionally, we also check the position of the predicted derail moment to see if it is at the beginning or end of the track.

4.6.2. VOCALS DETECTION

In our user study, people report that vocals are something they notice when listening to music and there is indirect evidence that the vocals are processes uniquely in the brain [30]. We investigate the role vocals play in determining whether a moment in the music is distracting. Examining research on detecting vocals in music tracks in the MIR community, we observe that there are two different directions: Singing voice detection [31] and Source separation [32]. Singing voice detection tackles the problem of identifying where there are vocals in a music track, while source separation algorithms aim to separate the vocal and the instrumental parts of the track. In this chapter, we concentrate on singing voice detection as we want to use it as a feature to identify the derail moments in a given music track. There have been multiple approaches proposed in the MIR community for singing voice detection and an excellent survey of these approaches is presented by You et al. [31].

In this chapter, we use the features proposed by Lehner et. al. [28]. Lehner et. al. propose a set of features for singing voice detection: *fluctogram*, *spectral flatness*, *spectral contraction*, *MFCC*, and *vocal variance*. The Fluctogram is basically an extension of a feature suggested by Sonnleitner et al. [33] for speech detection in mixed audio signals. The basic idea behind their feature is to detect sub-semitone fluctuations of partials by using the cross correlation. Spectral flatness provides a way to quantify how noise-like a sound is, as opposed to being tone-like. The other feature on spectral con-

traction measures how much of the energy in the spectrum resides in the centre. The Vocal variance comprises 5 values, computed on the first five MFCCs only. For each of these 5 first coefficients, we compute its variance over 11 successive frames centred on the current frame. The features are computed at the resolution of 200 ms and dimensionalities of the individual features are as follows: fluctogram (17), spectral flatness (17), spectral contraction (17), MFCC (30 + 30 delta), and vocal variance (5). More details are provided in [28] and we use the code provided by the authors to extract these features. On the 116-dimensional feature vector, we apply dimensionality reduction (Principal Component Analysis) to obtain a feature vector that has 4 dimensions. Hence we have a 5-dimensional feature vector for each segment for capturing characteristics related to vocals.

4.6.3. BASELINE

Our method focuses on detecting the first derail moment that occurs in a music track. AMT participants indicated that any sudden change would result in distraction, which informed our baseline strategy. We built our baseline strategy on these lines. We identify the segment boundaries using the best possible combination of segmentation, labelling algorithms, and features identified in Section 4.4.1. We ignore the first segment because it usually is a very short segment (< 1 second). From the remaining segments, we pick the first segment boundary, where the label of the segment changes (obtained from the labelling algorithm), as the derail moment.

We measure the accuracy and distance metrics as explained previously and compare it against our proposed method.

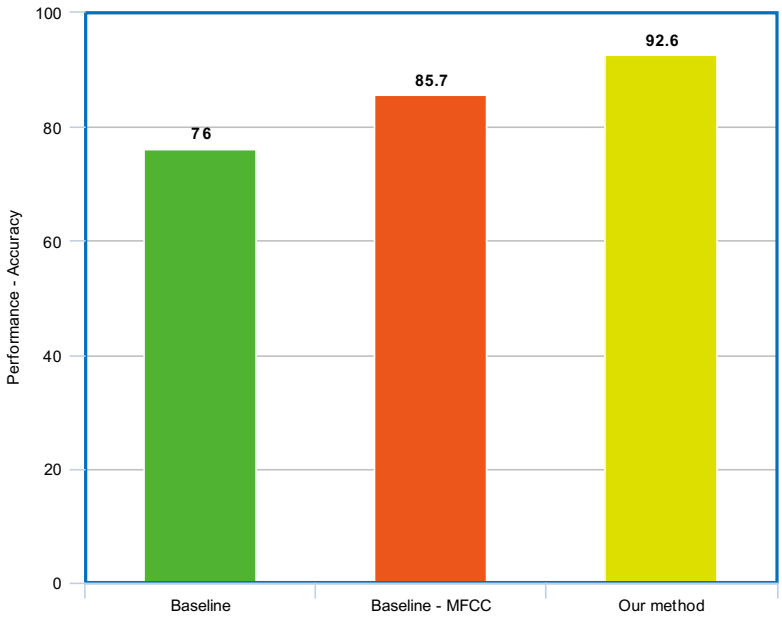
We also propose to use another baseline strategy for comparison. From Section 4.4.1, we observe that MFCC emerges as a good feature for structure segmentation. We propose to use this insight in our second baseline strategy, where we use MFCC features for training our CRF model. The model is trained as per the training procedure described in Section 4.5.1, but using MFCC features instead of features discovered from our neuro-, cognitive-, and social- psychology literature survey and user study on AMT. We evaluate the model by testing it on our test set of music tracks.

4.7. RESULTS

In this section, we report the performance of our method and compare it against the baseline detector. In total, we have 864 music tracks in our dataset, and we use 608 tracks for training, 100 tracks as development set and the remaining 156 tracks for evaluation. As indicated earlier, we use two different evaluation metrics: accuracy and distance to evaluate how our proposed approach answers our research question (RQ2). Figures 4.6 and 4.7 report the performance of our method and compares it against the two baseline strategies. We observe that our method performs much better than the baseline in terms of both the metrics. Our method obtains an accuracy of 92.6%, while the baseline achieves an accuracy of 76%. Observing the distance metric, which measures the average distance between the actual derail moment and the predicted derail moment across the test set, is also much better when compared to that of the baseline. On an average, our method detects derail moments 7.9 seconds before the actual derail moment, while

the baseline detects 20.1 seconds before the actual derail moment. We also outperform the other baseline strategy using MFCC as a feature for building the model, as observed in Figures 4.6 and 4.7. Though we improve upon the baseline strategy, MFCC features still fall behind the different features, in terms of accuracy and distance, motivated from our neuro-, cognitive-, and social- psychology literature survey and the user study. Looking at this result from the perspective of our envisioned music player, we are very close to the actual derail moment and we can make a switch to the next track before the listener hits the derail moment.

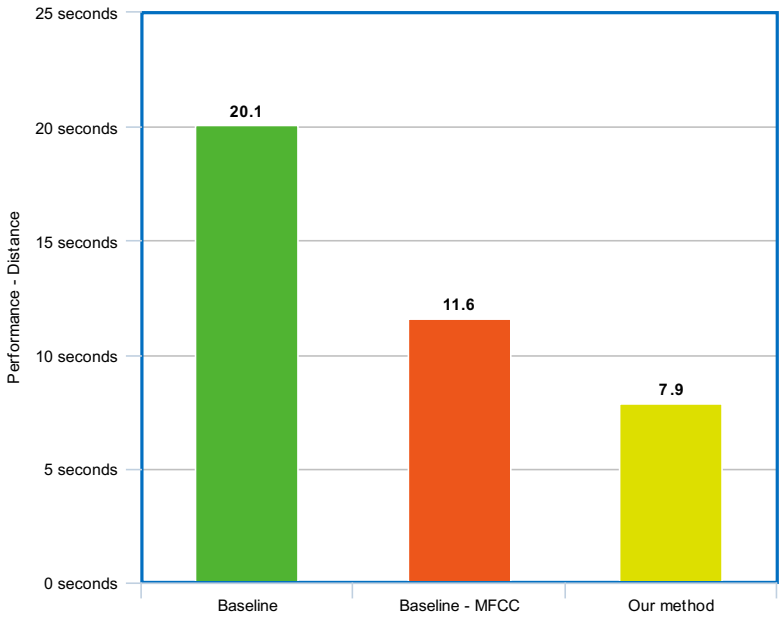
Figure 4.6: Comparison of accuracies for the two baseline strategies and our proposed method.



While collecting our dataset from AMT, we asked an additional question to the participants on the nature of the derail moment they are providing i.e., if the timestamp mentioned by the participant is "the" derail moment or does the music at the timestamp just starting to distract the listener. In other words, if the derail moment is a punctual one or a gradual one (Section 4.1). Here, we want to investigate how the presence of examples with this information in the training set can affect the performance of the derail moment detector. We have two separate experiments in which we carry out our investigation.

In the first experiment, we remove examples labelled as gradual derail moments i.e., remove the example tracks where the listeners say that the music is starting to distract them. From the training set of 608 tracks, 110 tracks (19%) are labelled to have a gradual derail moment as per our user study participants. We remove the 110 derail moments from our training set and retrain the model. We train the model as mentioned in Section 4.5.1 and perform the evaluation on our test set, which remains the same as the one used

Figure 4.7: Comparison of distance (in seconds) for the two baseline strategies and our proposed method.



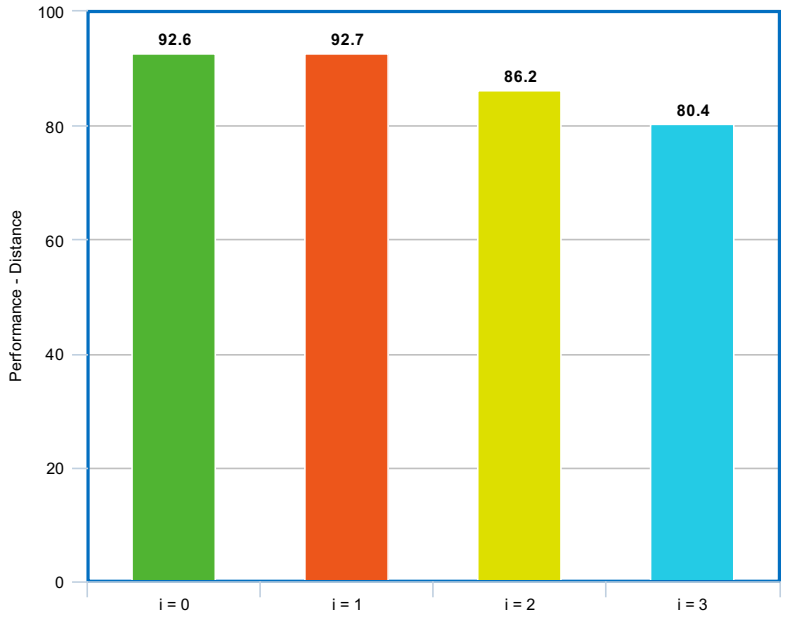
in our previous experiments. Removing these examples from the training set provided a slight improvement in the performance on the test set. We obtain an accuracy of 94.2% while the distance between the predicted and the actual derail moment is 7.4 seconds. There is an improvement of 1.2% in terms of accuracy and the average distance between the actual and predicted derail moment goes down by 0.5 seconds.

In the second experiment, we investigate another aspect of the derail moment i.e., since there are examples labelled as gradual derail moments, the actual derail moment (timestamp provided by the participant) might occur at some point after the indicated timestamp. Considering this information and the observation that derail moments occur close to the segment boundaries, we label the segment boundaries after the mentioned timestamp as the actual derail moment. We perform multiple sub-experiments where we mark the actual derail moment i segments after the derail moment mentioned by the participant, where $i = \{1,2,3\}$, in the training set. Through this experiment, we investigate if we improve the predicting capability of the trained model by using the information that for some examples, the derail moment can occur a while after the reported timestamps.

We report the results of this experiment in Figures 4.8 and 4.9, where we report the two evaluation metrics for different values of i. Observing the figure, we see that the results almost remain the same for $i = 1$, in comparison to the results reported for our method in Figure 4.6.3. However, for values of i greater than 1, there is a drop in the performance of our method as illustrated in the figure. This could be because the actual derail moment is very close to the reported timestamps and labelling the succeeding

segments as derail moments is hampering the performance of our model.

Figure 4.8: Comparison of Accuracy by predicting derail moments $i = \{0,1,2,3\}$ segments after the ground-truth segment boundary.



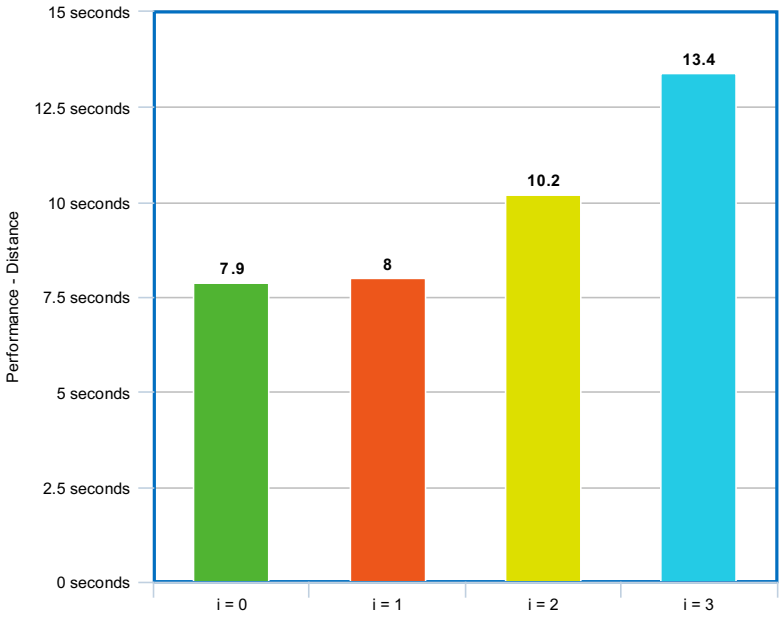
4.8. RESULTS ANALYSIS

In this section, we further analyse the results in order to understand their implications for possible applications and future work.

4.8.1. ABLATION ANALYSIS

In this section, we investigate how the performance of our method varies when we systematically remove certain features. Our feature set is motivated from the neuro-, cognitive-, and social- psychology literature survey and the user study. The feature set contains the following broad categories of features: low-level features, events, vocals. We compare our method by first removing the events feature, then the vocals feature and finally both these features. We evaluated the approach on the dataset collected from AMT and measured the following metrics: Accuracy and Distance. The results are reported in Figures 4.10 and 4.11. Observing the figures, we see a decline in performance, in terms of distance and accuracy, when we remove the vocal and event features. This provides an indication that each set of features adds information to the model that is useful to improve the performance.

Figure 4.9: Comparison of Distance (in seconds) by labelling derail moments $i = \{0, 1, 2, 3\}$ segments after the reported timestamps.



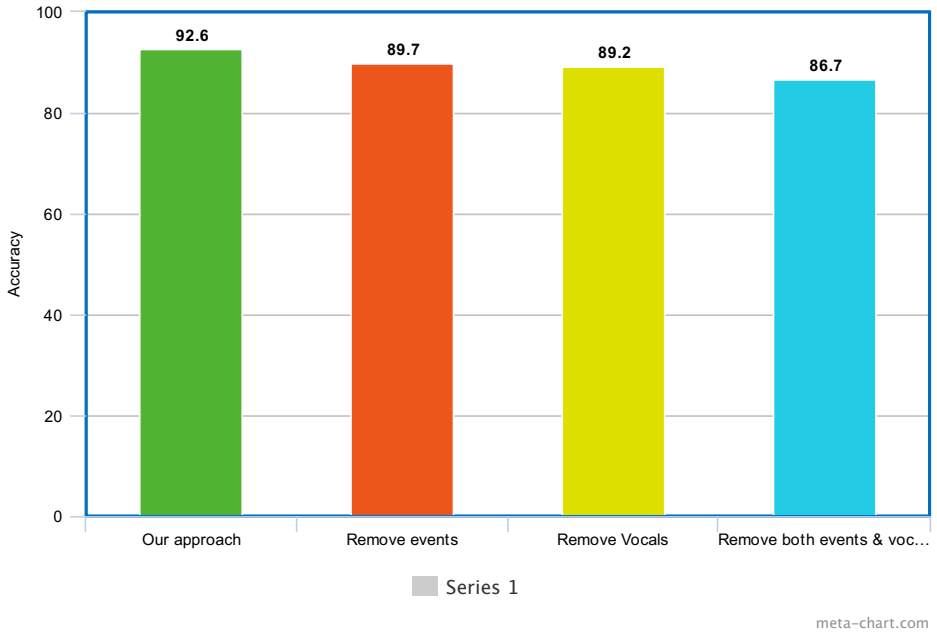
4.8.2. FALSE ALARM ANALYSIS

Until now, we have experimented on tracks that have a derail moment during its playtime as indicated by the participants on AMT. For our method to be useful for a user-oriented application, like the one introduced in Section 4.1, we want to investigate how our method fares when we provide clean tracks (i.e., tracks without any derail moment) as input. When we were to run our detector on a large music collection, we do not want it to detect a derail moment when there is none. An alternative scenario to our envisioned smart music player could be that we don't play a particular track if there is a derail moment in the track. In order to verify the performance of our method on music that does not have a derail moment, we analyse a set of clean tracks.

Our training set does not contain any such tracks, and hence we collect such clean tracks from friends and colleagues. We asked them to give us music tracks that they use while working on a task and which do not distract them. We collected 40 such tracks, without any derail moments, from 16 people who frequently listened to music while working.

We first perform structure segmentation on the clean tracks and extract features from each segment. We use the model trained on structural segments of music tracks used in our previous experiments and provide the clean tracks as input. Of the 40 tracks, we observe that our method does not detect a derail moment for 12 tracks, which is impressive considering that we did not have any clean tracks in our dataset. Such a performance indicates that the trained model has tried to learn the right characteristics of a derail

Figure 4.10: Comparison of Accuracy for our approach indicating the importance of events and vocals features.



moment.

Of the remaining 28 tracks, our method says there is a derail moment at the last segment boundary for 14 tracks. Again, this is impressive as the segments in the end are usually short, and there is a high chance that the artist wants to end the track with a bang. Manually listening to these tracks indicated that some of these had a very loud ending when compared to the rest of the track.

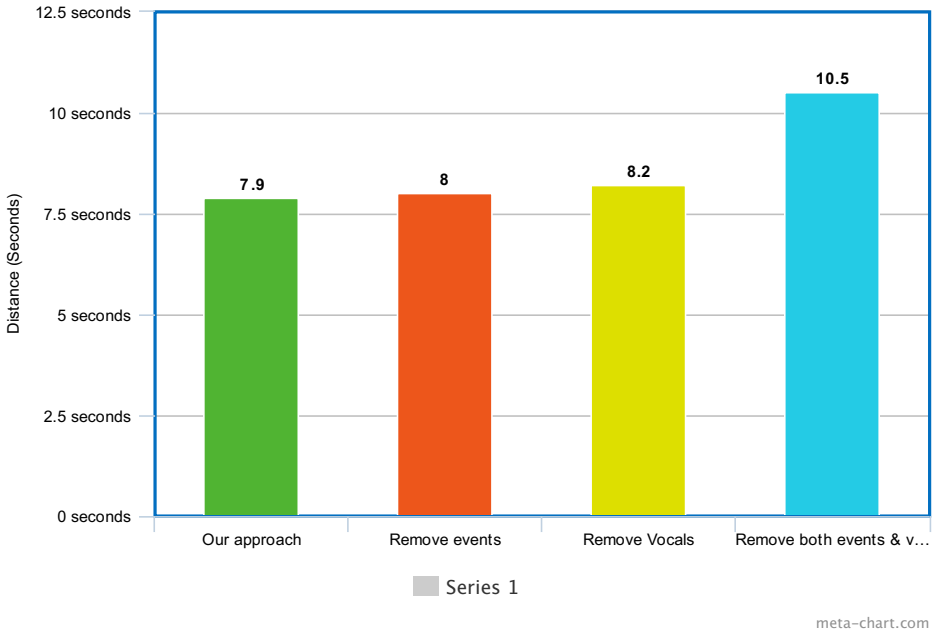
For ten tracks, our detector predicted a derail moment approximately after 50% of the track had been played out. For our envisioned application, where the music player smoothly blends into the next track when it sees a derail moment approaching, this performance is good as the player will be able to play a significant part of the clean music tracks. For the remaining four music tracks, we detect a derail moment within the first few seconds.

This experiment provides an analysis of our training model when we see a different kind of data than it has been trained on. Using the results of this experiment, we answer our third research question on recognising derail moments in a given music track. In the future, we will further perform a large-scale investigation in this direction.

4.8.3. FAILURE ANALYSIS

In the chapter, we proposed a method to detect a derail moment in a given music track where a listener might get distracted from his/her work. There are several components

Figure 4.11: Comparison of distance (in seconds) for our approach indicating the importance of events and vocals features.



in our method which may effect the performance of our method. In this section, we investigate the failure cases, where the derail moment was not rightly detected.

We begin by investigating the effect of high-level events on the performance of the derail moment detector. Drop-like events are high-level events, and hold a certain structural significance in a music track i.e., drops play a major role in the unfolding of a dance music track and are of a certain length. In addition to the participants mentioning a drop being a derail moment, we also listened to some music tracks in our dataset looking for drop-like events. One of our observations was that a drop-like event is not always a derail moment as we have two examples in our dataset where the listener has marked the derail moment after the drop. In one of the examples, which is an electronic music track, the listener marked the vocals (occurring after a drop) as a derail moment instead of the drop itself. In this example, the drop is a very subtle one but the vocals after the drop start with screaming. Other scenarios include a particular drop-like event that is not detected by the drop detector [6].

There are a few other situations where the performance of our drop detector is affecting the performance of our derail moment detector. There are short and subtle drop-like events in our dataset and sometimes these subtle events are detected by the drop detector. Many times, these subtle events are missed by the drop detector. According to our observations on the dataset, there is no guarantee that the subtle drop-like events would occur close to the derail moments and hence affects the performance of our method.

From our survey of the literature on the effects of music on the brain and also as per our intuition, drop-like events form a major part of musical characteristics that can attract a listener's attention towards the music. However, there is comparatively little direct mention about drop-like events from the participants on AMT. A few participants mention directly about the drop being a distracting element, but many participants mention the characteristics of a drop indirectly. For example, a sudden change in tempo is one of the frequent comments from the participants and this occurs as part of a drop.

Structure segmentation is an integral part of our derail moment detector and we investigate how it effects the performance of our method. As we indicated earlier, a majority of the derail moments are very close to a segment boundary and there are three possibilities: the derail moment occur before, after, or coincide with the segment boundary. When the derail moment coincides with the segment boundary, or if it happens after the segment boundary we do not have a problem. There were a few cases (14% of the test set) in which the derail moment occurred just before the segment boundary. In these cases, the machine learning model predicted a derail moment at the segment boundary preceding the actual derail moment. This had a negative effect on our evaluation metric measuring distance between the predicted and actual derail moment. Considering the average segment size to be around 20 seconds, the predictions where the model predicted a derail moment before the actual derail moment adds a higher value to the metric. Having these examples in the test set increases the overall distance metric by 1.3 seconds.

4.9. CONCLUSION AND FUTURE WORK

The aim of this chapter was to 1) determine whether we can justifiably infer an element of universality in the nature of derail moments, and 2) to develop a method to possibly detect the derail moments automatically. Our first task was to identify a large user base and a specific kind of task that people perform on a daily basis. AMT proved to be an ideal setting in our case, as there are specific tasks and a large number of users working on these tasks. We tapped into this user population and asked them questions on the kind of music that would distract them from their task. We focused on a specific kind of task: text transcription task, which is a well-defined type of task and a popular one on AMT. This gave us a reasonably sized dataset which we could use to evaluate our proposed method. The dataset comprising the YouTube ids of music tracks and their corresponding annotation of a derail moment is publicly available on OSF (https://osf.io/n3zmp/).

Once we identified the AMT task, we moved towards answering our research questions. We conducted a neuro-, cognitive-, and social- psychology literature survey looking for models which can explain how music attracts the attention of the listener (e.g., BRECVEMA [14]). Different dimensions of this model suggest different musical characteristics that can potentially attract the attention of the listener. Our assumption is that, if the listener's attention is diverted towards the music, they are distracted from the task they are performing. Studying the BRECVEMA model gave us an indication about the musical characteristics that are possibly independent of the listeners' background and preferences. We also conducted a user study on AMT about the distracting elements of a music track and performed a card sort of the responses from participants. A number of

categories emerged from the card sorting process and correspond to the different ways a listener can get distracted. Combining the results of the card sorting and literature survey, we find further evidence in favour of our hypothesis on the existence of an element of universality to the phenomenon of derail moment.

The literature survey and the user study provided us with a list of musical features that we could use to guide the building of a machine learning model to automatically detect derail moments. We conceptualised the derail moment as a musical event, and used features discovered from the literature survey to train a conditional random field model to detect these events. Experimental results indicated that our model performs well in comparison to a baseline and can be utilised in our envisioned music player to skip to the next track when approaching a derail moment. A significant finding from our experiments is that our method is very good at detecting a punctual derail moment with a very high accuracy and a minimal distance score. We also performed a qualitative experiment to analyse what happens if we feed "clean" tracks, without any derail moments, into our trained model. The results of our experiments are encouraging and inspire our future work towards building user-oriented applications.

An immediate direction of research would be to include "clean tracks" in our training set, which will help improve the performance of derail moment recognition for any given music track. This will be the next step towards building our envisioned music player as we would have to tackle a large of variety of music tracks that may or may not contain derail moments. Currently, we make a binary decision on the position of a segment boundary but assigning a confidence score to each segment boundary would be more useful for the envisioned music player. This could be another interesting direction of research and a possible way to improve the detection performance. Similarly, the output of the labelling algorithm also plays an important role in our training procedure (Section 4.5.1). In this chapter, we did not optimise for the cluster quality but that could be an important and interesting direction of research.

The main focus of this chapter was to investigate what musical characteristics play an important role while detecting a derail moment. It is a proof-of-concept that investigates the nature of a derail moment and proposes a method to automatically detect it. While building this proof-of-concept, we did not optimise for the best machine leaning model. Our choice of CRF was based on the insights that temporal information is important while predicting a derail moment. However, we could also investigate more sophisticated models, like Recurrent Neural Networks (RNN), to predict derail moments. This could be a potential future research direction, where we can investigate the correlation between the low-level musical features and the individual responses of the neurons in the neural network.

A noticeable property in our dataset is that the derail moments mostly occur during the initial part of the track (first 30% of the track). This could be because it is relatively easier to re-find the derail moments that occur during the beginning of the track. This property could have translated into predicting derail moments in the beginning of the track. However, our small-scale experiment on 40 clean tracks provided an indication that there is not a big effect of this property on the final outcome. We observe that the model predicts a derail moment at the end of the track for some examples. This result it encouraging but we need further experiments to how this property can effect the perfor-

mance of our method.

An important direction of future research is to incorporate user's preferences for certain kind of music while working. A recent work by [34] proposed and evaluated a system—FocusMusicRecommender—which can recommend music for focusing on a specific task. One of the important findings of Yakura et al. is that the music which is neither liked nor disliked can help listeners to focus on their work. Collecting users' musical preferences is done via manual input by providing buttons like "keep listening" and "skip". Our method to automatically detect a derail moment can potentially reduce the manual input from the listener by having it as a plugin in such a music player.

An important aspect of listening to music while working is to ensure that the user remains in a flow state [3]. Some of the existing services, like Focus@Will, gradually change the music after a fixed period of time. An interesting and useful future research direction could be to automatically identify music that can keep a user in this flow state.

REFERENCES

[1] K. Yadati, M. Larson, A. Demetriou, C. C. Liem, and A. Hanjalic, *Automatic identification of derail moments in focus music,* in *(Under review).*

[2] A. C. North, D. J. Hargreaves, and J. J. Hargreaves, *Uses of music in everyday life,* (University of California Press Journals, 2004) pp. 41–77.

[3] A. Demetriou, M. Larson, and C. C. S. Liem, *Go with the flow: When listeners use music as technology,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2016).

[4] M. Kamalzadeh, D. Baur, and T. Möller, *A survey on music listening and management behaviours,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2012).

[5] G. Bonnin and D. Jannach, *Automated generation of music playlists: Survey and experiments,* in *ACM Computer Surveys*, Vol. 47 (2014) pp. 1–35.

[6] K. Yadati, M. Larson, C. C. S. Liem, and A. Hanjalic, *Detecting socially significant music events using temporally noisy labels,* in *IEEE Transactions on Multimedia*, Vol. 20 (2018) pp. 2526–2540.

[7] S. Chachada and C. C. J. Kuo, *Environmental sound recognition: A survey,* in *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference* (2013).

[8] J. Schlüter and S. Böck, *Improved musical onset detection with convolutional neural networks,* in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (2014).

[9] X. Shao, Z. Cheng, and M. S. Kankanhalli, *Music auto-tagging based on the unified latent semantic modeling,* in *Multimedia Tools and Applications*, Vol. 78 (2019) pp. 161–176.

[10] M. Müller, *Fundamentals of music processing: Audio, analysis, algorithms, applications,* (Springer International Publishing, Cham, 2015) pp. 167–236.

[11] X. Wang, D. S. Rosenblum, and Y. Wang, *Context-aware mobile music recommendation for daily activities,* in *Proceedings of the ACM International Conference on Multimedia* (2012).

[12] R. Dias, M. J. Fonseca, and R. Cunha, *A user-centered music recommendation approach for daily activities.* in *Proceedings of the ACM Workshop on Content based Recommender Systems* (2014).

[13] K. Yadati, C. C. Liem, M. Larson, and A. Hanjalic, *On the automatic identification of music for common activities,* in *Proccedings of the ACM International Conference on Multimedia Retrieval* (2017).

[14] P. N. Juslin, *From everyday emotions to aesthetic emotions: towards a unified theory of musical emotions.* in *Physics of life reviews*, Vol. 10 3 (2013) pp. 235–66.

[15] W. Trost and P. Vuilleumier, *Rhythmic entrainment as a mechanism for emotion induction by music,* in *The Emotional Power of Music* (2013) pp. 213—-225.

[16] P. Belin, R. J. Zatorre, and P. Ahad, *Human temporal-lobe response to vocal sounds,* in *Cognitive Brain Research*, Vol. 13 (2002) pp. 17 – 26.

[17] M. Weiss, S. Trehub, E. Schellenberg, and P. Habashi, *Pupils dilate for vocal or familiar music,* in *Journal of experimental psychology, Human perception and performance*, Vol. 42 (2016).

[18] S. Sara and S. Bouret, *Orienting and reorienting: The locus coeruleus mediates cognition through arousal,* in *Neuron*, Vol. 76 (2012) pp. 130 – 141.

[19] J. Reynolds, A. Mcclelland, and A. Furnham, *An investigation of cognitive test performance across conditions of silence, background noise and music as a function of neuroticism,* in *Anxiety, Stress, & Coping*, Vol. 27 (2013) pp. 410—421.

[20] J. Yang, J. Redi, G. Demartini, and A. Bozzon, *Modeling task complexity in crowdsourcing,* in *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* (2016).

[21] R. T. Nakatsu, E. B. Grossman, and C. L. Iacovou, *A taxonomy of crowdsourcing based on task complexity,* in *Journal of Information Science*, Vol. 40 (2014) pp. 823–834.

[22] J. G. Fox and E. D. Embrey, *Music - an aid to productivity.* in *Applied ergonomics*, Vol. 3 4 (1972).

[23] K. Choi, G. Fazekas, and M. B. Sandler, *Automatic tagging using deep convolutional neural networks,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2016).

[24] O. Nieto and J. P. Bello, *Systematic exploration of computational music structure research,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2016).

[25] J. Serrà, M. Müller, P. Grosche, and J. L. Arcos., *Unsupervised music structure annotation by time series structure features and segment similarity,* in *IEEE Transactions on Multimedia*, Vol. 16 (2014) pp. 1229–1240.

[26] O. Nieto and T. Jehan, *Convex non-negative matrix factorization for automatic music structure identification,* in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (2013).

[27] J. Foote, *Automatic audio segmentation using a measure of audio novelty,* in *Proceedings of the IEEE International Conference on Multimedia Expo* (2000).

[28] B. Lehner, G. Widmer, and R. Sonnleitner, *On the reduction of false positives in singing voice detection,* in *Proceedings of the International Conference on Acoustics, Signal, and Speech Processing* (2014).

[29] J. Lafferty, *Conditional random fields: Probabilistic models for segmenting and labeling sequence data,* (2001) pp. 282–289.

[30] A. K. Andrew Demetriou, Andreas Jansson and R. M. Bittner, *Vocals in the music matter: The relevance of music in the minds of listeners,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2018).

[31] S. D. You, Y.-C. Wu, and S.-H. Peng, *Comparative study of singing voice detection methods,* in *Multimedia Tools and Applications*, Vol. 75 (2016) pp. 15509–15524.

[32] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji, *Improving music source separation based on deep neural networks through data augmentation and network blending,* in *Proceedings of the International Conference on Acoustics, Signal, and Speech Processing* (2017).

[33] R. Sonnleitner, B. Niedermayer, G. Widmer, and J. Schlüter, *A simple and effective spectral feature for speech detection in mixed audio signals,* (2012).

[34] H. Yakura, T. Nakano, and M. Goto, *Focusmusicrecommender: A system for recommending music to listen to while working,* in *Proceedings of the International Conference on Intelligent User Interfaces* (2018).

5

Conclusion

In this thesis, we presented the results of our investigation on how we can extend MIR research from analysing "what music is" to support "what music does", thereby increasing the value of music to people. To address this challenge, we looked at two related tasks: music event detection and identifying music for activities. We presented our methodology, findings, and empirical results in the three technical chapters of this thesis. Here, we reflect on our results and provide future recommendations to take this research forward.

5.1. MUSIC EVENT DETECTION

In Chapter 2.1 and 2.2, we presented our approach to detect high-level events in a music track. We identified EDM as a test bed and picked the three most important events in this genre: Drop, Build, and Break. These events have a certain emotional impact on the listener and elicit reactions on social media platforms: in other words, they are socially significant, and invite actions in users. As a consequence, the detection of these types of events can be seen as a first step into investigating "Music as Technology" tasks. Upon identifying the events we want to detect, we moved onto answering the question how to effectively and efficiently detect them. We utilised manually acquired annotations and the timed comments on SoundCloud to build machine learning models that can detect the three events of interest.

Regarding the effectiveness of the event detection, one of our important findings is that structural segmentation is an important step towards detecting high-level musical events. As structural segments indicate a change in texture of the track, we identified that the starting points of the events are very close to, or sometimes coincide with, the segment boundaries. We also explored a variety of audio and image features for classifying the segments into events. In addition to the image features proposed in the literature [1], we also explored other image features in our experiments. Our finding that image features perform better than the audio features is consistent with the literature [2].

Regarding the efficiency of event detection, we explored the usability of timed comments as training labels to, at least partially, replace expensive and tedious manual annotation. Though noisy in nature, the timed comments were found to be useful in this respect. Our experiments demonstrate that using timed comments alone as training labels outperform a naive baseline, especially in terms of the metric ea_dist, which measures the distance between the starting point of the actual event and the predicted event. Though the ea_dist is still big (around 18 seconds for a drop), it is reasonable for an application like non-linear access because these events are longer in duration. However, an interesting future work could be to conduct a user-study with DJs to further investigate whether this value of ea_dist is good.

When we combine a certain proportion (40%) of these noisy timed comments with manually acquired expert annotations, we obtain results that are very close to the ones we would expect if we had the expert annotations for the whole training set, especially for break event in terms of f-scores. For the other two events, the combination is 20% - 80% to get an f-score close to that of a model trained with 100% expert annotations. However, in terms of the distance metric (ea_dist) all three events obtain a good performance at a combination of 40% - 60% (for non-linear access application). Our experimental results demonstrate that we can reduce the manual labelling effort by utilising the freely avail-

able timed comments as weak training labels. A part of the dataset used in Chapter 2.1 is uploaded on the Open Science Framework, to encourage other researchers to utilise timed comments in their research.

Looking ahead, we identify some interesting directions of research based on our experimental findings. We chose EDM as a test bed for our experiments, and the three events of interest were a natural choice looking at the data. However, our findings could be useful in other music domains as well. In movies, for example, music is used as a tool to build up anticipation in a scene and the structure of such music is very similar to a buildup in EDM. Another observation from SoundCloud data was that the comment density is higher around a drop, though these comments do not explicitly mention the word "drop". We can exploit such patterns in users' feedback to identify other popular events in music and in a broader variety of music genres.

Another important aspect which we could further investigate are the methods to denoise the training labels. We proposed one strategy that can be considered as a combination of classification filtering and ensemble methods. There are few other strategies [3] that can help us denoise the training labels, like boosting based methods, graph-based methods or kNN based methods. We can explore these strategies to check how they improve the classification performance. One of the challenges in our dataset is that the noise is two-fold. There is a temporal noise in addition to the semantic noise of the timed comments. We need to come up with label denoising strategies that can help us in this scenario.

Deep learning has been garnering a huge amount of attention in music analysis. There is also recent literature on how we can train Convolutional Neural Networks (CNNs) with noisy labels [4]. Most of this recent literature is about noisy labels for images from social media platforms, while our case is different because of the two-fold noise. A direction of research could be to work on developing models and strategies to deal with this kind of noise.

5.2. MUSIC FOR COMMON ACTIVITIES

In Chapter 3, we explored the idea of using music as a tool to accomplish another activity. Our hypothesis was that we need to extend existing MIR methods for such a task. We provide empirical results that proves our hypothesis to be correct.

Through a data-driven approach, we establish the following common activity categories, for which music is available on social media sharing platforms (YouTube in our case): Workout, Study, and Relax. One of the major findings from our research is that existing metadata like genre, artist, or instrument do not help in classifying a music track into one of the activity categories. Listeners could prefer multiple genres of music across different activities. We presented both qualitative and quantitative evidence to support our hypothesis that genre and instrument information is not sufficient for out task.

Another important finding is that the traditional way of feature extraction i.e., extracting features from a small window (50-100 ms), does not give good classification performance. We explored different window sizes for extracting features and found that a 25-second window gives a good classification performance, which is way more than the traditional window size (50-100 ms). Through carefully designed experiments, we also showed that other existing approaches, like extracting a fixed length segment from

the middle of a track and using it for classification (genre recognition) instead of using the whole track, is not helpful. Our focus was entirely on identifying the appropriate window size for feature extraction and we chose a simple classifier (k-NN). Future work could involve investigating more advanced classification and feature selection strategies.

Additionally, we explored a variety of low-level and high-level features for classification. Our findings indicate that features related to rhythm are very important in classifying a music track into an activity category. Presence/absence of events like "drop" also makes a lot of difference in the classification performance. This is an interesting result because our previous model, which could detect drops in a given music track, can also be used as a feature to classify music tracks into activity categories. Workout music had more drop-like events than the other two activity categories: study and relax. Intuitively, this result makes sense: people sometimes need to push themselves physically while working out, and a drop-like event can motivate them to do so. On the other hand, one would not want to listen to a drop-like event when one is relaxing.

5.3. DERAIL MOMENTS IN FOCUS MUSIC

Taking inspiration from our work on identifying music for common activities, we wanted to delve deeper into the phenomenon of "music as technology". One of the limitations of our previous work on identifying music for common activities was that we did not have any negative examples for the three activity categories. Given any music track, we would put it into one of the three activity categories. Additionally, people have different musical tastes (in terms of genre, artist etc.) and prefer to create their own playlists while doing an activity, rather than using an automatically created mix or playlist [5]. These limitations brought us to a user scenario, in which the user wants to listen to his/her own music collection, but does not know if a specific track is suitable for the chosen activity or not. We define our end user application as follows: The user has created a playlist of his/her favourite tracks and started working on a task. Our system will identify the point in each music track when it starts becoming unsuitable for the task (derail moment) and before that point arrives in a track, the player smoothly blends into the next track.

We used Amazon Mechanical Turk (AMT) as our test bed, because it contains a variety of well-defined tasks and thousands of users working on those tasks. Through a text transcription task on AMT, we asked users to give us examples of derail moments they find while working online. We hypothesised and found evidence that the concept of a derail moment has a universal element to it in addition to the listeners' personal preferences from the following sources: neuroscience/psychology literature and a pilot study on AMT.

One of the major findings from our research reported in Chapter 4 is a list of musical characteristics that can create a derail moment in a given music track. We surveyed literature in neuroscience and came up with a list of features that form the basis for our machine learning model that can automatically detect a derail moment. Through our experiments, we found that we can indeed automatically detect a derail moment from the features extracted from the music signal.

Based on our experimental findings in Chapters 3 and 4, we list potential future directions to pursue. An immediate direction to investigate further would be to expand the research on automatic playlist generation to address the playlist suitability for a given

task [6]. Combining the users' taste data with our work on identifying music for common activities, we could build a hybrid recommender system that can be of help to listeners with specific needs. In our investigation, we looked at very specific activities, like working, to provide a proof of concept that we can indeed automatically identify music suiting a specific situation. We can look at a host of other activities/situations where music can help people to move forward.

An exciting opportunity to build automatic systems that make music more useful to people would be to talk to human curators who create playlists for specific situations (Google Play, Spotify, Apple Music). By understanding the characteristics they base their selection of tracks for a particular situation, we can devise automatic ways of extracting such information from music tracks. This will exponentially scale up the choice of tracks to choose from while building a playlist for a specific situation/activity. A good way to start in this direction could be to look at the playlists/mixes created by DJs/artists on YouTube, Spotify, or SoundCloud on similar lines as our work in Chapter 3.

Our work involved building a proof of concept for identifying music for common activities, and identifying a derail moment where a music track could potentially become distracting for an activity. When we started off with this research, we did not know what to expect and we wanted to thoroughly understand different musical characteristics that can affect the classification performance. With the recent development on explainable deep learning models [7], we could explore these complex architectures to see if we can improve our classification performance further. One of the limitations, at least in the work on detecting a derail moment, is the limited availability of data as it is expensive and time consuming to acquire. Many of these deep learning models require a large amount of data and this is a limitation in exploring these models. For this purpose, we could investigate transfer learning approaches [8].

5.4. FINAL REMARKS

Ubiquitous availability of music and the unprecedented growth of the internet has made music more accessible than ever. People can listen to music anywhere, anytime with just the click of a button or simply touching the screen of their phones. This has led to an increased use of music in a variety of situations. Nowadays, people not only listen to music for entertainment, but use it as a tool to accomplish another activity they are simultaneously involved in. While traditional music analysis methods have focused on understanding the meaning of music, in this thesis, the focus was on researching how retrieval systems can incorporate the effect music has on its listener.

The results presented in this thesis are promising and we have answered our initial research question "how to develop automatic music analysis tools that can broaden the usefulness of music in terms of the effect it has on the listeners?", by developing methods and demonstrating the experimental results in our technical chapters. Through our methods and experiments, we highlight the significance of social media sharing platforms (SoundCloud, YouTube) and crowdsourcing platforms (AMT) in understanding the needs of the user while consuming music. These platforms provide us with an opportunity to collect both implicit and explicit feedback that can help us in building systems useful for the consumer. The research proposed in this thesis can be considered as a basis to building a more holistic music recommendation system that can recommend

music for specific situations.

REFERENCES

[1] J. W. Dennis, *Sound event recognition in unstructured environments using spectrogram image processing,* (2014).

[2] K. Yadati, M. Larson, C. C. S. Liem, and A. Hanjalic, *Detecting socially significant music events using temporally noisy labels,* in *IEEE Transactions on Multimedia*, Vol. 20 (2018) pp. 2526–2540.

[3] B. Frénay and M. Verleysen, *Classification in the Presence of Label Noise: A Survey,* in *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 25 (2014) pp. 845–869.

[4] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, *Learning from massive noisy labeled data for image classification,* in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2015).

[5] M. Kamalzadeh, D. Baur, and T. Möller, *A survey on music listening and management behaviours,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2012).

[6] G. Bonnin and D. Jannach, *Automated generation of music playlists: Survey and experiments,* in *ACM Computer Surveys*, Vol. 47 (2014) pp. 1–35.

[7] J. Chen, F. Zhuang, X. Hong, X. Ao, X. Xie, and Q. He, *Attention-driven factor model for explainable personalized recommendation,* in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval* (2018).

[8] K. Choi, G. Fazekas, M. B. Sandler, and K. Cho, *Transfer learning for music classification and regression tasks,* in *Proceedings of the conference of International Society for Music Information Retrieval* (2017).

Acknowledgements

A big *THANK YOU* to everyone who was part of my PhD. Special mention to a few people who made my journey beautiful: Martha Larson (For her dedicated support throughout my PhD), Pramoda Chilakamarthi (For her patience during my PhD and the cover of this book), Mom & Dad (For letting me follow my dreams).

CURRICULUM VITÆ

Narasimha Karthik YADATI

29-11-1984 Born in India.

EDUCATION

2003-2007 Undergraduate studies in Computer Science
International Institute of Information Technology Hyderabad
India

2010-2013 Master of Science in Computing
National University of Singapore
Singapore

2014-2019 PhD. Computer Science
Delft University of Technology
The Netherlands

EXPERIENCE

2007-2010 Software Engineer
International Business Machines (IBM) Bangalore
India

2017 Research Intern
Technicolor Rennes
France

2018-2019 Machine Learning Expert
Media Distillery Amsterdam
The Netherlands

LIST OF PUBLICATIONS

4. Karthik Yadati, Andrew Demetriou, Martha Larson, Cynthia C. S. Liem, and Alan Hanjalic, Automatic identification of derail moments in focus music. (To be submitted to TISMIR).

3. Karthik Yadati, Cynthia C. S. Liem, Martha Larson, and Alan Hanjalic. On the Automatic Identification of Music for Common Activities. In Proceedings of the 2017 ACM International Conference on Multimedia Retrieval 2017.

2. Karthik Yadati, Martha Larson, Cynthia C. S. Liem and Alan Hanjalic. Detecting Socially Significant Music Events using Temporally Noisy Labels, in IEEE Transactions on Multimedia, vol. 20, no. 9, pp. 2526 - 2540, 2018.

1. Karthik Yadati, Martha Larson, Cynthia C. S. Liem and Alan Hanjalic. Detecting Drops in Electronic Dance Music: Content based approaches to a socially significant music event. In Proceedings of the International Society for Music Information Retrieval 2014.