INSTITUT
POLYTECHNIQUE
DE PARIS

NNT : 2021IPPAT039

Thèse de doctorat

Personalized Audio Auto-tagging as Proxy for Contextual Music Recommendation

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°626
École doctorale de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat : Signal, Images, Automatique et Robotique

Thèse présentée et soutenue à Paris, le 16 Décembre 2021, par

KARIM M. IBRAHIM

Composition du Jury :

Talel Abdessalem Professeur, Télécom Paris	Président
Markus Schedl Professeur, Johannes Kepler University Linz	Rapporteur
Jean-François Petiot Professeur, Ecole Centrale de Nantes	Rapporteur
Kyogu Lee Professeur, Seoul National University	Examinateur
Elena Cabrio Assistant Professor, Université Côte d'Azur, Inria, CNRS	Examinateur
Geoffroy Peeters Professeur, Télécom Paris	Directeur de thèse
Elena Epure Research Scientist, Deezer	Co-directeur de thèse
Gaël Richard Professeur, Télécom Paris	Invité

Preface

This work has been done between October 2018 and October 2021. It is a collaboration between Deezer and Télécom Paris. The thesis has been carried out in both Deezer Research team, under the supervision of Elena V. Epure and ADASP team of LTCI laboratory, under the supervision of Geoffroy Peeters and Gaël Richard. This Work is part of the new-frontiers in Music Information Processing (MIP-Frontiers) project.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 765068.

Acknowledgements

It is hard to summarize the significance of the past three years, along with the experiences and people that had an influence on me, in just one page. The years leading to the writing of this thesis were very dense and intense, in a way that will take me some more years to fully digest. Nonetheless, I am already very grateful to every single one of them, and I would happily do it all over again.

I want to start with expressing my sincerest gratitude to my supervisors: Elena, Geoffroy, and Gaël. Without your continuous help, honest feedback, patient guidance, and kind understanding, it would not have been possible to finish this work. You have managed to get me to learn so much in such little time.

What seemed like a tough challenge in my project turned out to be an actual blessing. When my first supervisor in Deezer left the company to pursue a different career, I was thrown off and filled with doubts about the future of this project. However, once Elena took the role of my new supervisor, the future turned out to be very hopeful. Elena did not only provide me with technical and detailed help in my research, but also gave me continuous guidance in different aspects of life that helped me maneuver many difficulties.

I was also blessed to have two inspiring supervisors in Télécom Paris that perfectly complemented each other. Geoffroy always challenged me and pushed me to grow and aim higher, with an honest and direct feedback that helped me find the shortest path to move forward. At the same time, Gaël, with his calm and reassuring perspective, always helped me see the big picture and constantly reminded me that this is my training, pushing me to develop autonomy and independence as a researcher.

Beyond the intense research work, I am so grateful to have had a circle of friends and colleagues that supported me along the way. Your constant support has helped me go through this work during such challenging conditions that came with the Covid-19 pandemic. I will never forget the many fun times we had jamming together, or the difficult times of our complaining sessions. The inspiring friends I made in Deezer, Andrea, Tina, Darius, Bruno, and Félix. The journey companions at Télécom Paris, Kilian, Giorgia, Ondřej, and Javier. The friends I made through life, in the face of on-off quarantine restrictions, who always challenged my perspectives on life, Farouk, Manon, Maria, and Sarah. You all have made this so much easier, even though I do not say this enough.

Finally, special thanks to the wonderful MIP-Frontiers project, Télécom Paris, the EU research and innovation programme, and Deezer, for putting together such an amazing project. Even though the pandemic stood between us and the full potential of this project, it was still a fantastic opportunity to learn, grow, and have a continuous supporting network that I believe will last forever.

I also would like to thank Markus Schedl and Jean-Francois Petiot for the time they dedicated to review my manuscript. It takes much effort and dedication, and for that I am thankful.

Finally, none of this would have been possible if it was not for the support and belief I had from my family from the start and throughout. Without a doubt, this thesis is dedicated to my beloved parents. You are always remembered.

Abstract

The exponential growth of online services and user data changed how we interact with various services, and how we explore and select new products. Hence, there is a growing need for methods to recommend the appropriate items for each user. In the case of music, it is more important to recommend the right items at the right moment. It has been well documented that the context, i.e. the listening situation of the users, strongly influences their listening preferences. Hence, there has been an increasing attention towards developing recommendation systems. State-of-the-art approaches are sequence-based models aiming at predicting the tracks in the next session using available contextual information. However, these approaches lack interpretability and serve as a hit-or-miss with no room for user involvement. Additionally, few previous approaches focused on studying how the audio content relates to these situational influences, and even to a less extent making use of the audio content in providing contextual recommendations. Hence, these approaches suffer from both lack of interpretability.

In this dissertation, we study the potential of using the audio content primarily to disambiguate the listening situations, providing a pathway for interpretable recommendations based on the situation.

First, we study the potential listening situations that influence/change the listening preferences of the users. We developed a semi-automated approach to link between the listened tracks and the listening situation using playlist titles as a proxy. Through this approach, we were able to collect datasets of music tracks labelled with their situational use. We proceeded with studying the use of music auto-taggers to identify potential listening situations using the audio content. These studies led to the conclusion that the situational use of a track is highly user-dependent. Hence, we proceeded with extending the music-autotaggers to a user-aware model to make personalized predictions. Our studies showed that including the user in the loop significantly improves the performance of predicting the situations. This user-aware music auto-tagger enabled us to tag a given track through the audio content with potential situational use, according to a given user by leveraging their listening history.

Finally, to successfully employ this approach for a recommendation task, we needed a different method to predict the potential current situations of a given user. To this end, we developed a model to predict the situation given the data transmitted from the user's device to the service, and the demographic information of the given user. Our evaluations show that the models can successfully learn to discriminate the potential situations and rank them accordingly. By combining the two model; the auto-tagger and situation predictor, we developed a framework to generate situational sessions in real-time and propose them to the user. This framework provides an alternative pathway to recommending situational sessions, aside from the primary sequential recommendation system deployed by the service, which is both interpretable and addressing the cold-start problem in terms of recommending tracks based on their content.

Keywords - Music Auto-tagging, Context-aware, Music Recommendation, Interpretability, Playlist Generation, Music Streaming.

Résumé

La croissance exponentielle des services en ligne et des données des utilisateurs a changé la façon dont nous interagissons avec divers services, et la façon dont nous explorons et sélectionnons de nouveaux produits. Par conséquent, il existe un besoin croissant de méthodes permettant de recommander les articles appropriés pour chaque utilisateur. Dans le cas de la musique, il est plus important de recommander les bons éléments au bon moment. Il est bien connu que le contexte, c'est-à-dire la situation d'écoute des utilisateurs, influence fortement leurs préférences d'écoute. C'est pourquoi le développement de systèmes de recommandation fait l'objet d'une attention croissante. Les approches les plus récentes sont des modèles basés sur les séquences qui visent à prédire les pistes de la prochaine session en utilisant les informations contextuelles disponibles. Cependant, ces approches ne sont pas faciles à interpréter et ne permettent pas à l'utilisateur de s'impliquer. De plus, peu d'approches précédentes se sont concentrées sur l'étude de la manière dont le contenu audio est lié à ces influences situationnelles et, dans une moindre mesure, sur l'utilisation du contenu audio pour fournir des recommandations contextuelles. Par conséquent, ces approches souffrent à la fois d'un manque d'interprétabilité.

Dans cette thèse, nous étudions le potentiel de l'utilisation du contenu audio principalement pour désambiguïser les situations d'écoute, fournissant une voie pour des recommandations interprétables basées sur la situation.

Tout d'abord, nous étudions les situations d'écoute potentielles qui influencent ou modifient les préférences d'écoute des utilisateurs. Nous avons développé une approche semi-automatique pour faire le lien entre les pistes écoutées et la situation d'écoute en utilisant les titres des listes de lecture comme proxy. Grâce à cette approche, nous avons pu collecter des ensembles de données de pistes musicales étiquetées en fonction de leur utilisation situationnelle. Nous avons ensuite étudié l'utilisation de marqueurs automatiques de musique pour identifier les situations d'écoute potentielles à partir du contenu audio. Ces études ont permis de conclure que l'utilisation situationnelle d'un morceau dépend fortement de l'utilisateur. Nous avons donc étendu l'utilisation des marqueurs automatiques de musique à un modèle tenant compte de l'utilisateur afin de faire des prédictions personnalisées. Nos études ont montré que l'inclusion de l'utilisateur dans la boucle améliore considérablement les performances de prédiction des situations. Cet auto-tagueur de musique adapté à l'utilisateur nous a permis de marquer une piste donnée à travers le contenu audio avec une utilisation situationnelle potentielle, en fonction d'un utilisateur donné en tirant parti de son historique d'écoute.

Enfin, pour réussir à utiliser cette approche pour une tâche de recommandation, nous avions besoin d'une méthode différente pour prédire les situations actuelles potentielles d'un utilisateur donné. À cette fin, nous avons développé un modèle pour prédire la situation à partir des données transmises par l'appareil de l'utilisateur au service, et des informations démographiques de l'utilisateur donné. Nos évaluations montrent que les modèles peuvent apprendre avec succès à discriminer les situations potentielles et à les classer en conséquence. En combinant les deux modèles, l'auto-tagueur et le prédicteur de situation, nous avons développé un cadre pour générer des sessions situationnelles en temps réel et les proposer à l'utilisateur. Ce cadre fournit une voie alternative pour recommander des sessions situationnelles, en dehors du système de recommandation séquentiel primaire déployé par le service, qui est à la fois interprétable et aborde le problème du démarrage à froid en termes de recommandation de morceaux basés sur leur contenu.

***Mots-clés* - ** Auto-tagging musical, Context-aware, Recommandation musicale, Interprétabilité, Génération de listes de lecture, Streaming musical.

French summary

La musique est un élément fondamental de la culture humaine universelle qui remonte à des milliers d'années. Tout au long de cette période, la musique a eu différentes fonctions liées à divers aspects de la vie, des représentations religieuses aux divertissements. Pendant la majeure partie de cette période, la musique ne pouvait être jouée qu'en direct, c'est-à-dire avant l'invention de la musique enregistrée. Cette limitation a eu un effet sur la façon dont la musique est à la fois écoutée et jouée, car il s'agissait principalement d'un événement social. Depuis lors, les progrès technologiques ont entraîné des changements dans la façon dont la musique est jouée et consommée. En particulier, l'invention des services d'enregistrement et de diffusion en continu a modifié de façon permanente la façon dont nous écoutons la musique.

Les services de streaming de musique à la demande permettent à un utilisateur d'écouter instantanément toute musique enregistrée disponible dans leurs catalogues. Avec des millions de titres disponibles, l'exploration de la musique est entrée dans une nouvelle ère. Cette vaste quantité de possibilités a nécessité de nouvelles méthodes pour aider les utilisateurs à explorer et à retrouver la musique qu'ils souhaitent. Comme la plupart des catalogues en ligne, les services de streaming musical ont commencé à développer des algorithmes de recommandation à cette fin. Ces recommandeurs ont également entraîné un changement dans la façon dont les gens consomment la musique. Les utilisateurs ont indiqué qu'une bonne recommandation est l'une des principales raisons de choisir un service spécifique [PG13].

D'autre part, la disponibilité continue de la musique a fait que la musique est de plus en plus consommée comme une activité de fond. Les gens peuvent désormais écouter de la musique où et quand ils le souhaitent. Par conséquent, différents utilisateurs ont développé différents modèles d'écoute de la musique. Ils ont également développé des préférences différentes pour ces différentes situations. Il est donc devenu important de recommander non seulement les bons articles, mais aussi le bon moment. Il est bien documenté que le contexte, c'est-à-dire la situation d'écoute des utilisateurs, influence fortement leurs préférences d'écoute [NH96c]. C'est pourquoi on s'intéresse de plus en plus au développement de systèmes de recommandation sensibles au contexte [AT11].

L'une des propriétés les plus recherchées des systèmes de recommandation est la transparence et l'interprétabilité. Les systèmes de recommandation musicale contextuelle de pointe utilisent diverses techniques pour intégrer les informations contextuelles dans le processus de recommandation, par exemple des modèles basés sur les séquences qui prédisent les morceaux de la prochaine session en utilisant les informations contextuelles disponibles [HHM+20]. Cependant, la plupart de ces approches ne sont pas faciles à interpréter et ne laissent aucune place à l'implication de l'utilisateur. L'interprétabilité devient de plus en plus une priorité tant pour les utilisateurs que pour les services [ADNDS+20, VDH+18, ABB14]. Cet aspect est important car il permet d'établir la confiance et la compréhension du service fourni.

Une approche permettant d'atteindre l'interprétabilité consiste à utiliser des descripteurs lisibles par l'homme, semblables à ceux que les gens utilisent pour se décrire mutuellement de la musique. Ces descripteurs sont normalement ajoutés par les utilisateurs ou le service pour aider à organiser de grands catalogues. Cependant, avec des millions de pistes disponibles, l'annotation manuelle de la musique est une tâche difficile qui est également sujette au bruit. D'autre part, la découverte des descripteurs appropriés en analysant le contenu audio d'un morceau est une solution alternative. La recherche d'informations musicales (MIR) est un domaine interdisciplinaire qui aborde ce type de problèmes. MIR fait appel à la théorie musicale, à l'informatique, au traitement du signal et à l'apprentissage automatique afin d'extraire ou de générer des informations significatives liées à la musique.

L'un des objectifs fréquents de MIR est de combler ce que l'on appelle le "fossé sémantique" [ADNDS+20]. Le fossé sémantique fait référence au lien manquant susmentionné entre le contenu de la musique et un ensemble de descripteurs sémantiques humains. L'un des descripteurs les moins explorés est la situation d'écoute prévue. Peu d'approches précédentes se sont concentrées sur l'étude de la relation entre le contenu audio et ces influences situationnelles et, dans une moindre mesure, sur l'utilisation du contenu audio pour fournir des recommandations contextuelles. L'ajout de descripteurs personnalisés, c'est-à-dire de balises, aux pistes décrivant la situation d'écoute prévue de la piste par un utilisateur donné améliorerait considérablement l'exploration de la musique, l'organisation des catalogues et la fourniture de recommandations contextuelles interprétables.

Dans cette thèse, nous proposons les contributions suivantes :

Identification des situations pertinentes à l'écoute de la musique : Dans cette thèse, nous présentons une analyse approfondie des travaux antérieurs réalisés sur l'identification des situations pertinentes à l'écoute de la musique. Nous utilisons ces études antérieures pour collecter un large ensemble de situations potentielles, que nous étendons grâce à la similarité sémantique et aux mots-clés fréquemment associés sur les médias sociaux. Nous identifions 96 mots-clés décrivant plusieurs situations qui sont catégorisées en : *activité, temps, lieu et humeur*. De plus, nous avons identifié leur importance par leur fréquence lorsqu'ils apparaissent dans les titres des listes de lecture créées par les utilisateurs dans Deezer. Ces mots-clés constituent la base de toutes nos expériences futures, car ils décrivent les tags que nous souhaitons utiliser pour décrire les situations d'écoute. Cette procédure nous a permis de collecter 3 grands ensembles de données pour chaque expérience, qui ont tous été rendus publics pour des recherches futures. Ce type d'ensembles de données situationnelles à cette échelle est le premier du genre à être rendu public.

La relation entre le contenu audio et les situations d'écoute (potentiel des auto-taggers) : Grâce aux mots-clés issus de la première étude, nous avons développé une approche semi-automatique pour relier les pistes écoutées et la situation d'écoute en utilisant les titres des listes de lecture comme proxy, appuyés par un filtrage rigoureux. Grâce à cette approche, nous avons pu collecter le premier ensemble de données de pistes musicales étiquetées en fonction de leur utilisation situationnelle. Nous avons mené notre étude pilote sur l'exploitation des auto-taggers de musique pour identifier les situations d'écoute potentielles en utilisant le contenu audio afin d'établir une référence pour cette tâche. Enfin, notre analyse des résultats a renforcé notre hypothèse initiale selon laquelle certaines situations dépendent fortement de l'utilisateur.

Au cours de cette étude, nous avons été confrontés à un problème courant dans le cas d'ensembles de données à étiquettes multiples : les étiquettes manquantes. Compte tenu de la procédure utilisée pour collecter l'ensemble de données, nous avons identifié une méthode pour estimer notre confiance dans les étiquettes collectées. Nous avons ensuite utilisé cette confiance pour développer une perte pondérée basée sur la confiance pour tenir compte des étiquettes manquantes. Nos études ont validé l'utilité de cette approche dans l'apprentissage à partir d'un ensemble de données avec des étiquettes manquantes. La perte proposée est particulièrement utile dans le cas d'une architecture prédéfinie ou d'un réglage fin d'un modèle pré-entraîné, ce qui n'était pas le cas dans les approches précédentes traitant des étiquettes manquantes.

Dépendance de l'utilisateur et préférences d'écoute : Nous avons procédé à l'extension des autotaggers musicaux à un modèle tenant compte de l'utilisateur afin de faire des prédictions personnalisées. Les autotaggers précédents étaient tous uniquement dépendants de l'audio, un inconvénient que nous avons surmonté afin de l'adapter à notre problème. Nous nous sommes appuyés sur l'historique d'écoute des utilisateurs afin de modéliser leurs préférences globales. Nos évaluations centrées sur l'utilisateur ont montré que l'inclusion de l'utilisateur dans la boucle, représentée par son historique, améliore considérablement les performances de prédiction des situations. Cet auto-tagging de la musique en fonction de l'utilisateur nous a permis d'étiqueter une piste donnée à travers le contenu audio avec une utilisation situationnelle potentielle, en fonction d'un utilisateur donné à travers son historique d'écoute.

Inférer automatiquement la situation d'écoute Enfin, pour réussir à utiliser cet outil pour une tâche de recommandation, nous avions besoin d'un outil différent pour prédire les situations actuelles potentielles d'un utilisateur donné. À cette fin, nous avons développé un modèle pour prédire la situation à partir des données transmises par l'appareil de l'utilisateur au service, et des informations démographiques de l'utilisateur donné. Nos évaluations montrent que les modèles peuvent apprendre avec succès à discriminer les situations potentielles et à les classer en conséquence.

En combinant les deux modèles, l'auto-tagger et le prédicteur de situation, nous avons développé un cadre pour filtrer l'ensemble des pistes potentielles en temps réel, sur la base de la situation prédite, avant de déployer les algorithmes de recommandation traditionnels et de proposer les résultats à l'utilisateur. Ce cadre fournit une voie alternative pour recommander des sessions situationnelles, en dehors du système de recommandation séquentielle primaire déployé par le service, qui est interprétable à travers les tags. Notre évaluation a montré que le préfiltrage de ces sessions situationnelles avec les tags correspondants améliorait significativement les performances de l'algorithme de recommandation traditionnel lorsqu'il était comparé aux sessions situationnelles.

Table of Contents

Acronyms

ADASP	*Audio Data Analysis & Signal Processing*. 2, 13
API	*Application Programming Interface*. 13, 61, 87, 92
AUC	*Area Under Curve*. 13, 31, 32, 63, 66, 69, 70
CNN	*Convolutional Neural Network*. 13, 25, 27, 29, 62
CRNN	*Convolutional Recurrent Neural Network*. 13, 29
DNN	*Deep Neural Network*. 13, 37
FFN	*Feed-Forwards Neural Networks*. 13, 26, 27
FN	*False Negative*. 13, 31
FP	*False Positive*. 13, 31
FPR	*False Positive Rate*. 13, 32
HL	*Hamming Loss*. 13, 32, 63, 66
ISMIR	*International Society for Music Information Retrieval*. 13
LTCI	*Laboratoire de Traitement et Communication de l'Information*. 2, 13
MIR	*Music Information Retrieval*. 8, 13, 16, 17, 24, 25, 29, 56, 103, 104
MLML	*Multi-Label classification with Missing Labels*. 13, 67
MLP	*MultiLayer Perceptron*. 13, 25
NLP	*Natural Language Processing*. 13, 57
RNN	*Recurrent Neural Network*. 13, 29
ROC	*Receiver Operating Characteristic*. 13, 70
STFT	*Short Time Fourier Transform*. 13, 28
TF-IDF	*Term Frequency–Inverse Document Frequency*. 13, 57, 65, 66
TN	*True Negative*. 13, 31, 63, 66
TP	*True Positive*. 13, 31
TPR	*True Positive Rate*. 13, 32

Part I

Overview

Chapter 1

Introduction

1.1 General Context

Music is a fundamental part of a universal human culture dating back thousands of years. Throughout this time, music has had different functions related to various aspects of life, from religious performances to entertainment purposes. For the majority of this time, music could only be performed live, i.e. before the invention of recorded music. This limitation had its effect on how music was both listened to and performed, as it was mostly a social event. Since then however, advances in technology have led to changes in how music is performed and consumed nowadays. In particular, the invention of recording and streaming services have permanently changed how we listen to music [Web02].

Music-on-demand streaming services allow a user to instantly listen to any recorded music available in their catalogues. With millions of available tracks, music exploration has entered a new era. However, this vast amount of possibilities required new methods to help users explore and retrieve music they would like. Similar to most online catalogues, music streaming services started developing recommendation algorithms to this end. Those recommenders have also resulted in a change in the way people consume music. Users have conveyed that a good recommender is one of the main reasons for choosing a specific service [PG13].

Moreover, the continuous availability of music has resulted in music being increasingly consumed as a background activity. People can now listen to music no matter where or when. Hence, users have developed different patterns in listening to their music. They have also developed different preferences for these varying situations. It has become important to recommend not only the right items but also at the right moment and for the right situation. It has been well documented that the context, i.e. the listening situation of the users, strongly influences their listening preferences [NH96c]. Consequently, there has been an increasing attention towards developing context-aware recommender systems [AT11].

One of the most desired properties of recommender systems is transparency. State-of-the-art contextual music recommender systems use various techniques to embed the contextual information in the recommendation process, e.g. sequence-based models that predict the tracks in the next session using available contextual information [HHM$^+$20]. However, most of these approaches lack interpretability and serve as a hit-or-miss with no room for user involvement. Interpretability is increasingly becoming a priority for both users and services [ADNDS$^+$20, VDH$^+$18, ABB14]. This is important because it allows us to build trust and understanding of the provided service.

One approach to achieve interpretability is by using human-readable descriptors similar to the ones people use to describe music to each other. These descriptors are normally added by the users or the service to help organize large catalogues. However, with millions of tracks available, manual annotation of music is a challenging task that is also prone to noise. On the other hand, discovering the proper descriptors by analyzing the audio content of a track is an alternative solution. Music Information Retrieval (MIR) is an interdisciplinary field that addresses this type of problems. MIR relies on music theory, computer science, signal processing and machine learning in order to extract or generate meaningful information related to music.

One of the frequent goals in MIR is bridging the so-called “semantic gap” [CHS06]. The semantic gap refers to the aforementioned missing link between the content of the music and a set of human semantic descriptors. One of the least explored descriptors is the intended listening situation. Few of the previous approaches focused on studying how the audio content relates to these situational influences, and to a lesser extent making use of the audio content in providing the contextual recommendations. Adding personalized descriptors, i.e. tags, to the tracks describing the intended listening situation of the track by a given user would significantly improve music exploration, catalogues organization, and recommendation by making it contextual and explainable.

1.2 Relation to Previous Work

Previously, the main focus of music recommendation systems was finding music that would suit the user’s preferences regardless of the user’s different contexts. Despite the constantly improving performance of these recommendation systems in finding suitable tracks, recommending suitable music in the wrong context is not considered a good recommendation. Hence, there has been an increasing interest in context-aware recommendation systems recently [HAIS+17b, KR12].

Context-aware recommendation systems, along with classical recommendation systems, are common in many services other than music, e.g. in online shopping [PPPD01] or movie streaming [SLC+17]. However, it is specifically more pressing in the case of music streaming due to the dynamic nature of listening to music [SZC+18]. Music tracks often have a duration of few minutes while users would listen to music for hours during the day. This leads to a constant need for providing recommendations to the user. Additionally, the user context, e.g. activity or location, could change frequently while listening to music, which leads to changes in the user’s preference and consequently needs different types of recommendations. Hence, an understanding of the different types of user contexts and their effect on the music style and listening preferences is important for improving the current state of recommendation systems.

While it is established that contexts are important, there has not been a comprehensive study on how the different types of context relate to the actual content of the music. Contexts are often described as anything affecting the user’s interaction with the service [KR12, Dey00], which is a very broad definition. The vague nature of this definition makes it particularly challenging to address interpretability in context-aware recommenders. As we are increasingly concerned with interpretability, it is important to have a clear notion of this context that can be communicated with the user. This has rarely been the focus of previous work that focused on integrating all available contextual information in recommendations with little regard to how the final outcome could be interpreted.

Furthermore, we consider studying the relationship between audio content and contexts to be essential in understanding the influence of different contexts on user preferences. This has been largely missing in previous work that often integrated the contextual influence in a recommendation setup directly [KR12], with the exception of emotion-related influence. Finally, the extent to which each user's personal preferences influence these temporary contextual preferences is also important to understand, in order to provide a personalized experience.

Given the aforementioned mentioned gaps in previous studies, we dedicated the work in this dissertation to addressing these problems. We paid extra attention to setting a ground base for future work in terms of: thoroughly defining our setup, collecting large datasets that are made publicly available, finding suitable evaluation protocols, and sharing the sourcecode for our developed models and evaluation experiments.

In this dissertation, we address this semantic gap between the audio content and the intended use of music influenced by the listening situation. We study the potential of using the audio content primarily to disambiguate the listening situations, providing a pathway for recommendations based on the situation. We explore the usability of one of the powerful tools in MIR; the music auto-taggers, in learning the relationship between the content and the situation. This alternative approach of treating the context as a tag describing a listening situation provides a strong case for interpretable recommendations, since these tags can be easily communicated with and understood by the user.

This alternative approach, however, faces a challenge that is uncommon in the traditional case of auto-tagging, which is user dependency. Unlike tags that describe attributes totally dependent on the content of the music, the intended listening situation is both content and user dependent. Hence, further investigation of how this user dependency can be integrated in the traditional auto-tagging setup is essential in our studies.

Finally, in order to employ this personalized auto-tagging approach in an actual real-world automated recommendation process, it is also important to be able to predict when a specific listening situation is being experienced. This multi-faceted problem is approached in this dissertation in a highly data-driven manner, relying on data retrieved and evaluated from real use-cases in Deezer [1], a popular online music streaming service.

1.3 Challenges

By reviewing the previous work on context-aware music recommendations, we identified the following challenges:

User-side interpretability: Most previous work focused on providing reliable recommendations in terms of performance and evaluation metrics. Some work that is domain-specific, e.g. location-aware systems [SS14, KRS13], could provide recommendations interpretable by the user. However, they are narrow in their range of applicability and do not fit the industrial needs. Interpretable recommendations are increasingly becoming a top priority for both the service and the users. Hence, we seek to approach the problem in a manner that is aims at providing more interpretable recommendations.

The semantic gap: Music is highly complex and is often challenging to be analyzed and described in human readable terms. This missing link between the content of the music and a set of human semantic descriptors is referred to as the semantic gap. There are several way to bridge this gap and extract useful attributes that can describe the music [CHS06]. One very common way, which is massively used when searching for music, is the intended use-case, which can also refer to the listening situation. By observing the

---

[1] `https://www.deezer.com/`

user created playlists in online music streaming services, we find thousands of them made for a specific situation, e.g. an activity such as running. Hence, to understand, extract, and link such semantic descriptors automatically to the content of the music tracks is a solution for the semantic gap [Smi07].

Personalization: While contextual influence has been well documented in previous work [NH96a], the personal differences between users within a given context, are far less explored even though they were observed [GSS18]. That is to say, in the same listening situation, two users could have two completely different preferences of the music they listen to. Hence, there is need to discover and integrate this personal bias when disambiguating the listening situation.

Not only what, but also when? As mentioned earlier, the listening situation has a strong influence on *what* a user would prefer to play. While the first half of this work focuses on unraveling this relationship through the audio content, in order to be fully usable we have to be able to detect *when* a listening situation is being experienced. This is particularly challenging given the wide range of potential situations and the little information we have about the user of a service at a given time.

1.4 Research Objective

1.4.1 Main Research Objective

Our main research objective is to reduce the semantic gap by adding situational tags to the tracks. We aim at achieving this by developing an audio-based approach to disambiguate the potential listening situations of a given track using the audio content. Furthermore, we aim at developing a system that is personalized, i.e. predicts the listening situation of a given track according to a given user. Achieving this objective would allow us to communicate with the user the predicted listening situation to interpret the recommended tracks.

To achieve this objective, we have to first achieve a preliminary objective, which is identifying the relevant situations that influence the music listening preferences. We seek to do this in a way that allow for future research to build on top of our defined situations and provided datasets. This will help tackle the challenge of non-uniform definitions of relevant situations in the previous work.

1.4.2 Research Questions

The research conducted in this dissertation aims at answering four main research questions. These questions are meant to first formalize the problem and lay the foundation for future work addressing the same topic. Subsequently, they aim at developing a solution that is adaptable to the industry, e.g. Deezer in our case, for providing explainable and timely contextual recommendations. The research questions addressed in this dissertation are:

How to identify relevant situations to music listening? Context-awareness can be regarded as a large umbrella describing various external influencing factors on the users. As we prioritize interpretable and semantically meaningful approaches to the problem, it is essential to identify the most relevant, i.e. influential, situations in terms of listening preferences. To answer this question, we need a proxy to discover relations between the users' listening situations, their corresponding preferences, and the frequency of such situations.

What is the relationship between the audio content and the listening situations? Can auto-taggers be employed for this task? Given the influence of the listening situations on the users' preferences, we investigate how the audio content can reflect this influence. We want to study the potential of using the audio content directly in predicting the listening situation for a given track. This would allow us to add semantic tags that describe the tracks in terms of their intended use case. To this end, we study the usability of one specific tool, audio auto-taggers, which have proven to be useful in identifying several semantic descriptors of a tracks, such as the genre, mood, or instrumentation. In order to investigate this, we need to identify a proper dataset collection procedure that can link the listened tracks to the listening situations, identified from answering the previous research question.

How to integrate user-dependency in the auto-tagging setup? Contrary to most used tags, which describe attributes purely related to the content of music tracks, the case of tagging with intended listening situations is likely user-dependent as much as it is content-dependent. Hence, we aim at investigating this user dependency, and how to adapt current music auto-taggers that only analyze the audio content, to also consider the user in question. Similar to the problem of linking the audio content to the listening situation, we need a proper approach to develop a dataset collection pipeline that can identify/describe the different listeners, and associate them with the listening situation and listened tracks.

To what extent can we infer the users' listening situations automatically? While answering the previous questions can be insightful for understanding how people listen to music and their potential intentions regarding a track, it can be efficiently used in real-world recommendations only if the listening situation can be inferred in real-time. Hence, the logical next step is to study the potential of using the available data in music streaming services in order to predict the listening situation. Here, we need to be careful when answering this question, in terms of the complexity of the model, which determines the real-time applicability, and the used data, which should be limited to only basic data available during a streaming session.

Finally, after investigating all the previous research questions, the last stage would be to study the applicability of this approach in an actual recommendation setup. Hence, we study the effect of employing both the predicted situational tags using our proposed model, along with the predicted situation timing, on the quality of the recommendations of a state-of-the-art recommendation algorithm. These tags allow us to filter the potential list of recommended tracks to only include tracks that are associated with the current listening situation, which in turn is predicted using the user data available to the service.

1.5 Contributions

Identifying relevant situations to music listening: In this thesis, we present an extensive analysis of the previous work done on identifying the relevant listening situations in music. We use these previous studies to collect a large set of potential situations, which we extended through semantic similarity and frequently associated keywords on social media. We identify 96 keywords describing several situations that are categorized into: *activity, time, location, and mood.* Furthermore, we identified their importance through their frequency as they appeared in playlist titles created by the Deezer users. These keywords lay the foundation for all our future experiments, as they describe the tags we aim at using to describe listening situations. This procedure allowed us to collect 3 large datasets for each experiment which were all made public for future research. This type of situational datasets on this scale is the first of its kind made public.

The relationship between the audio content and the listening situations (potential of auto-taggers): Through the keywords resulted from the first study, we developed a semi-automated approach to link the listened tracks and the listening situation using playlist titles as a proxy, backed by rigorous filtering. Through this approach, we were able to collect the first dataset of music tracks labelled with their situational use. We conducted our pilot study of exploiting music auto-taggers to identify potential listening situations using the audio content to set a benchmark for this task. Finally, our analysis of the results has further strengthened our initial hypothesis that certain situations are highly user-dependent.

During this study, we were faced with a common challenge in the case of multi-label datasets: missing labels. Given the procedure used in collecting the dataset, we identified a method to estimate our confidence in the collected labels. We further employed this confidence in developing a confidence-based weighted loss to account for the missing labels. Our studies validated the usability of such approach in learning from a dataset with missing labels. The proposed loss is particularly useful in cases of a predefined architecture or fine-tuning a pretrained model, which has been missing from previous approaches addressing missing labels.

User dependency and listening preferences: We proceeded by extending the music-auto-taggers to a user-aware model to make personalized predictions. Previous auto-taggers have all been solely audio-dependent, a drawback we overcome in order to adapt for our problem. We relied on the users' listening history in order to model their global preferences. Our user-centered evaluations showed that including the user in the loop, represented through their history, significantly improves the performance of predicting the situations. This user-aware music auto-tagger enabled us to tag a given track through the audio content with potential situational use, according to a given user through his/her listening history.

Inferring the listening situation automatically Finally, to successfully employ this tool for a recommendation task, we needed a different tool to predict the potential current situations of a given user. To this end, we developed a model to predict the situation given the data transmitted from the user's device to the service, and the demographic information of the given user. Our evaluations show that the models can successfully learn to discriminate the potential situations and rank them accordingly.

By combining the two model; the auto-tagger and situation predictor, we developed a framework to filter the set of potential tracks in real-time, based on the predicted situation, before deploying traditional recommendation algorithms and proposing the results to the user. This framework provides an alternative pathway to recommending situational sessions, aside from the primary sequential recommendation system deployed by the service, which is interpretable through the tags. Our evaluation showed that pre-filtering those situational sessions with the corresponding tags significantly improved the performance of the traditional recommendation algorithm when compared in situational sessions.

1.6 Thesis Structure

This thesis is split in three main parts. The first part I, which is composed of the current chapter 1, gives an overview of the general context of the problem in question and set the objectives and contributions presented in this Thesis. The second part II aims at introducing the required preliminaries for understanding our approach to the problem and an overview of previous work in the same domain. That part includes first a chapter 2 describing music auto-taggers, their evolution, applications, potential use in our case, and the common evaluation methods used. Afterwards, as our final goal is facilitating music recommendations, we give an overview of the common approaches in developing a recommender system and the different evaluation methodologies in Chapter 3. Finally, as we are specifically interested in context-awareness, the last chapter 4 in this part introduces the previous approaches and attempts in defining and employing contextual information in the recommendation process. Throughout all those chapters, we give additional attention to the case of music and its specific requirements.

The third part III of this thesis encompasses our proposed work and contributions addressing the research problem. We start in Chapter 5 by defining a pipeline to link the listening situation and the audio content by using playlist titles as proxy. Through this approach, we investigate the usability of music auto-taggers in learning to automatically tag the tracks with the context using the content. Afterwards, and given our findings in the previous chapter, the next chapter 6 focuses on the usability of the user information in developing a user-aware music auto-taggers that is capable of giving personalized tags. In the last chapter 7, we investigate the potential of using the user-aware auto-taggers in a real-world recommendation scenario, by developing a system to predict the listening situation while using the service. Finally in Chapter 8, we conclude our work through a detailed discussion about the insights we gained from our studies, along with a detailed section about the future work that can be further conducted given our findings.

1.7 Publications and Talks

In this section, we present publications and seminars that occurred during the PhD thesis. For all publications, all code and redaction have been made by the PhD student.

Publications

Ibrahim, K. M., Royo-Letelier, J., Epure, E. V., Peeters, G., Richard, G. Audio-based auto-tagging with contextual tags for music. *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020.

Ibrahim, K. M., Epure, E. V., Peeters, G., Richard, G. "Confidence-based Weighted Loss for Multi-label Classification with Missing Labels." *Proceedings of the 2020 International Conference on Multimedia Retrieval*, Dublin, Ireland, 2020.

Ibrahim, K. M., Epure, E. V., Peeters, G., Richard, G. "Should we consider the users in contextual music auto-tagging models?" *Proceedings of the International Society for Music Information Retrieval Conference*, Montreal, Canada, 2020.

Ibrahim, K. M., Epure, E. V., Peeters, G., Richard, G. "Audio auto-tagging as Proxy for Contextual Music Recommendation" *Manuscript in preparation.*

Talks and Seminars

Karim M. Ibrahim, *Finding Contextual Information from Playlist Titles*, Seminar at the Audio Data Analysis and Signal Processing, 2019.

Karim M. Ibrahim, *Audio Content and Context in Music Recommendation*, Data and Research Seminar, Deezer, 2019.

Karim M. Ibrahim, *Context Auto-Tagging for Music: Exploiting Content, Context and User Preferences for Music Recommendation*, MIP-Frontiers Summer School at UPF, Barcelona, 2019.

Karim M. Ibrahim, *Audio Content and Context in Music Recommendation*, Seminar at the Audio Data Analysis and Signal Processing, 2019.

Karim M. Ibrahim, *User-Aware Music Auto-Tagging*, Seminar at the Audio Data Analysis and Signal Processing, 2020.

Karim M. Ibrahim, *Recommending Situational Music*, Data and Research Seminar, Deezer, 2020.

Karim M. Ibrahim and Philip Tovstogan, *Cross-Dataset Evaluation of Auto-Tagging Models*, The first MIP-Frontiers workshop, Online, 2020.

Karim M. Ibrahim, *Situational Music Playlists Generation*, Data and Research Seminar, Deezer, 2021.

Karim M. Ibrahim, *Situational Music Playlists Generation*, Seminar at the Audio Data Analysis and Signal Processing, 2021.

Karim M. Ibrahim, *Audio Auto-tagging as Proxy for Contextual Music Recommendation*, MIP-Frontiers Final Workshop, Online, 2021.

Part II

Preliminaries: Music Auto-taggers, Recommender Systems, and Context-awareness

Chapter 2

Music Auto-taggers

One of the most useful tools in the MIR domain is the auto-taggers. The objective of an auto-tagger, given the content of a piece of media, is to output "tags" that describe that item's properties [BMEML08]. Those tags are remarkably useful in organizing a catalogue of items, browsing through it, facilitating the search functionality, and finally as a tool to support recommender systems. Tags are high-level descriptors that reflect certain attributes of an item. They are widely used all over the web to describe books, movies, or music. Normally, tags were added by the actual users of the service through a process called crowd-sourcing [GSMDS12]. Subsequently, these manually annotated items provided large datasets [ÇM+17] that allowed training a predictive system to automatically and reliably annotate any given item using its content. In the following, we will give an overview on the case of music tags and auto-taggers. An extensive study and summary of the progress in this domain can be attributed to the recent work by Pons on music auto-taggers [PP+19].

2.1 Music Tags

Music became instantly available on a very large scale to millions of people online. This has resulted in the creation of adaptive ways [TBL08] to retrieve and search for new music based on different descriptors other than just the artist, album, or track name. For example, a user may need to search for something as broad as "jazzy night music" without knowing a specific artist or track. Perhaps they have a sudden urge to find "melodic metal music with female vocalist", without knowing where to start. Hence, we can already see how tags can be a shortcut to find such music with little information on any specific track. Music auto-taggers can then be broadly defined as *any system that aim at predicting musically relevant tags from the audio signal* [BMEML08].

This important task of automatically annotating music tracks with high-level descriptors, i.e. tags, became one of the primary tasks in music information retrieval. Such tools can be useful in numerous applications, because they allow the users to communicate with the service in broad terms in order to search for and find suitable music with a wide variety of specific qualities [LYC11, Lam08]. Additionally, they can be used in providing recommendation by observing the recent trends in an active session, and subsequently provide music with similar descriptors, i.e. tags.

Those tags have been covering a very long list of possible categories, including the instruments, genre, mood, time period, language, or specific musical characteristics. Specific examples of those tags can be meter tags (e.g., triple-meter, cut-time), rhythmic tags (e.g., swing, syncopation), harmonic tags (e.g., major, minor), mood tags (e.g., angry, sad), vocal tags (e.g., male, female, vocal grittiness), instrumentation tags (e.g., piano, guitar), sonority tags (e.g., live, acoustic), or genre tags (e.g., jazz, rock, disco).

Music auto-tagging can be described as a multi-class binary classification task. This task can be multi-label multi-class problem, i.e. the model predicts multiple labels from a set of non-exclusive classes, e.g. instrumentation tags. Alternatively, it can be a single-label multi-class task that predicts the correct tag out of a set of exclusive classes, e.g. high-level exclusive genres. Typically, these tags are then used in a variety of subsequent problems such as music recommendation or retrieval. Hence, this task has received extensive attention from the research community working on MIR [KLN18, PPNP+18, WCS19, CFS16a, CFSC17, WCS19, WCNS20].

Traditionally, the proposed systems relied on a set of carefully hand-crafted features that were deemed suitable for the target tags [PKA14, BMEML08]. However, most recent approaches started adopting deep neural networks with various architectures, which proved to be the most successful approach for this problem so far. Deep neural network have the advantage of automatically discovering and extracting the relevant features from the raw input in a data-driven manner [WCNS20]. In the following, we will give an overview of the progress in deep neural networks related to the task of auto-tagging, with focus on the parts that are later used in our work.

2.2 Neural Networks Applied to Music

Neural networks describe a large family of algorithms that share one common target: minimizing an objective function that approximates a specific task through optimization methods [SCZZ19, BCN18, LBBH98]. Most neural networks are composed of basic building blocks stacked together, which are optimized to learn and extract the relevant features for the given task. Hence, we will dive into some of those building blocks which are commonly used in developing music auto-taggers and are used in our approaches: the multilayer perceptron (MLP) [GD98], and the convolutional neural networks (CNNs) [ON15].

Formally, music auto-tagging neural networks can be described a function that maps an audio input x to a set of tags $\hat{\mathbf{y}}$ such that $\hat{\mathbf{y}} = f(x)$. This is achieved through optimizing a set of trainable parameters θ, and can be further described as $\hat{\mathbf{y}} = f(x; \theta)$. In the following, we will go through how the building blocks can be described in terms of these trainable parameters, the intuition behind it, and how they are optimized for a specific task.

2.2.1 Training a Neural Network

Training refers to the process of searching for the best set of parameters θ that approximates a specific objective function. The most basic training algorithm for neural networks is stochastic gradient descent (SGD). SGD updates the model parameters' θ as follows [RM51]:

$$\theta_{i+1} = \theta_i - \mu_i \nabla \mathcal{L}(\theta_i) \tag{2.1}$$

Where i refers to an iteration index, as the algorithm is applied iteratively till convergence. However, SGD does not guarantee to reach a global minimum when optimizing non-convex error functions, which is the case when training a deep neural network. Nonetheless, SGD proved to be working well in practice. The algorithm is deployed on a set of parameters θ that are, most commonly, initialized randomly. Afterwards, a backpropagation algorithm [RHW86] computes the gradient of the objective function $\mathcal{L}$ with respect to each parameter to find the update direction $\nabla\mathcal{L}(\theta_i)$. The negative sign indicates that the update aims at minimizing the objective function, i.e. minimizes the error in predictions. μ_i, which is known as the learning rate, controls the update rate of the parameters at each iteration. Finally, in the case of supervised learning, the objective function $\mathcal{L}(y, \hat{y})$ is a function that estimates the error between the predictions $\hat{\mathbf{y}}$ and the groundtruth labels $\mathbf{y}$. Hence, minimizing this error by searching the optimal set of parameters is what is referred to as training a predictive algorithm.

To put this into perspective, we will give an example of training a music auto-tagger model. Given a set of tracks $\mathbf{x}$ and their corresponding groundtruth tags $\mathbf{Y}$. The tracks are batched at each iteration, i.e. a random subset of the tracks are forwarded into the predictive model, which is the “stochastic” part in SGD. At each iteration i, the model predicts the corresponding tags for each input track $\hat{\mathbf{y}}$ using the randomly initialized parameters such that $\hat{\mathbf{y}} = f(x; \theta_i)$. The error between the predictions and the groundtruth is then computed $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$, which in turn are averaged and used to compute the gradient with respect to the model parameters $\nabla\mathcal{L}(\theta_i)$. Finally, the parameters can be updated after each batch using the equation described in 2.1.

It is important to emphasize that SGD is not the only optimization algorithm used for training a neural network. Several other algorithms were proposed, which tries to overcome several of the drawbacks of basic SGD. To name a few: the momentum variant [RHW86], adam [KB14], or adadelta [Zei12]. In the following, we will be diving into some of the most commonly used building blocks of neural networks.

2.2.2 Feed-Forward Neural Networks

The earliest proposed building blocks for artificial neural networks is the multi-layer perceptron. A single layer is defined as $\hat{\mathbf{y}} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$ [Ros58]. The input $\mathbf{x}$ is linearly transformed with the trainable matrix $\mathbf{W}$ and shifted with the trainable bias $\mathbf{b}$. Finally, an activation function σ is applied to produce the output of the layer $\hat{\mathbf{y}}$, which is often a non-linear transformation.

However, feed-forward neural networks (FFN) are composed of multiple layers, called the hidden layers, stacked one after the other, i.e. “deep” neural networks. In case of a network with three layers the predictions follow this sequence:

$$\hat{y} = \sigma_{(2)}(W_{(2)}h_{(2)} + b_{(2)}) \tag{2.2}$$

$$h_{(2)} = \sigma_{(1)}(W_{(1)}h_{(1)} + b_{(1)}) \tag{2.3}$$

$$h_{(1)} = \sigma_{(0)}(W_{(0)}x + b_{(0)}) \tag{2.4}$$

Here, the subscript refers to the layer number, where zero is the input layer. The final predicted output is $\hat{\mathbf{y}} \in \mathbb{R}^{d_{output}}$ and the input is $\mathbf{x} \in \mathbb{R}^{d_{input}}$. The trainable parameters are the weight matrices $W_{(l)} \in \mathbb{R}^{d_{(l)} \times d_{(l-1)}}$, and the bias vector $b_{(l)} \in \mathbb{R}^{d_{(l)}}$, where l stands for the index of the layer. Hence, each layer adds new trainable parameters to the network. The intermediate output of those layers is referred to as $h_{(l)}$. The dimension of each of those outputs is $d_{(l)}$, and is often referred to as the number of nodes. $\sigma_{(l)}$ can be any activation function for each layer independently. Some of the most commonly used activation functions are the Sigmoid, tanh, and the rectified linear unit (ReLU) [Aga18].

We find that FFNs predict an output by computing a weighted average considering all its input, note that $W_{(l)}$ interacts with the whole input signal, either x or $h_{(l)}$. As an example, let's say we aim to predict whether an audio contains music or speech. In this case, and assuming that our latent $h_{(2)}$ representation contains relevant features like timbre or loudness, we want to weight with $W_{(2)}$ the relevance of these $h_{(2)}$ features to predict ($\hat{y}$) whether our input contains music or speech. Note that the same rationale also applies to the lower layers of the model, where this weighted average would help defining more discriminative features. Hence, training a neural network refers to both finding those relevant features to extract, and learning the appropriate weighting to the task in question.

2.2.3 Convolutional Neural Networks

Similar to feed-forward neural networks, convolutional networks use a cascade of transformational layers to process the input data and predict an output. However, unlike perceptron layers that use linear matrix multiplication, CNNs use the convolutional operators [LBD+89]. The 1D convolution operation can be defined as

$$s[n] * w[n] = \sum_{m=-\infty}^{\infty} s[m] \cdot w[n-m] \tag{2.5}$$

While the more commonly used 2D convolution is defined as:

$$s[x,y] * w[x,y] = \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} s[m_1,m_2] \cdot w[x-m_1, y-m_2] \tag{2.6}$$

Where $w[n]$ or $w[x,y]$ shifts along the input signal $s[n]$ or $s[x,y]$ to be multiplied and aggregated.

Hence, convolutional networks replace the basic perceptron layer with layers that perform this convolutional operation such that:

$$h^{(k)}_{(l)} = \sigma_{(l)}(W^{(k)}_{(l)} * h_{(l-1)} + b_{(l)}) \tag{2.7}$$

And in case of applying the operation in the input layer, then:

$$h^{(k)}_{(1)} = \sigma_{(0)}(W^{(k)}_{(0)} * x + b_{(0)}) \tag{2.8}$$

Note, here the superscript (k) refers to the k^{th} filter, since convolutional layers often apply multiple trainable filters $W^{(k)}_{(l)}$ simultaneously at each layer. This allows each of the filters to extract different relevant features locally. This locality is what makes CNNs so powerful in extracting relevant features, compared to FFNs which process the whole input altogether. For example, a 2D input $x \in \mathbb{R}^{T \times F}$ and a filter $W^{(k)}_{(0)} \in \mathbb{R}^{i \times j}$ allows to extract local features of $i \times j$, while being shifted along to cover the whole input. Note that a CNN filter can be transformed into a FF layer by setting $i = T$ and $j = F$, which will then process the whole input at once.

Originally, CNNs were largely explored and used in processing 2D images [Neb98], as they allow for searching for, extracting, and preserving the local features that are needed in classification or detection tasks. However, this powerful tool proved to be useful in a multitude of domains, including processing audio and music signals [DBS11]. While CNNs can be applied to a 1D waveform, they are often applied to a 2D representation of

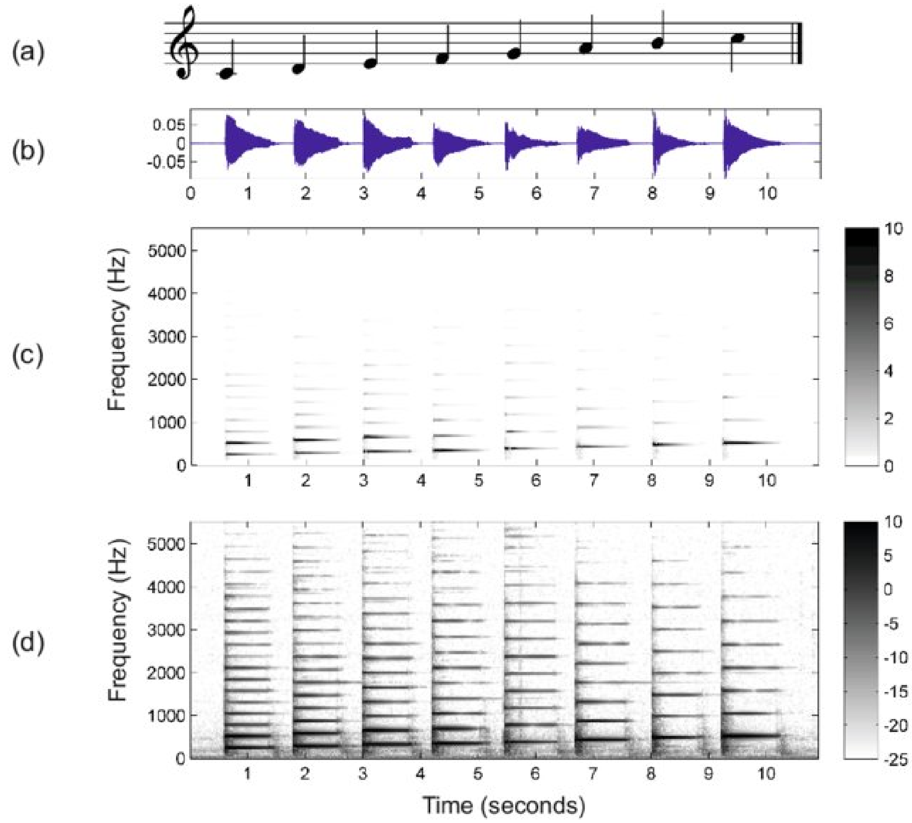


Figure 2.1 – Waveform and spectrogram of a music recording of a C-major scale played on a piano. (a) The recording's underlying musical score. (b) Waveform. (c) Spectrogram. (d) Spectrogram with the magnitudes given in dB. Source: Meinard Müller, 2015 [Mül15].

the waveform in the time-frequency domain [Wys17]. The intuition would be to learn how to process the variation across frequencies and time locally across the input signal and extract the relevant features for the given task. Hence, in the following we will elaborate on processing the 2D time-frequency spectrograms of audio signals, which will be the basis of the models used later in this thesis.

2.2.4 Spectrogram-based Convolutional Neural Networks

A spectrogram of an audio signal is a representation of the spectrum of frequencies as it varies across time. Spectrograms can be derived through a number of transformations, most commonly through applying the Short-Time Fourier Transformation (STFT) on the raw audio signal. Spectrograms are a very useful representation as they capture both the variations across each frequency and through time simultaneously. They have been widely used in a variety of applications for music processing such as auto-tagging [PPNP+18], source separation [JHM+17] or transcription [SSH17]. A visualization of a spectrogram of a musical scale can be found in Figure 2.1. It is instantly intuitive to observe the variation of the fundamental frequencies and their harmonics across the time as the scale ascends.

An important stage of employing spectrograms into a neural network is preprocessing and normalization. One of the most common preprocessing steps is converting the STFT representation using a mel-scale. Mel-scales transform the linearly separated frequencies into a non-linear scale such that it amplifies the perceptually relevant bands in comparison to those less relevant to the human ear [SVN37]. In other words, the Mel-scale is constructed such that sounds of equal distance from each other on the Mel scale, also "sound" to humans as they are equal in distance from one another. Hence, we can say it maps the time-frequency representation to one that is similar to the one perceived by the human ear. Afterwards, the melspectrograms are normalized to a zero-mean and unit variance for being better processed by the neural networks [GB10, CFSC18].

Current State of Music Auto-taggers

As described earlier, the research on music information retrieval has given particular attention to developing music auto-tagger. To achieve this, several tools have been proposed that make use of different architectures and input formats. Some of the proposed approaches relied on using the audio content directly as raw signal [KLN18, PPNP+18, WCS19]. Others relied on using the previously mentioned pre-processed spectrograms, such as [CFS16a, CFSC17, WCS19, WCNS20]. Those proposed approaches have proven to be very useful to achieve easy retrieval, categorization, and overall organizing large catalogues using interpretable descriptors, i.e. the tags.

While the format of the input data were explored in these studies, the architecture of the used models were also investigated. Some of those studies used CNNs as their main building block, applied both on the 2D spectrograms [CFS16a] and the 1D waveform [KLN18]. Others applied RNNs [SWH+18], and self-attention [WCS19]. One approach tried to combine the advantages of extracting local features using CNNs, while learning the sequential structure of those features using RNNs through a front-end plus a back-end approach using CRNN [CFSC17].

By observing the cross-evaluation between those methods [KLN18], we find their performance is tightly comparable across different datasets. However, RNN-based approaches could suffer from huge computational power and are generally harder to train due to gradient vanishing/exploding problems [PMB13]. Hence, other factors in choosing a model are often the complexity and time/memory requirements.

Even though CNN-based approaches are appealing in terms of performance and complexity in comparison to other approaches, they also have some undesirable sides. CNNs in the domain of music/audio are barely interpretable, despite many attempts to explain the underlying process [MSD18, MSD17, CFS16b]. Nonetheless, CNNs are widely used in MIR to take advantage of its time-frequency invariance and robustness to distortion. However, there is active work-in-progress on developing more musically motivated models that are designed specifically to process music [PS17, PSG+17].

2.3 Multi-label and Single-label Auto-tagging

Auto-tagging can either aim at tagging an input track with a single label from a set of exclusive labels, e.g. high-level genre classification, or with multiple labels, e.g. the used instruments in the track. Both problems are largely similar except for few differences, primarily in the objective function that allow one vs. multiple correct answers. This criteria is decided based on the available training data and the target tags.

Single-label classification is achieved through applying the soft-max function function in the last layer. Soft-max is defined as:

$$\hat{y}_i = \sigma_{softmax}(h_i) = \frac{e^{h_i}}{\sum_{j=1}^{K} e^{h_j}} \tag{2.9}$$

where h_i is the output for the i^{th} label, and K is the total number of labels to be predicted. $\sigma_{softmax}$ aims at normalizing the outputs of the last layer into a probability distribution over predicted output classes, such that $\sigma_{softmax} : \mathbb{R}^{\mathbb{K}} \rightarrow [0,1]^K$. The correct label is selected by picking the one with the highest probability. Training a neural network for single-label classification is often achieved by employing the cross-entropy loss function defined as:

$$CE_{singlelabel}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{K} y_i \log(\hat{y}_i) \tag{2.10}$$

We find that the loss penalizes the specific case where the groundtruth $y_i = 1$, in which case the loss would be zero only if the predicted label $\hat{y} = 1$ is also equal to 1. Single-label classification is generally regarded as a simpler problem both while collecting the dataset and while training a predictive model.

On the other hand, multi-label classification is achieved by applying the sigmoid function in the output layer. The sigmoid function is defined as:

$$\hat{y}_i = \sigma_{sigmoid}(h_i) = \frac{1}{1 + e^{h_i}} \tag{2.11}$$

This results in an independent prediction probability for each class $\in [0, 1]$. The final predictions are then derived by applying a threshold, which is often 0.5, but can be also optimized based on the performance of the trained model on a validation subset. Multi-label classification models are trained using the sum of the cross-entropy loss function applied to each class

$$CE_{multilabel}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{K} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \tag{2.12}$$

In the multi-label setup, each class is penalized independently if the predicted value does not match the groundtruth. Compared to single-label, multi-label classification is significantly harder, specially in collecting and annotating a scalable dataset [DRK+14]. In cases where multiple labels are associated with each instance, a comprehensive annotation is required to ensure a consistent and complete list of labels for every sample. Hence, multi-label datasets often suffer from the problem of missing labels, a problem that we will be encountering in our setup as well. It is shown in [ZBH+16] that learning with corrupted labels can lead to very poor generalization performances.

Finally, in the following, we will be elaborating on the evaluation metrics used in those classification problems, both in the cases of single- and multi-labels.

2.3.1 Evaluation Metrics

Evaluating a predictive model has been well established in previous work with a number of standard metrics [HS15]. These metrics are used to evaluate different aspects of the model's performance. The goal of evaluation is often to assess both the quality of the predictions and the model's generalization to unseen data. This evaluation is done on two stage, the training stage and the testing one.

During the training stage, the evaluation metrics are used to optimize the predictive model and fine-tune its parameters. Hence, this stage helps in finding the optimal model which is expected to give the best performance in future evaluation during testing. The testing stage aims at evaluating the actual effectiveness of the model when deployed on unseen

Table 2.1 – Confusion Matrix for Binary Classification and the Corresponding Notion for Each Case

	Actual positive	Actual negative
Predicted positive	True positive (TP)	False Positive (FP)
Predicted negative	False negative (FN)	True negative (TN)

data. A key difference between the stages is that the testing stage cannot be used in fine-tuning the model or finding the best parameters.

To further explain the evaluation metrics, we have to define some key concepts that are employed in the evaluation. For a simple binary classification setup, each predicted label can have one of four states based its groundtruth: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). The meaning of each of these states can be understood by looking at the confusion matrix for binary classification in Table 2.1.

The most widely used metric for evaluation is the accuracy. Through accuracy the quality of produced solution is evaluated based on percentage of correct predictions over total instances. The advantages of accuracy or error rate are, this metric is easy to compute with less complexity; applicable for multi-class and multi-label problems; easy-to-use scoring; and easy to understand. The accuracy is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.13}$$

While accuracy describes the overall performance in predicting presence or absence of a tag, it is often useful to further look into the model's ability to correctly predict the present tags. Two very common metrics used to evaluate this are the recall and precision [Bis06]. The recall measures the model's ability to correctly classify all the existing positive samples. On the other hand, the precision describes the ratio of the positively predicted instances that are indeed actual positives.

Hence, the recall is defined as:

$$Recall = \frac{TP}{TP + FN} \tag{2.14}$$

The precision is defined as:

$$Precision = \frac{TP}{TP + FP} \tag{2.15}$$

However, a model can easily achieve a recall of 1 by predicting all positives, while the precision in this case would be much lower. Hence, a metric that describes the harmonic mean of the precision and recall has been developed and is widely used to properly evaluate a model. This metric is the F_1 score and is defined as:

$$F_1 = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{2.16}$$

These metrics are quite powerful in evaluating the overall performance of the model. However, those metrics are computed through a threshold applied on the predicted probability, and does not reflect the models ability to discriminate between the different labels in terms of ranking. Hence, one of the most popular metrics used to evaluate a model's ability to discriminate between classes is the area under the receiver operating characteristic (AUC). AUC was proven theoretically and empirically to be better than accuracy in evaluating a classification model [LHZ03].

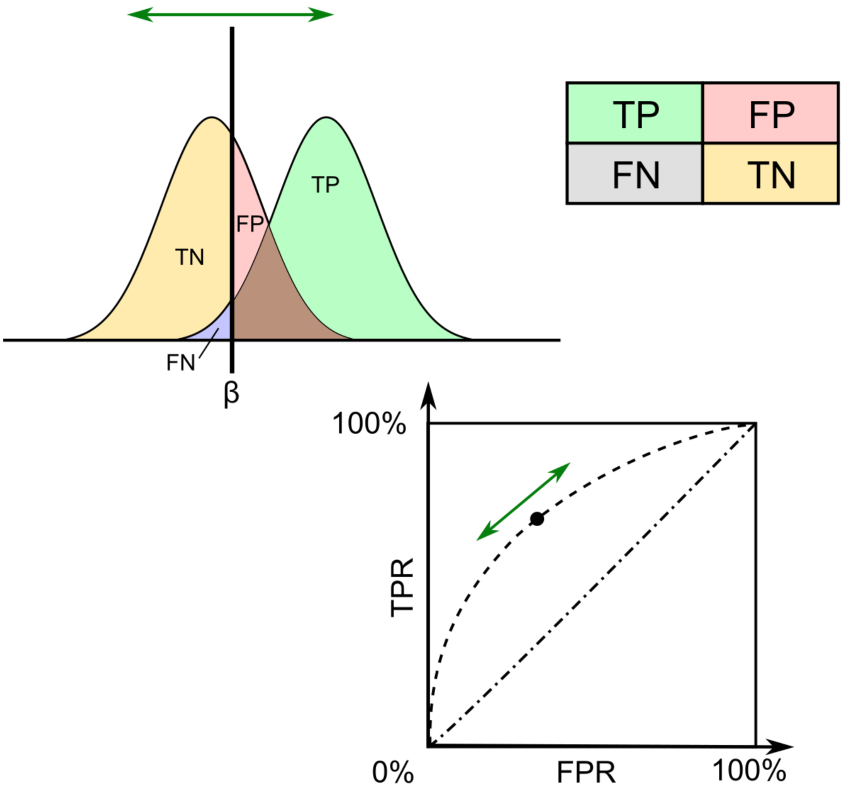


Figure 2.2 – ROC Curve plotted when threshold β is varied. Source: Sharpr 2015, distributed under the CC BY-SA 4.0. via Wikimedia Commons

AUC is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example. To compute the AUC, we first have to define two terms: the True Positive Rate $(TPR) = \frac{TP}{FN+TP}$, which is synonym for recall, and the False Positive Rate $(FPR) = \frac{FP}{TN+FP}$. FPR and TPR both have values in the range $[0, 1]$. FPR and TPR both are computed at varying threshold values, such as $\{0.00, 0.02, 0.04, \ldots, 1.00\}$, and the curve of the trade-off between the two terms is drawn. AUC is the area under the curve when plotting False Positive Rate vs True Positive Rate at different thresholds in $[0, 1]$. That is to say, a random model would have a consistent, i.e. linear, increase in both the true and false positives as the threshold increases. A well discriminative model will have a higher increase in true positives compared to false positives as the threshold increases, resulting in a higher curvature, i.e. larger area under the curve. Figure 2.2 is a visualization of the process of changing the threshold, β, and its effect on the drawn curve and the trade off between FPR and TPR.

The previous metrics are commonly used in both single- and multi-label problems. However, given the specificity of multi-label classification, other metrics have been proposed for evaluating this case. One of the most prominent metrics is the hamming loss. The hamming loss describes the ratio of the labels that are misclassified by the model, hence the lower the value the better. Hamming loss (HL) is defined as:

$$HL = \frac{1}{N} \sum_{i=1}^{K} \frac{y_i \oplus \hat{y}_i}{K} \tag{2.17}$$

where $\oplus$ represent the XOR operation, and N is the total number of samples in the dataset. These classification metrics will be repeatedly used to evaluate our models in several different evaluation scenarios.

2.4 Conclusion

In this chapter, we presented the basic building blocks often used for building a music auto-tagger using deep neural networks. While this introduction is not inclusive to all the different approaches used for developing a classification model, it has covered all the preliminary knowledge needed for further understanding the models developed in this thesis. Additionally, we have given an overview of the importance of tags, and subsequently auto-taggers, in exploring online catalogues, and specifically music catalogues. Finally, we presented the common evaluation metrics used for testing different aspects of the trained models.

Chapter 3

Recommender Systems: Methods and Evaluations

3.1 Introduction

The need for recommender systems is not due to the growth of online music primarily, but rather to the emergence of online services in general. Online recommender systems can be traced all the way back to the 1990s with the beginnings of online services [BS97, BHK98, SKR99, AEK00]. The undisputed value of personalized recommendations, versus an intimidatingly large information space such as The Web, has been deduced in the literature in the early stages of online services [BS97]. In the same paper, the authors also highlighted the early categorization of recommender systems between Collaborative filtering (CF), content-base filtering (CBF), and hybrid approaches. However, this categorization kept on growing as additional design requirements were needed to meet different services. Since then, recommender systems proved to be essential for helping users navigating the vast amounts of content available in online services. They improved the users' experience in exploring massive amounts of content online [BOHG13]. Hence, a growing attention has been given to transferring successful approaches to industrial-level applications [LGLR15, JMN$^+$16, GUH15, CAS16, AG19].

In this chapter, we give an overview of the different methodologies previously used in developing recommender systems. We elaborate on both the advantages and drawbacks of different approaches and how they manifest in the case of music recommendations. Those drawbacks highlight the motivation for interpretable task-specific approach that we study in this thesis. Additionally, we explain the common methods used for evaluating and testing the recommender systems, some of which will be later applied in our experiments to test the effectiveness of situational auto-tagging as an additional layer on top of recommender systems.

3.2 Definition

As the name entails, recommender systems aim at recommending item to the users of a service. Early previous studies [RV97, HKR00] define a recommender system as: *"a system that is able to learn the user's preferences in order to provide new relevant items that might be of interest to the user"*. Hence, a recommender system should be personalized to the users to guide them in an individual way [Bur02]. This requirement focuses on individualization and personalization in the system.

Meyer [Mey12] defines 4 essential features for a recommender system. He assumed that recommendation is related to 4 key actions. 1) “Help to decide”: by predicting the item’s rating according to the user. 2) “Help to compare”: by providing a ranked list of items personalized to the user. 3) “Help to discover”: by recommending new items that are potentially relevant to user. 4) “Help to explore”: by retrieving items that are similar to a particular item. Sharma and Mann [SM13] present a formal definition for recommender systems:

$$\forall u \in U, I_u = \arg\max_{i \in I} \mathcal{R}(u, i) \tag{3.1}$$

where U is the set of users, I is the set of items that can be recommended, and $\mathcal{R}$ is a recommendation function that computes the relevance of an item i regarding a given user u. This predicted relevance is expressed by a score or a rating, measuring the likelihood a user would find an item relevant. The relevance can also be measured using the user’s information such as the age or the gender, along with the item’s information based on the available data for the service.

3.3 Recommendation Approaches

The large body of researchers working on this problem, but in different settings and services, resulted in multiple different approaches to recommend items. The service provided, available data, frequency of recommendations, and real-time requirements, all result in different designs for each system. Consequently, various new approaches and categorizations of recommender systems were developed. Burke [Bur07] provided a general taxonomy of the recommender systems types that has long been used as a reference in this research area. He introduced three main approaches:

1. Content-based filtering approach (CBF): These systems utilize the content of the items to measure their similarity with the users preferred items. The items’ similarity is calculated using features associated with the type of media being recommended. For example, if the user has positively rated a book from a particular genre, the system can recommend this genre of books.
2. Collaborative filtering approach (CF): These system rely more on the users’ similarity rather than the item’s content. The system recommends items that other users with similar tastes liked. The users’ similarity is often measured using their rating history.
3. Hybrid approach: The system combines the two above-mentioned approaches.

Traditionally, collaborative filtering, using Matrix factorization [KBV09, HZKC16], overruled most domains by providing reliable recommendations while being more practical on large scales applied to various types of items (books, movies, etc.) [BOHG13]. Recently, following the progress in deep learning, new complex and intelligent approaches appeared in the front [Sch19]. Those approaches were shown to be better at modeling the nature of user-item interactions [DWTS16, WAB+17, BCJ+18, RCL+19, KZL19]. In the following, we will give an overview on this progress in each of those approaches.

Content-based Approaches

Content-based filtering (CBF) is an approach that focuses on the analysis of items in order to generate predictions. These approaches are domain-specific and rely heavily on the type of media to be recommended, being more challenging for certain types than others.

For example, when items such as web pages, articles or news are to be recommended, content-based filtering techniques are the most successful. In content-based filtering, recommendation is made based on the user profiles using features extracted from the content of the items the user has consumed in the past [BOHG13]. Items that are mostly related to the positively rated items are recommended to the user.

Also, if the user profile changes, CBF technique still has the potential to adjust its recommendations within a very short period of time. The major disadvantage of this technique is the need to have an in-depth knowledge and description of the features of the items in the profile. A high-level description of CBF recommender systems consists of three main components [DGLM+15]: 1) items preprocessing with a content analyzer, 2) a profile learner to generate user profile, 3) a filtering component to match the similar items. More details for these three components are provided as follows:

- Content Analyzer: this component can also be described as a feature extractor. Based on the type of data, a preprocessing step is needed to extract relevant information. The goal of this step is to prepare the data in a compact format that could be used for estimating the similarity. For example, the input can be a recorded track while the output would be a set of computed acoustic features.
- Profile Learner: this component is responsible for representing the user preferences. It often uses the features computed with the content analyzer of the items in the user's history to model it. For example, a simple representation of the user profile can be the average of the tracks' features extracted earlier.
- Filtering Component: this component makes use of the output of the two previous components to retrieve relevant items. It often uses a similarity metric, e.g. cosine similarity, to retrieve items closest to the user's modeled preferences.

Designing a CBF recommender uses knowledge from two domains which are Information Retrieval and Machine Learning. Content-based recommendation techniques are tightly related to the progress in information retrieval. For example, content-based music recommenders often make use of traditional music information retrieval techniques like content embeddings or auto-tagging, which is the primary focus of the work in this thesis.

Advantages and Disadvantages of CBF

When CBF recommenders are used, they provide several advantages [DGLM+15]:

- *User Independence:* Since CBF uses primarily the item data for finding similar items, it does not depend on the behaviour of other users. This is useful in multiple cases, e.g. in a new service with few users to derive collaborative similarity.
- *Transparency:* Interpreting and explaining why the recommended items are retrieved has always been a requirement to trust an algorithm. This is achievable with CBF systems as their extracted features can provide insight on how the similarity was estimated. Furthermore, it can help the users explore more similar items using those features.
- *New items:* One key advantage of CBF systems is their ability to recommend new items with no previous interactions. Hence, they do not suffer from the cold start problem often occurring with recommender systems. This is particularly useful in services with a long tail usage, i.e. few items are frequently rated compared to a long tail of items rarely rated.

However, content-based systems still suffer from several disadvantages [DGLM+15]:

- *Limited content analysis:* The main challenge in CBF recommenders is the complexity and domain-dependence of the content analyzers. Each domain requires specific knowledge of the items and extensive development of feature extractors. Deriving the relevant features is in itself a challenging task, with no complete model with all the relevant information to measure all the factors influencing the user's experience.
- *Over-specialization:* CBF recommenders are not the best when it comes to novelty, also called lack of serendipity problem. It is challenging for a CBF recommender to provide unexpected new items that are not similar to the items in the user's history.
- *New user:* While particularly useful in recommending new items, CBF systems fall short when it comes to new users. To understand the user preferences and build a user profile, the CBF system favours users with several past interactions. New users with no interactions provide no insight in retrieving similar items.

Content-based Filtering for Music Recommendation

Music is unsurprisingly one of the most challenging media for preprocessing, extracting features, and developing content analyzers in general. However, several approaches targeting content-based filtering have been proposed. For example, MusicSurfer [CKW05] is an early system for content based retrieval of music from large catalogues. The system was developed to retrieve tracks similar to a query tracks. This early system highlighted the limitations of using low level features for music, which is unable to capture the complex aspects of music which listeners consider. Hence, they argued for using high-level descriptors such as rhythm, tonal strength, key note, key mode, timbre, and genre, which were shown to be more successful in measuring similarity.

More recent methods rely on deriving a representation, i.e. an embeddings, of the input tracks which reflects their content similarity. Deriving those embeddings can be done based on different criteria, e.g. tracks that are in the same genre or by the same artist should be more similar. Several approaches rely on deep neural networks for extracting and measuring the similarity between tracks [DS14, WW14, VDODS13]. Even though hand-crafted features describing the music content were traditionally the most prevalent [BHBF+10], DNN-based models have deemed them obsolete, given their consistently superior performance [WW14, VDODS13].

Collaborative Filtering Approaches

Collaborative filtering (CF) describes the family of approaches that rely on the users' behavioural similarity to provide recommendations, contrary to the items' content similarity in the content-based filtering [SK09]. CF approaches are one of the most commonly used approaches for recommendations in general, not only in the case of music. These approaches rely on the users feedback, either implicit or explicit, to derive a similarity metric between the users. Hence, the recommender system can suggest new items that were liked by similar users [SM95]. Consequently, CF methods do not rely on the recommended media and do not need to analyze the content of this media. This is particularly useful in cases where the media content is rather complex, as in the case of music.

General techniques

The main objective of CF methods is predicting which items the user would like based on his/her past ratings/history. Traditionally, previous work [BHK13, RRS11] has categorized CF algorithms into two main categories: memory-based and model-based. Memory-based approaches are described as methods that make use of the entire database of users to make predictions, e.g. nearest-neighbor approaches. Model-based algorithms are described as methods that train a predictive model using available data in order to make predictions of unknown ratings.

For a formal description of CF models, let U be the set of users in the service, and I be the set of available items. The ratings matrix would be $\mathbf{R}$ of dimensions $|U| \times |I|$, where each element $r_{u,i}$ in a row u is equal to the rating of the user u to the item i, or null if not rated. Hence, the objective of CF methods is predicting those null entries. This rating can be predicted either by finding similar users (user-based CF) who rated the item, or finding similar items rated by the user (item-based CF), and then aggregating those ratings. Here we give formulas for user-based CF. Given an active user u and an item i, the predicted rating for this item is:

$$\hat{r}_{u,i} = r_u + K \sum_{v \in V} w(u,v)(r_{vi} - r_v) \tag{3.2}$$

where r_u is the average rating of user u, V is the subset of users in the database who rated the item i, $w(u,v)$ is the similarity of users u and v, K is a normalization factor such that the absolute values of the weights sum to unity. [BHK13]. Various methods have been proposed to compute this similarity score w. The two most common are Pearson correlation using Equation 3.3 [RIS+94] and Cosine distance using Equation 3.4 [SKKR00] measures:

$$w(u,v) = \frac{\sum_{j=1}^{k}(r_{uj} - r_u)(r_{vj} - r_v)}{\sqrt{\sum_{j=1}^{k}(r_{uj} - r_u)^2 \sum_{j=1}^{k}(r_{vj} - r_v)^2}} \tag{3.3}$$

$$w(u,v) = \frac{\sum_{j=1}^{k} r_{uj} r_{vj}}{\sqrt{\sum_{j=1}^{k} r_{uj}^2 \sum_{j=1}^{k} r_{vj}^2}} \tag{3.4}$$

where k is the number of items both users u and v have rated.

On the other hand, model based algorithms use a probabilistic approach to compute the expected value of the user rating given their previous ratings. In its simplest form, this probabilistic approach can be described as in Equation 3.5, where m is the maximum rating available in the service. the predicted rating of a user u for an item i is:

$$\hat{r}_{ui} = \sum_{j=0}^{m} P(r_{ui} = j | r_{uk}, k \in R_u) j \tag{3.5}$$

where R_u is the set of ratings of the user u, and $P(r_{u,i} = j | r_{uk}, k \in R_u)$ is the predicted probability that this user u will give a rating j to the item i, given their previous ratings [BHK98]. This probability has been traditionally predicted using Bayesian Networks and Clustering approaches [BHK98, SFHS07].

More recently, a family of methods, known as matrix factorization, has proven to be one of the most effective and popular techniques for recommendations [KBV09, KB15], specifically after the Netflix Prize competition in 2007 [BL+07]. The main objective of matrix factorization is deriving latent semantic representation of both the users and the items in two reduced matrices. This is often achieved by using Singular Value Decomposition (SVD).

Formally, let the $\mathbf{R}$ of dimensions $|U| \times |I|$ be the ratings matrix, matrix factorization finds f latent factors by computing two matrices, $\mathbf{P}$ (of dimension $|U| \times f$) and $\mathbf{Q}$ (of dimension $|I| \times f$), such that their product approximates the matrix $\mathbf{R}$:

$$\mathbf{R} \approx \mathbf{P} \times \mathbf{Q}^T = \hat{\mathbf{R}} \tag{3.6}$$

Each row of the users matrix $\mathbf{P}$ is a vector $\mathbf{p_u} \in \mathbb{R}^f$. This vector $\mathbf{p_u}$ is an embedded representation of the user u, and similarly the item vectors $\mathbf{q_i} \in \mathbb{R}^f$. The dot product of the user's and item's vectors then represents the predicted user's u rating item i:

$$\hat{r}_{ui} = \mathbf{p_u}\mathbf{q_i^T} \tag{3.7}$$

Hence, the objective of a matrix factorization approach is to learn those two reduced matrices $\mathbf{P}$ and $\mathbf{Q}$. This is achieved by minimizing a regularized squared error between the predicted ratings matrix and the original ratings. Several approaches have been proposed for minimizing this error, notably stochastic gradient descent [KBV09] and alternating least squares [BK07]. This embedded representation has proven to be an effective way to model the users and their similarity, which will be later employed in our proposed approaches to profile the users for personalization purposes.

Disadvantages of CF

While being widely successful, specially in real-world applications, CF methods suffer from several shortcomings [SK09]:

- *Cold start:* This describes the case of a new user or item entering the service. When very few ratings are available for a new entry in the ratings matrix, it is particularly challenging to predict the correct rating in these cases.
- *Data sparsity:* This is another common problem of CF. Services often have millions of active users and items. In this case, most of the elements in the ratings matrix are null. Hence, this often results in lower accuracy in the predictions.
- *long tail problem:* This is also know as the popularity bias, which is a very common problems in CF methods. It describes a low diversity in the recommended items. This is a result of popular items, that are often highly rated, getting recommended more frequently. Item that have few ratings, even if they are likely to be preferred by the user, are ignored and overwhelmed by the popular items.

Collaborative Filtering for Music Recommendation

CF methods have been one of the earliest approaches used for music recommendation, particularly before the progress in music information retrieval for developing content analyzers. As early as 1994, Shardanand and Maes [SM95] proposed an approach that uses the ratings of users given to a set of artists to recommend items based on the ratings from similar users. Soon afterwards, Hayes and Cunningham [HC00] proposed an early online radio able to customize the content based on the similarity between the users. The users were able to rate the played tracks, which was subsequently used for computing the similarity. Other similar approaches and services were proposed shortly afterwards [HF01].

On the commercial side, an example of one of the most popular and successful commercial internet radios is Last.fm [1], which relies heavily on collaborative filtering. Through this service, the users are able to listen to a customized radio station based on a starting point, which could be a specific artist or tracks. The high quality of the recommended items can be attributed to the large user-based and available data. However, common to all approaches using CF, the system suffers when it comes to recommending music that is not popular. Finally, CF approaches are currently among the most widely used approaches in many popular services in their actual recommendation process, including Deezer.

Hybrid Approaches

As described earlier, both CBF and CF suffer from several shortcomings. However, most of these shortcomings are only challenging in one of those two methods. Hence, hybrid approaches have been proposed to overcome these challenges by exploiting the advantages of both methods. Here, we give some examples of what has been achieved.

General techniques

An extensive overview of hybrid systems was given by Burke [Bur07]. The survey elaborates on some of the approaches to make a hybrid recommender system, which includes:

- *Weighted:* The simplest way to combine the two approaches is by producing a weighted average of the predictions from both methods. Typically, this weighting is derived from the dataset properties.
- *Switching:* As the name entails, this method switches between the two available models based on certain criteria, e.g. a content-based model is to be used for the case of a new item.
- *Mixed:* In this approach, the results from both models are presented jointly. This is particularly useful for providing diversified recommendations.
- *Feature combination:* The features derived from both methods are forwarded to a single model that utilizes both information for providing predictions.
- *Cascade:* This popular approach refines the recommendations from one method by passing it through the second model.

Finally, given that our later proposed approach makes use of both a content analyzer and a collaborative-based embeddings, we can place our work in the hybrid category. Moreover, we can categorize it as a *feature combination* technique, where the derived representations of both the users and the items are jointly used in the prediction phase.

3.4 Recommender Systems Evaluation

The majority of our work focuses on evaluating the auto-tagging model in a classification setting. However, in the last stages of our work we evaluate the effectiveness of this model in a recommendation setup. Hence, here we will give an overview of how recommender systems are evaluated, which will be partially used in chapter 7.

Recommender systems are some of the most challenging systems in evaluation. The nature of the problem, being largely subjective, results in uncertainty of the real-world

---

[1] https://www.last.fm/

performance of the system. Hence, several evaluation methods and metrics have been used to assess the quality of recommendations. This notion of quality reflects the relevance of the items recommended. However, it also needs to consider other factors such as the diversity of the recommendation or the timing. Naturally, some of these factors are easier to measure than others, which requires defining certain proxies to assess [HZ11].

3.4.1 Main Evaluation Paradigms

The exact choice of the proxies and evaluation scenarios depend on the type of the recommended media and dataset. Previous work [GS15b] categorized those evaluation paradigms into 3 main groups: the offline setting, the user studies, and the online setting. In the following, we will detail how each of them is performed.

Offline

As the name suggests, this type of evaluation is performed offline, i.e. without involving the active users. it relies mainly on the previously collected datasets and ratings from the users. Since this methodology does not include active users, it is one of the most popular and frequently used methods for evaluation [GS15b]. It often serves as a first evaluation phase before evaluating with real users. This scenario is based on one main assumption, that the users behaviour in the collected dataset would be similar to their online behaviour when the recommender system is used.

Offline evaluations are heavily based on splitting the dataset into a training and testing subsets. The quality of the recommender system is then evaluated based on the performance on the testset. Several methods are available for splitting the dataset, which depends mainly on the application domain and the service. In the following we will describe some of the most popular splitting techniques [SB14].

`Random split`: The simplest approach is to split the dataset randomly by selecting a percentage of the data for training and the rest for testing. This splitting is performed without replacement, i.e. the samples used in the training split are never reused in the test split. K-fold cross-validation describes a common practice of repeating this process of splitting and testing K times and merging the results of these repeated tests for higher confidence in the results.

`Given-n split`: In this scenario, only a fixed number (n) of interactions for each user is used for training. The rest of the available data is then used for testing. This scenario allows for an equal representation of the users during training, avoiding bias of users with large amounts of available ratings. Choosing a small value for n allows for testing the system in a setting similar to that of the cold-start case.

`Chronological split`: In many cases, the timing of the recommendation and the evolution of users preferences are much more dynamic. In these cases, it is preferred to use this chronological splitting method. Simply, the dataset used for training is split by considering the data before a certain point of time. The data available after this threshold is then used for testing the system.

While being the simplest evaluation setting, offline evaluations are necessary for preliminary evaluation for a recommender system. However, they suffer from a number of limitations. There is no guarantee that the performance in the offline case will match that of the system when deployed to active users. Hence, further evaluation scenarios with active users are often performed afterwards. Offline evaluation will be the primary evaluation scenario used in our experiments later on.

User Studies

One intermediate scenario for evaluation recommender systems is through user questionnaires. This evaluation is often performed with a small group of recruited users who are asked to interact with the system. The users' responses to the system, along with their explicit feedback, are used to further evaluate the performance. This evaluation, however, suffers from a number of limitations as well. First the number of participants is often small to draw firm conclusions. Additionally, the participants awareness of being in a study affects their behaviour in comparison to a naturally recurring session, which could bias the results.

Online

Online evaluation is the most rigid form of evaluating a recommender system. An online evaluation is performed in a real-world setting with actual users [Fis01, KDF+13]. This is often performed by comparing the users' behaviour with and without the recommender system deployed and use this comparison to draw conclusions. One common way to achieve this is through A/B testing.

A/B testing describes an evaluation setting to evaluate new systems on active users [Ama13]. The test is performed by comparing the users behaviour and interactions with both the newly developed system and the old one. The effect of deploying the new system is then measured and evaluated according to the evaluation metrics of this service. A/B tests are often the last evaluation phase before deciding whether to adopt a new system or not.

While being the most conclusive scenario, online evaluation still suffers from a number of drawbacks. Since this evaluation is done with active users in the service, there is a risk of delivering a negative experience to the users involved in the testing. Hence, extensive offline evaluations are often performed before moving to the potential phase of online evaluations.

3.4.2 Evaluation Metrics

As mentioned earlier, evaluating a recommender system is a challenging task as it is evaluating subjective preferences of users. Hence, multiple evaluation metrics are often used to evaluate different aspects of the system. In the following, we will iterate through the most commonly used evaluation metrics. Those metrics can be broadly categorized as: prediction accuracy metrics and top-N metrics [GS15b].

Prediction accuracy metrics

The first metric is often used to assess the quality of the predictions. In the cases where the ratings are to be predicted, Mean Absolute Error (MAE) and similar metrics derived from it such as root mean square error (RMSE) are used. This metric computes the error in predicting the correct ratings, and hence, the lower the better. Those metrics are defined as follows:

- *The Mean Absolute Error (MAE):* As the name suggests, this metric computes the absolute error, i.e. deviation, between the predicted rating $\hat{r}_{ui}$ and the groundtruth rating r_{ui}, and averages them across all sample in the testset $\mathcal{T}$. It is defined as follows:

$$MAE = \frac{1}{|\mathcal{T}|} \sum_{u,i \in \mathcal{T}} |\hat{r}_{ui} - r_{ui}| \tag{3.8}$$

- *The Root Mean Squared Error (RMSE):* Similarly, RMSE computes the root of the squared deviation between the predictions and the groundtruth. This squaring is to penalize the larger deviations compared to MAE. It is defined as follows:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{u,i \in \mathcal{T}} (\hat{r}_{ui} - r_{ui})^2} \tag{3.9}$$

However, one main limitation for these metrics is their need for explicit ratings. Hence, in cases where the service does not have a rating system, those metrics are not usable. This is indeed the case in our setup, as we do not have available ratings for the items.

Top-N metrics

Another family of evaluation metrics is the Top-N metrics. These metrics are aimed at assessing the quality of the recommendations in the case of providing the top-N items deemed relevant by the system. This relevancy is determined based on the users interactions with the items, e.g. previous tracks from the users history are considered relevant. Some of the most widely used metrics are the precision and recall [BYRN+99]. Let $\mathcal{L}_u(N)$ be the set of items recommended by the system to user u, where N is the number of items to be recommended which is chosen during evaluation. Let $\mathcal{S}_u$ be the set of items deemed relevant to this user. Finally, let $\mathcal{T}_u$ be the set of all users in the testset split. The precision and recall in this case can be defined as:

- *Precision:* The precision metric computes the fraction of relevant items in the recommended set of items. It is defined as follows:

$$Precision@N = \frac{1}{|\mathcal{T}_u|} \sum_{u \in \mathcal{T}} Precision_u@N = \frac{1}{\mathcal{T}_\sqcap} \sum_{u \in \mathcal{T}_u} \frac{|\mathcal{L}_u(N) \cap \mathcal{S}_u|}{|\mathcal{L}_u(N)|} \tag{3.10}$$

- *Recall:* The recall measures the ratio of the relevant recommended items to all relevant items for this user. It is defined as follows:

$$Recall@N = \frac{1}{|\mathcal{T}_u|} \sum_{u \in \mathcal{T}} Recall_u@N = \frac{1}{\mathcal{T}_u} \sum_{u \in \mathcal{T}_u} \frac{|\mathcal{L}_u(N) \cap \mathcal{S}_u|}{|\mathcal{S}_u|} \tag{3.11}$$

However, there is one clear flaw with those metrics, they depend and vary significantly with the number of recommended items N. Increasing N will in turn increase the recall until all relevant items are recommended. However, it will result in decreasing the precision. Hence, similar to their use in the classification setup, a different metric called F-1 score is used to combine the results of these two metrics, such that it will give a balanced average. It is defined as:

$$F1@N = \frac{1}{|\mathcal{T}_u|} \sum_{u \in \mathcal{T}_u} 2 \times \frac{Precision_u@N \times Recall_u@N}{Precision_u@N + Recall_u@N} \tag{3.12}$$

However, those metrics do not account for the actual rank of relevant items in the recommended list [CZTR08]. In cases where a rating metric is available, further evaluation metrics were developed to measure it. Notably, the Normalized Discounted Cumulative Gain (NDCG) and the Mean Reciprocal Rank (MRR) are used for assessing the quality of the ranking. Similarly, we give their formal definition as follows:

- *The Discounted Cumulative Gain (DCG)* is first defined for user u as follows:

  $$DCG_u@N = \sum_{i=1}^{N} \frac{rel_{ui}}{log_2(i+1)} \tag{3.13}$$

  where $rel_{ui} = 1$ if the item at rank i is relevant for user u and $rel_{ui} = 0$ otherwise. *The Normalized Discounted Cumulative Gain (NDCG)* [JK02] is a normalized version of (DCG) and is defined as follows:

  $$NDCG@N = \frac{1}{|\mathcal{T}_u|} \sum_{u \in \mathcal{T}_u} \frac{DCG_u@N}{DCG_u^{\star}@N} \tag{3.14}$$

  where $DCG_u^{\star}@N$ is the best possible DCG_u obtained if all recommended items were relevant.
- *The Mean Reciprocal Rank (MRR)* computes the reciprocal of the rank of the first relevant item in the full ordered list of items, denoted by $rank_u$ for user u, and is defined as follows:

  $$MRR = \frac{1}{|\mathcal{T}_u|} \sum_{u \in \mathcal{T}_u} MRR_u = \frac{1}{|\mathcal{T}_u|} \sum_{u \in \mathcal{T}_u} \frac{1}{rank_u} \tag{3.15}$$

Those metrics are among the most common to evaluate a recommender system. However, based on the specific problem and setting, several other metrics can be used as well. Notably, the mean average precision [BYRN$^+$99], the area under the receiver operating characteristic (ROC) curve [Bra97], the expected reciprocal rank [CMZG09], and the hit-rate [DK04].

3.5 Current Challenges

As shown earlier, recommender systems have been progressing continuously since the emergence of online services. However, there are still several limitations to be tackled by future research. One of the main challenges still is the **"cold start"** one. As explained earlier, new users and items with no previous interactions with the service are particularly challenging to recommend. Although content-based approaches are less prone to this problem, they are still limited in other aspects, notably in the case of new users and in scalability. Content analysis is a rather costly process, specially with complex content such as music.

Another common challenge is the **domain-dependency**. Even though the end goal is similar for most recommenders, i.e. recommend relevant items to the users, it is still highly dependent on the actual service and media to be recommended. While different domains can make use of the progress in recommendations from other domains, they still largely require fine-tuning and adaptation to their specific setting. For example, music recommendation is very particular in its requirements. Unlike books or movies, music require continuous recommendations when the users are active. Additionally, music is largely influenced by the listening situation as explained earlier. Hence, a music recommender system would have a fundamentally different design than those of other domains.

Finally, **context-awareness** remains one of the most common challenges of recommenders, specially in the case of music. While previous evaluation scenarios focused on finding relevant items, it is becoming evident that recommending relevant items at the correct time is just as important. What users consider relevant at one point of time might not be relevant at another. What dictates “good timing” is in itself a challenge. Identifying the influencing factors on the users while interacting with the systems and looking for recommendations is essential in finding the right items to recommend. Hence, there has been a significant focus on developing context-aware recommenders in all the different domains.

The Case of Music Recommendation

The music domain is particular in its recommendation process compared to other domains, such as movies, E-commerce, or books [SZC+18]. Music is often listened to in sessions, each containing multiple short tracks played one after the other. Additionally, it has been shown that these sessions often have a common theme [HHM+20], with a common ”concept“ between the tracks in this session. This theme is influenced by multiple factors including the listener’s situation, time of the day or even the current season [PTM+19]. Additionally, sessions often contain tracks from the user’s recent consumption history [AKTV14], i.e. the listening history is indicative of what tracks a listener is likely to choose in a new session. Users could have multiple sessions with multiple themes in a single day, often with a repeating pattern.

Hence, several approaches have been proposed to recommend music while considering the multiple factors in play. Those approaches were placed in various categories by their developers and reviewers, based on the data used and the setup, including content-based [PB07], sequence-aware [QCJ18], context-aware [RD19], session-aware recommendation [HHM+20, QCJ18], or a combination of them. Furthermore, their mode of use was categorized into: generic next-item recommendation setting [ZTYS18, KMG16], predicting the first item in the next session [RCL+19], as well as predicting all items in the next session [WCW+19]. Hence, music recommender systems operate either actively by recommending one track after the other, or all-at-once by generating a playlist/session of recommended tracks. Those systems also vary in their feedback from users between explicit (e.g. a user liked a track), or implicit (e.g. a user listened multiple times to the same track) [JWK14].

Within this recent growth of recommender systems, music services found an increasing relevance to both the user and the situation [GPT11]. Hence, slowly but steadily there has been a growing interest in music context-awareness, where the focus is on modeling users’ preferences and intents during a specific session.

3.6 Conclusion

In this chapter we gave an overview of the progress, categorization, and evaluation of the recommender systems. Given this survey, we find the recommender systems facing multiple challenges that are constantly being addressed in the research community from different domains. While several approaches try to address all of the challenges jointly, we will be focusing on splitting one particular challenge in a manner that can be further integrated in any music recommender system. This challenge is the context-awareness and the situational influence on the listening preferences. In the following, we elaborate on the current state and challenges within these particular situational use-cases.

Chapter 4

Context-Awareness and Situational Music Listening

As recommender systems progressed, there has been a clear need for providing contextual recommendations rather than only relevant ones [AT11]. The preferences of the users have been shown to be affected by the surrounding factors while using a service. Hence, items' relevance is not dependent solely on the content or the features describing the items, as assumed by non-contextual systems. The recommendation process needs to be dynamic and adaptable to the change in users' needs and preferences. However, defining those influencing factors proved to be a challenging problem that is also dependent on the domain.

The rapid progress in context-awareness across those several domains resulted in confusion on the definition of contexts [Dou04]. Several studies approached context in different ways, either using the available user data directly as the contextual data, or using it as the observed data needed to infer the latent context. Even in the second case, the range of the latent contexts varied between activities, moods, location, time, companionship, etc. This is not uniform across all domains, and each domain requires a refinement of the context categories that concern its users. The challenge of these loose boundaries in defining context has been well observed in previous work since the emergence of context-awareness [AT11]. Other synonyms for context-aware computing include adaptive, responsive, situated, context-sensitive, and environment-directed computing [Dey00].

4.1 Context Definition

The concept of context has been the focus of several previous studies. The term "context-aware" was mentioned for the first time by Schilit and Theimer in 1994 [ST94]. They defined it broadly as *"location, identities of nearby people and objects, and changes to those objects"*, and highlighted the necessity of considering it when designing systems for a mobile environment. Since then, several new definitions were proposed that added additional factors such as time and season [BBC97], identity and environment [RPM98] and the emotional state of the user [Dey98].

For the majority of context-aware systems that came after, the most commonly and widely used definition for context was the one presented by Abowd et al. [ADB^{+}99] as follows: *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user, and applications themselves".* However, this broad definition, while enclosing the different types of contexts, requires further refinement to more tangible factors when adapted to a specific domain.

Another very relevant concept to context is the user intentions when interacting with a service. Context describes the cognitive and social factors influencing the user. However, understanding the goal and intention of the user during this context is what is needed to provide relevant recommendations at this point of time. Several subsequent studies [Son99, Coo01, IB04, QM02] tried to explore this broader definition of context, i.e. the cognitive and social environment of the user, and how it relates to the influencing situations such as location and time. However, the broader the definition is, which is more inclusive, the harder it is to identify tangible factors that can be used when designing a system, specially a system that relies on "contextual tags".

One practical definition of context [Ake17] that can help identify the factors, to be considered when designing a system, is: `‘‘a set of dimensions that describe and/or infer user intentions and perception of relevance’’`. In this sense, the users intentions are the end goal to be inferred, while the available contextual information is the tool used to infer this intention.

Examples of contextual information

Given the previous definitions of context, there has been several studies using a subset of contexts based on the exact recommendation setting. Some of the most commonly used contexts are:

- **Location information:** Location has to be one of the most widely used contextual information. This information can be used in various recommendation scenarios [LSEM12, YYLL11]. Broadly speaking, the users' geographical location in itself is useful in identifying the cultural and local preferences for a service that operates around the globe. More specifically, it can help inferring the current activity of the user, e.g. in office or in the gym, which in turn influences the user preferences. Additionally, for some specific domains, e.g. recommending a cafe or a restaurant, considering the location information is a must for the service to work properly. In the case of music recommendations, various previous studies have proposed entire systems centered around the location of the users [SS14, KRS13]
- **Temporal information:** Time is one of the most influencing factors when it comes to recommendations. Time can either be the hour in the day, the day of the week, or even the season. This is particularly evident in services that provide clothing or tourism. Several previous studies have focused primarily on the effect of considering the temporal information in recommender systems [CDC14, Kor09]. Similarly, music listening is a dynamic process that could be greatly influenced by the temporal information. Hence, several systems were proposed that adapt the music recommendations according to the time context [DF13, CPVA10]
- **Social information:** Another important influencing factor on users' decisions is the social surroundings and networks [MYLK08, AT11]. For example, movies recommendation would differ greatly when alone, compared to being accompanied by either friends or family. Previous work on music recommendation has also studied and identified the influence of the social factors on both the listener preferences and reaction to music [TBLY09, LJV13].

4.2 Paradigms for Incorporating Context

Contextual information are multi-dimensional by nature. While traditional recommender systems derive recommendations using the user and item data solely, context-aware systems need to consider several additional information [HSK09]. For example, merely the temporal contextual information can be extended to include several dimensions such as the day of the week, the time of the day, or the season. Different values for each of those dimensions will consequently require different recommendations. A context-aware system must be able to incorporate this additional information while deriving the recommendation, alongside the user and item information. We described the objective of traditional RS as estimating a utility function $\mathcal{R}$ defined on the space $U \times I$. For the case of context-aware recommender systems (CARS), this problem is extended with extra contextual dimensions $U \times I \times C$, where C represents the set of relevant contextual factors. Previous work [AT11] identified 3 paradigms to incorporate this contextual information: contextual pre-filtering where context is used for input data selection, contextual post-filtering where context is used to filter recommended items, and contextual modeling where context is directly incorporated into the predictive model.

Contextual pre-filtering

As the name entails, this paradigm aims at filtering the data before inputting it to the recommender system. The contextual factors are used to select the data to be considered in the retrieval stage. For example, a popular method for pre-filtering is splitting the user's profile into several micro-profiles, each corresponds to a specific contextual state [BA09]. Similarly, this can be applied on the items to generate a reduced list of potential items that match the current contextual state [BR14]. Finally, this splitting can be further done on the users and items simultaneously [ZMB13].

Contextual post-filtering

In contextual post-filtering, the procedure of filtering the potential set of items is performed after retrieving the set of recommended items [HMB12]. This is achieved by either removing irrelevant items, or reordering them based on the contextual information. Both pre-filtering and post-filtering are particularly useful when adapting a traditional recommender system to a contextual setting. However, choosing between one of these two strategies was shown to largely depend on the problem with no universal advantage of one over the other [PTG⁺09].

Contextual modeling

Finally, contextual modeling refers to a set of algorithms that directly exploit the contextual information in training a multidimensional system. One of the most popular methods for contextual modeling is tensor factorization [KABO10]. Similar to matrix factorization, the multidimensional tensor is approximated using reduced matrices. The additional dimensions in this case correspond to the contextual data available to the system. Instead of deriving only two reduced matrices, for the users and items, tensor factorization derives several matrices each corresponding to one of the dimensions in the tensor, including the traditional user and item dimensions. Several previous studies have used this approach for developing contextual recommenders [KABO10, Ren12]. One key disadvantage of this type of modelling is the added computational complexity, resources, and the challenge of scalability to large datasets.

4.3 Context Acquisition

As described earlier, context includes a long list of factors describing the situations of the users. One of the most challenging aspects of context-awareness is defining those relevant factors for a given service, and retrieving them to be incorporated in the recommendation process. Previous work [AHT08] highlighted the principal way to retrieve the contextual information as:

- **Explicitly**: The easiest approach is to directly ask the users to enter their usage situations while using the services. For example, a previous study [LYM04] asked the user to select concepts describing their context from the an online ontology in order to identify and model the user's context.
- **Implicitly**: A different approach to represent context is through collecting the relevant data to describe the surrounding environment of the user, such as GPS location or the time of the day [PTG08, GPZ05, HSK09]. This data will then be used directly as the contextual information provided to the recommender.
- **Inferring**: A more elaborate approach is to try to infer the explicit situation of the user using the previously collected data. This methodology relies on using data mining and statistical inference to model the current situations of the users. Given our quest towards interpretable representation of contexts, this paradigm will be the one followed in our work.

4.4 Music Context-awareness

As described earlier, context is any information that influences the preferences and interactions of the user with the service. For example, the listening situation of the music service users such as their activities and moods can be influencing the music style and choice. Hence, it is important to be considering such information when recommending music, alongside the longterm preferences [NR04]. Given the dynamic nature of music listening, the previous research focused on two primary sub-problems, first identifying the relevant influencing situations for music listening, and second integrating these information effectively in a contextual recommendation system.

In this section we review this previous work that address contextual music recommendations. We first summarize the work from the psychomusicology domain on identifying how the listening situation influences the listening preferences. Afterwards, we give an overview on previous work aiming at integrating this contextual information in a recommendation system.

4.4.1 Relevant Studies from the Psychomusicology Domain

As music can be listened to in various situations [GL11], there have been many studies on the relationship between music preferences and the listening situation. For example, an early work by North et al. [NH96c] studied the influence on 17 different listening situations on music preferences. The study showed that music preferences are not only

Table 4.1 – Situations studied by North et al. [NH96c] and their categorization

Situation	Category
At an end-of-term party with friends	A
At a nightclub	A
Jogging with your Walkman on	A
Doing the washing-up	A
Ironing some clothes	A
In the countryside	C
In a French restaurant	B
At a posh cocktail reception	B, D
Having just broken up with a boyfriend/girlfriend	D
On Christmas Day with your family	A, C
Your parents have come to visit	B
First thing on a Sunday morning	B
Last thing at night before going to bed	B
Making love	B
Trying to woo someone over a romantic candlelit dinner for two	B
In church	C
Driving on the motorway	A, C

A: activity, B: localized subdued behaviour, C: spirituality, D: social constraint

dependent on the emotional response they evoke, but also on how they affect the quality of the listening situation. This suggests that music preferences are strongly associated with the listening environment. They categorized these 17 different situations into: *activity, localized subdued behaviour, spirituality, and social constraints*. Table 4.1 shows the studied situations and their categorization. Additionally, they studied the relationship between these situations and some acoustic attributes of the music, such as loudness and rhythm, and found that they are associated with the listening situations. This reflected the potential of using the audio content to find the appropriate music for a specific listening situation.

A later similar study [SOI01], also categorized different listening contexts into 3 categories: *personal, leisure, and work*. They further expanded each category into subcategories that are more specific to the situation. For example, personal is categorized into three subcategories: personal-being, e.g. sleeping or waking up, personal-maintenance, e.g. cooking or shopping, and personal-travelling, e.g. driving or walking. Table 4.2 shows a summary of the situations studied and categorized in [SOI01]. Earlier, Sloboda et al. [Slo99] also studied the functions or purpose of listening to music for different users. They found that users listen to music for different purposes, such as *activity*, e.g. waking up or exercising, and *mood enhancement*, e.g. to put in better mood or for motivation.

The previous studies all showed that there is a strong relationship between user's context, including listening situation, surrounding environment, cultural background, and user's demographics, and music listening preferences. However, there is an additional important factor in music preferences, which is the user's personality and own taste. Ferwerda et al. [FYST19] studied how the personality traits affect users taste and music preferences. Similarly, Rentfrow et al. [RGL11] studied the links between music preferences and user's personality.

All of these previous studies show that there are multiple factors affecting the user's choice of music in any given moment and what they would consider relevant. However, up till recently, most used recommender systems treated the user from a one dimensional point

Table 4.2 – Situations studied by North et al. [SOI01] and their categorization

Category	Situations
Time fillers	doing nothing, waiting
Personal - being	states of being (e.g., sleeping, waking up, being ill)
Personal - maintenance	washing, cooking, eating, housework, shopping
Personal - travelling	leaving home, driving, walking, going home
Leisure - music	listening to music
Leisure - passive	watching TV, putting on radio, relaxing, reading
Leisure - active	games, sport, socialising, eating out, chatting with friends
Work - self	writing, computing, marking/assessing, reading for study
Work - other	planning for meeting, in lecture/seminar

of view by recommending tracks that are likely to match his/her global taste, regardless of any of the other factors. Recently, and given the findings from the previously mentioned cognitive studies, context-aware recommender systems have been gaining increasing attention from the research community. However, the current state of these systems is far from ideal and there is plenty of room for improvement. For example, the definition and usage of user context is often picked differently in each study. There is no common definition, or categorization of different possible contexts in music consumption that could better help the development of this field. In the following, we will look into how previous recommender systems integrated those various contextual information in the music domain.

4.4.2 Previous Work on Context-aware Music Recommender Systems

Previous studies on context-aware recommender systems used a variety of contextual information including location, activity, time, weather, or raw sensor data collected from the user's phone. An extensive review of the progress related to context-awareness in music can be found in the recently conducted survey by Lozano et al. [LMJBVR+21]. In the following, we will iterate through some examples of previously developed contextual recommenders for music.

Cheng et al. [CS14a] developed a location-aware recommender system for music. They proposed a probabilistic model that considers the user listening history, music content features, and user's location. They focused on five locations: *canteen, gym, library, office, and transportation.* Due to difficulties in accessing the user's location, the approach relied on detecting the location using audio content. However, the trained model had unsatisfying classification accuracy which would lead to noisy recommendations. Additionally, they did not consider the problem as a multi-label classification problem, i.e. the same track can be listened to in different locations.

In a different study [KR11], Kaminskas et al. studied location-aware recommendation for places of interest (POI). They relied on using emotion tags that are associated with both music and POIs to find music that is more suitable for certain locations. Afterwards, they extended their work to include a knowledge-based approach to find relationships between music and POIs [KRS13]. This is one of the early explorations of music auto-taggers employed for predicting listening situations, even though it was limited to locations only. A similar study [BKL+11] investigated a recommender systems for playing music in cars. They proposed using information relevant to driving such as weather, surrounding landscape, traffic condition, and user's mood.

Reddy and Mascia [RM06] proposed another location-aware music recommender system called Lifetrak that uses the user's current context, including location (represented by a ZIP code) as well as time, weather, and activity information. The context is obtained using the sensors of the mobile. Foxtrot [AS11] is another mobile music application that allows the user to assign certain locations to different tracks. However, instead of automatically tagging and associating music with locations, they relied on a crowd-sourcing approach to tag the data.

Other studies focused on using the user's activity as the contextual information. Wang et al. [WRW12] developed a recommender system for different daily activities. They relied on mobile phone sensors to detect user's activity from a set of 6 activities: *running, walking, sleeping, working, studying, and shopping*. The approach was constrained by the limited amount of available data, specifically music tracks that are labeled with certain activities. The study required that part of the data be hand-labelled using human participants, which is time consuming and not scalable.

Another study on music for activities [YLLH17], Yadati et al. relied on an automated procedure to label the data using Youtube search queries on "music for X", where x is an activity. However, they focused on only three activities: *relax, workout, and study*. Dias et al. [DFC14] relied on a crowd-sourcing approach to label tracks with their suitable contexts. Lee et al. [LCHL17] proposed using hand-coded rules or a trained classifier to map the detected activity to the suitable music. Additional studies that focused on activities for music recommendation include [LHR09, MvNL10, CCG08].

Given the previous summary, we find that several studies have been focusing on methods of integrating the contextual information in the music recommendation process. However, one common challenge in all the previous results is the lack of common set of situations, a labelled dataset, or a uniform procedure to label new data. Additionally, most of the previous approaches have been “situation-specific”, i.e. they were developed for only a subset of situations such as the location. This scattered overly-specialized work on music context awareness has resulted in weak adaptability of those proposed methods in an industrial setting. Nonetheless, it shows the level of necessity and interest in developing music recommenders that are aware of the listening situation and intentions.

4.5 From Context-aware Systems to Situation-driven Systems

Recent work [PCL$^+$16] introduced the concept of the “contextual turn”, creating the need for context-driven RS (CDRS), in which the context is central rather than just an additional information. CDRS aims to contextualize recommendations, i.e. fitting the recommendations to the user intent and situation, rather than personalize, i.e. fitting recommendations solely to the individual. The main assumption is that users have more in common with other users in the same situation than with their previous preferences. Recommendations are based on what is going around the user, i.e. the user's situation, and on what the user is trying to accomplish, i.e. the user's intent.

The context concept is perceived as the cognitive, the social, and the professional environment related to several factors like time, locations, etc. The use of these factors is very crucial to boost the performance of any system. However, they only form a low-level layer extracted from available devices/sensors, which need to be interpreted into a high-level layer that defines a situation. As expressed by [Bou13b, Bou13a], *“Situation awareness focuses on the modelling of a user's environment to help him/her to be aware* *of his/her current situation"*. This formulation of the problem is inline with our goal in representing the contextual influence through an interpretable high-level tags to describe the use-case of a given track. Hence, in our work we will be concerned with identifying those explicit listening situations, and develop a situation-driven [MDILA12] framework for recommendations, which can be easily interpreted by the user.

Finally, it is important to emphasis the complexity of defining an intent when it comes to listening to music. While previous work highlighted the influence of the listening situation on the preferences, it is important to remember that listening to music can be in itself a separate activity. In this case, traditional approaches of music recommendations, e.g. finding music that fits the general taste of the users or help them explore similar music, would be the appropriate approach for the recommendation. Hence, we intend to simplify the problem by decoupling the use-cases of a situational influence from this traditional case, unlike some approaches that intend to address both cases simultaneously.

4.6 Conclusion

In this chapter, we presented an overview of the concept of context-awareness from previous work. We highlighted the challenges in defining, collecting, and incorporating the relevant contextual information from a recommendation point of view. Furthermore, we surveyed the previous work from both the psychomusicology and the recommendations domains that addressed the problem of context-awareness in the specific case of music. This highlighted the current challenges in having a uniform set of relevant interpretable contexts and standard datasets for studying this problem. Finally, we explained why we separated our setup, which aims at defining semantic situations for interpretability, from the different scopes of context-awareness. Previously, context-aware approaches aimed at integrating the contextual information directly rather than disambiguating the listening situation.

Part III

Auto-tagging for Contextual Recommendation

Chapter 5

Situational Music Autotaggers

The work presented in this chapter has been published in the following papers:

Ibrahim, K. M., Royo-Letelier, J., Epure, E. V., Peeters, G., Richard, G. Audio-based auto-tagging with contextual tags for music. *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020.

Ibrahim, K. M., Epure, E. V., Peeters, G., Richard, G. "Confidence-based Weighted Loss for Multi-label Classification with Missing Labels." *Proceedings of the 2020 International Conference on Multimedia Retrieval*, Dublin, Ireland, 2020.

5.1 Introduction

In this chapter, our goal is to evaluate the efficacy of music auto-taggers in identifying the potential listening situation using the audio content of the track. In order to achieve this, we have to overcome a number of challenges beforehand. The first challenge is to find a reliable method to link the listening situation and the listened tracks, to subsequently analyze the content of these tracks. The second challenge is to identify the set of situational tags to be predicted. As we aim for interpretability, it is important to identify these tags as interpretable keywords that can be shared with the users. Finally, the last challenge is to find or collect a reliable and large dataset that allows us to train a predictive model and draw conclusions from our studies. Hence, we start by reviewing certain similar studies that inspired the decisions we made for our approach.

As explained in Chapter 4, few studies have already addressed the annotation of music datasets with user context tags [YLLH17, WRW12]. However, there has been no standard procedure on how to find context tags relevant to music and how to employ them. Previous studies have focused on either a subset of contexts, e.g. only locations, or a number of contexts that were defined arbitrarily by the authors [NH96c, WRW12]. Additionally, even scarcer research has primarily investigated the relationship between audio content and listening contexts, and the feasibility to automatically predict context from a music track's audio content [WRW12]. Such studies are important for automatically generating context-aware playlists/sessions [CLSZ18] or for facilitating music discovery by context tags and reduce the semantic gap.

Our work in this chapter proposes the following contributions: 1) a process to label music tracks with context tags using playlists titles; 2) a dataset of ∼50k tracks labelled with the 15 most common context tags, which we make available for future research; 3) benchmark results of a trained auto-tagging model to predict context tags using audio content; 4) a strategy to account for the confidence in the sample tags to overcome the problem of missing negative labels, that we observed to hinder the training of the auto-tagging models [CFCS18].

5.2 Defining Relevant Situations to Music listening

As described in the previous chapter, the listening situation is an important factor in influencing the music preferences of the listener. One major challenge in studying the influence of the listening situation is having a clear and concise definition for these situations. Several previous studies have been trying to answer this question through different approaches. Some studies conduct surveys on music listeners to narrow down potential listening situations [NH96a]. Other approaches rely on available data in streaming services to derive a model for those situations in a data-driven manner [HHM^{+}20]. Here, we first iterate through one particular relevant study that aimed at understanding and deriving a model for the listening situations by using playlist titles as proxy. Afterwards, we go through our proposed pipeline to link the most frequent listening situations to the audio content, overcoming a number of challenges in that previous study [PZS15].

Background

Understanding why people listen to music and their intent at a given time has been a recurring question in the research community long before the emergence of MIR. Fields such as cognitive psychology have concerned with answering this question, often through surveys with music listeners. A series of studies by North and Hargreaves, one of earliest in this domain, simply asked the question “how musical preferences might vary with the listening situation?” [NH96a, NH96b, NH96c, HN99]. These studies focused on validating the hypothesis that listening preferences change with the listening situation. They successfully validated this by observing the variance of several music attributes across several situations. The selected situations were based on the social/environmental situations that were shown to have an emotional response on the participants from previous research [RWP81]. Although those situations were not primarily selected based on their influence on music, they proved the influence of situations on musical preferences, leading the way to further studies on the influencing situations in music listening.

As work on contextual recommendation progressed, new studies attempting to categorize those different contexts were conducted. Several different categorizations of contexts were presented in different studies [RDN06, Dey98, Dey00, HAIS^{+}17a]. Those categorizations helped narrow down the scope of relevant situations significantly. However, so far there has been no data-driven approach to extract all relevant situations until a series of studies by Pichl et al. [ZPGS14, PZS15, PZS16], which proposed using the playlist titles coupled with social media data as proxy to infer the listening context. Such data-driven approaches allow for scalability in collecting large datasets, which is essential in training complex models.

Playlists Titles as Proxy

A new method that extracts and integrates contextual information from playlist titles into the recommendation process was proposed by Pichl et al [PZS15]. The method is based on the findings in previous work [CBF06] on how people create playlists with a specific theme/function. Their studies [PZS15] validated the potential of using this proxy to improve contextual recommendations. As we will be employing the majority of their approach in our dataset collection stage, we will be giving extended description of their pipeline.

The main goal of Pichl's approach [PZS15] is to create "contextual clusters" derived from playlist titles. They define a contextual cluster as a collection that aggregates different playlists under a common context. For example, playlists titled with "my summer playlist", "summer 2015 tracks", "finally summer", and "hot outside" belong to a summer-related context. To achieve this, they need to perform two objectives: 1) to homogenize and clean the playlist titles, and 2) to derive a distance measure for the titles similarity. These studies were conducted on the #nowplaying dataset [ZPGS14].

The first objective is achieved by utilizing common NLP techniques. These techniques include lemmatization, a technique to find the lemma of a given word, i.e. the base form. Additionally, they clean the titles by removing non-contextual playlists that are named after a specific artist, or album, using named entity recognition techniques. To overcome the short length of these titles, they extend the title with synonyms and hypernyms using WordNet [Mil95]. The results of this stage is playlists with extended titles that are assumed to reflect the contextual use of this playlist.

The second objective is achieved using term frequency-inverse document frequency (TF-IDF) matrix [R+03]. i.e. using a bag-of-words with the TF-IDF weights to represent each playlist title as a vector. This representation can then be used with a clustering algorithm to cluster playlists with similar titles together. They test the effect of these contextual clusters on a collaborative-filtering recommendation method. Their results showed a significant improvement of the recommendations compared to the non-contextual ones.

While these contextual clusters proved to be useful in a recommendation case, it is not clear how interpretable these clusters are, i.e. is it possible to use semantic tags to describe the content of a cluster? To further validate the potential of clustering playlist titles to retrieve a context representation encapsulating similar contexts, we analyzed the results retrieved from following Pichl's method, both on the #nowplaying dataset and on the playlist catalogue in Deezer.

The Challenge of Automatically Clustering Titles

The #nowplaying dataset contains ~157K unique playlist titles. The goal is to cluster the playlists based on their names in a way where the clusters group playlists that are sharing a similar theme, ideally the context. We investigated different ways to represent the playlist titles and different clustering approaches, in order to test with up-to-date methods in text representation and clustering since the first study was conducted. Regarding text representation, we experimented with different methods: 1) Term frequency–inverse document frequency (TF-IDF) (similar to Pichl's work), 2) Word2Vec pretrained embeddings [GL14] (trained on google news articles), 3) GloVe pretrained embeddings [PSM14] (trained on Wikipedia articles), and 4) GloVe embeddings retrained on musical articles. Regarding the clustering techniques, we experimented with: 1) K-means clustering (similar to Pichl's work), 2) Hierarchical clustering, and 3) Latent Dirichlet Allocation (topic modeling).

`Results:` Throughout those several experimental setups, we observed similar outcomes. Our analysis of the outcome clusters gave us various insights. The first observation was that indeed some of the clusters made sense in terms of listening context, e.g. one cluster included all meditative and relaxation playlists with keywords such as *'meditation', 'relaxation', 'sleep'*, and *'nature'*. Another included classical music with the keywords *'live', 'classic'*, and *'orchestra'*. However, we found certain challenges with using this approach. The main drawback is that clusters are made in a way where titles are semantically close according to the word embeddings. General linguistic semantic similarity does not necessarily reflect the similarity as a listening situation, e.g. one cluster included keywords such as *'country'* with *'house'*. All music genres are relatively indistinguishable because, for the word embeddings, they are all considered similar. We further investigated these drawbacks on playlists retrieved from the Deezer catalogue.

Nonetheless, the playlist title approach is indeed a reliable pathway to link the situations to the corresponding tracks, even if the clustering step is unreliable. We further validate this observation by employing the same pipeline to retrieve contextual clusters from the playlists available in Deezer. However, again after further investigation, we found that these clusters are not sufficiently specific to the situational use cases of music, similar to the results from the #nowplaying dataset. In most cases, even after filtering, the playlists were not exclusively made for situational use-cases. Additionally, the clusters are not successfully merging similar situations. This is mainly due to the difference in semantic similarity from the actual meaning of musical keywords. We, again, found clusters that included the word "House" with keywords that reflect being at home, while the content of those playlists was intended for the musical genre.

Given the noise in the produced clusters, and as we will be closely studying the relationship between the audio content and the listening situations, we prioritize the quality of the studied situations and their corresponding tracks. Hence, we resort to a semi-automated pipeline that ensures the quality of selected situations, while still utilizing the powerful potential of playlist titles.

A Semi-automated Pipeline

Inspired by the previous work on retrieving situational playlists, we rely on the same concept, but with more rigid filtering to ensure the quality of the investigated playlists. Our previous analysis revealed that inferring the situations entirely from the titles alone is not sufficient and prone to noise. Hence, we add a manual filtering stage to retrieve exclusively the situational playlists. The filtering is achieved by collecting situational keywords to look for in these titles.

Given the large pool of potential situations, we seek to narrow it down tangible keyword-based situations. To start, we collect an extensive list of situations from similar previous studies [NH96c, WRW12, GS15a, YLLH17]. We then extend this list with synonyms from WordNet similar to Pichl's previous work [PZS15]. Additionally, we also retrieve the most frequent hashtags that appear with our initial set of keywords on Twitter. This phase results in a large pool of potential keywords that are not all necessarily situational. Hence, we apply manual filtering to ensure the selected keywords are situational.

Defining which keywords are situational and which are not is also challenging, since it is a relatively subjective decision. Hence, we rely on previous definitions and categorization of situations related to music to apply this filtering. Several previous studies have different definitions of relevant situations. We have chosen to follow the categorization proposed in [KR12], because it is targeted exclusively to music listening situations, but based on previous studies in the same domain [Dey00]. They categorize music listening situations primarily into *environment-related (information about the location of the user, the current* *time, weather, temperature, etc.), and user-related (information about the activity of the user, the user's demographic information, emotional state).* To further restrict the situations to tangible keywords, we exclusively match them to one of those four subcategories: `Activity, Location, Time, Mood`. If one of the collected keywords fits into one of these categories, we consider it a situational keyword and retrieve playlists containing it in the title. This stage results in ninety six keywords each fitting one of the previous categories. The full set of keywords is shared as well[1]

Once the keywords are established, we apply a similar procedure to match them with the publicly available playlist titles, i.e. parsing the titles. We first collect all the public playlists in the Deezer catalogue that include any of the keywords. Afterwards, we removed all playlists that contain more than 100 tracks, since playlists with many tracks tend to be less focused on a specific situation and rather noisy. We also removed all playlists where a single artist or album makes up more than 25% of all the tracks in the playlist, to further ensure that the playlist was not intended for a specific artist. Additionally, we removed playlists with the same set of tracks, which are often copies created by a different user, and keep only one version.

This procedure allows us to have quality playlists targeted at a specific situation. These playlists can provide a range of information that we will be heavily relying on in our studies. It provides the set of tracks intended for this situation, allowing us to investigate how the audio content relates to this situations. It is also linked to information about the users, both the creators and the listeners, which allow us to study the role of the users in defining the content of the tracks. Finally, it is linked to detailed data about the interactions between the users and the service while listening to these playlists, which will allow us to study how far these interactions can help identify the listening situations.

Challenges with Keyword-based Playlist Context Annotation

Although relying on the playlists titles and situational keywords helps us achieve our goal in collecting a dataset with situational tags, there are a number of shortcomings with this approach. The primary one being the absence of a structured representation of these situations. Many situations have intrinsic relationships with each other, which are expected to be reflected in the associated music as well. For example, sport-related situations are highly similar to each other. However, the brute keyword based approach ignores these overlaps between situations. The problem is harder to solve since defining a line between distinctive situations from similar ones is rather subjective, which is a complex challenge. However, the keyword-based approach is sufficient to collect a reliable dataset and conduct our studies. Hence, until a hierarchical representation of these listening situations is properly developed, we proceed without associating different situations with each other and treat each independently.

5.3 Dataset collection

Given the previously mentioned procedure for linking the audio content with the listening situation, the first step of studying the relationship of contexts and audio is to collect a well-labelled and reliable dataset. We will be following the procedure described earlier. For our first study, we selected a subset of contexts to work on. For this initial draft of the dataset, we selected only the 15 most frequent keywords we found in the Deezer

---

[1] https://bit.ly/2XzNI4t

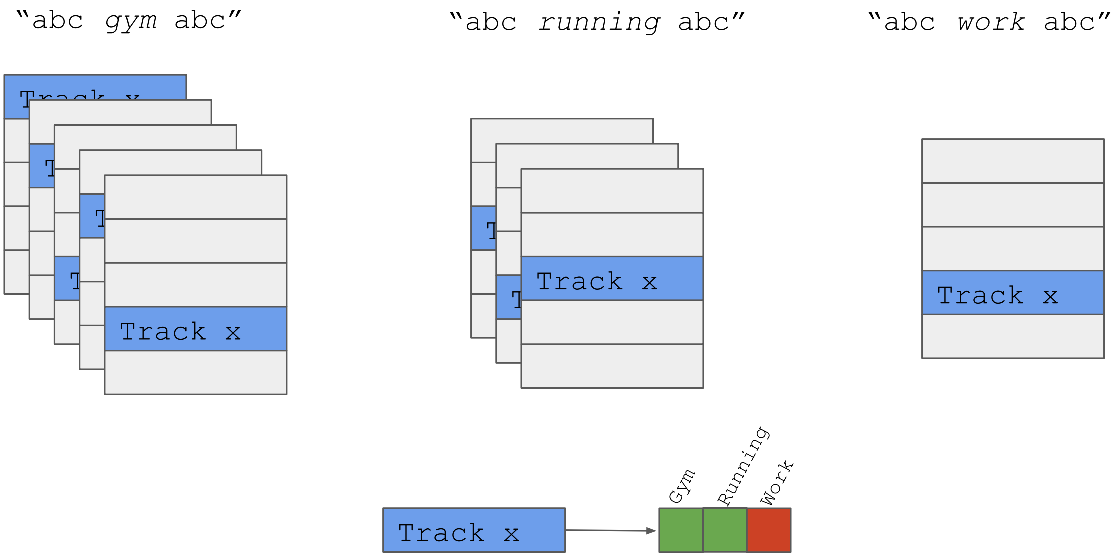


Figure 5.1 – An example of a given track "x" being labelled based on its appearance in different situational playlists

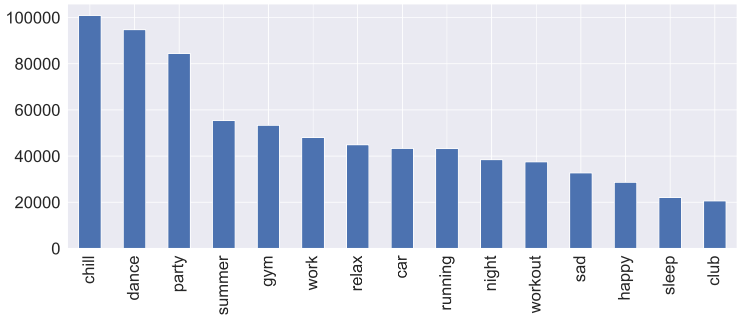
catalogue. The keywords are:
`car, chill, club, dance, gym, happy, night, party, relax, running, sad, sleep, summer, work, workout`

The following step is to filter the playlists to include only contextual playlists. We first collected all the public playlists in the Deezer catalogue that included any of the 15 keywords, and applied the filtering steps mentioned before. We label tracks that appeared in more than 3 playlists containing the same contextual keyword, out of the 15 possible contextual keywords, with that contextual label. For example, a track that appears in 5 playlists containing the word "gym", 3 playlists containing the word "running", and 1 playlist containing the word "work", would be labelled as "gym" and "running" but not "work. This is to ensure we have high confidence that the selected track belongs to the labelled context. Figure 5.1 provides a visual representation of this filtering step.

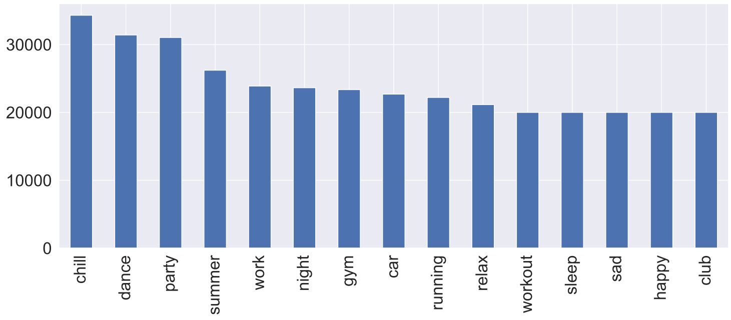
Dataset balancing

After applying the previous filtering to the catalogue of Deezer, we retrieved 612k playlists that belonged to one or more of the 15 selected contexts. The playlists contained 198k unique tracks. However, the dataset was highly imbalanced due to the popularity of some contexts compared to others. Figure 5.2a shows the number of tracks that were labeled with each of the context classes. We find that certain classes are less represented compared to other popular classes, which proved to be problematic in our experiments. Hence, we balance the dataset to keep a nearly equal number of tracks in each context class.

Since we are working with a multi-label problem, i.e. one sample can belong to multiple classes at the same time, it is not possible to have exactly the same number of samples in each class. Hence, we apply an iterative approach of adding samples to incomplete classes with a limit of 20K tracks, which is the number of tracks in the least represented class. Some classes are exceeding the limit due to their co-occurrence with other classes. However, the dataset is more balanced after this filtering. The number of tracks dropped to 49929 unique tracks. The new distribution of tracks can be seen in Figure 5.2b. Table 5.1 shows a comparison between the balanced and unbalanced datasets. Although the balanced dataset reduced the number of samples significantly, the ratios of positive



(a) Before balancing



(b) After balancing

Figure 5.2 – Number of samples in each context class before and after balancing

Table 5.1 – Summary of the dataset before and after balancing

Dataset	Unbalanced	Balanced
# samples	198402	49929
Avg. # positive samples/class	49837.13	23992.3
Avg. # positive classes/sample	3.76	7.2
Avg. ratio of positive samples per label	0.25	0.48

samples and labels is more balanced. The balanced dataset contains on average 24k positive samples per label and 7 labels per sample. We selected a split of 65% training, 10% validation, and 25% testing. We applied an iterative sampling scheme to ensure that there is no overlap of artists or albums between the splits, while having same proportional representation of each context tag [STV11].

We distribute the collected dataset to the research community [2], which is composed of the track ID in the Deezer catalogue and the 15 contextual labels. The audio content for each track is available as a 30 seconds snippet through the Deezer API using the track ID.

Analysis of Context Co-occurrences

As we mentioned earlier, the nature of the problem is a multi-label problem. Hence, the co-occurrences of context tags enable us to learn about the relationships between contexts. In Figure 5.3, we show the number of tracks co-labelled with each pair of contexts. We observe some interesting patterns in these co-occurrences. For example, we find that the three contexts “relax", “sad", and “sleep" co-occur more often together than with other contexts. This matches our expectation about the music style of the tracks

---

[2] `https://doi.org/10.5281/zenodo.3648287`

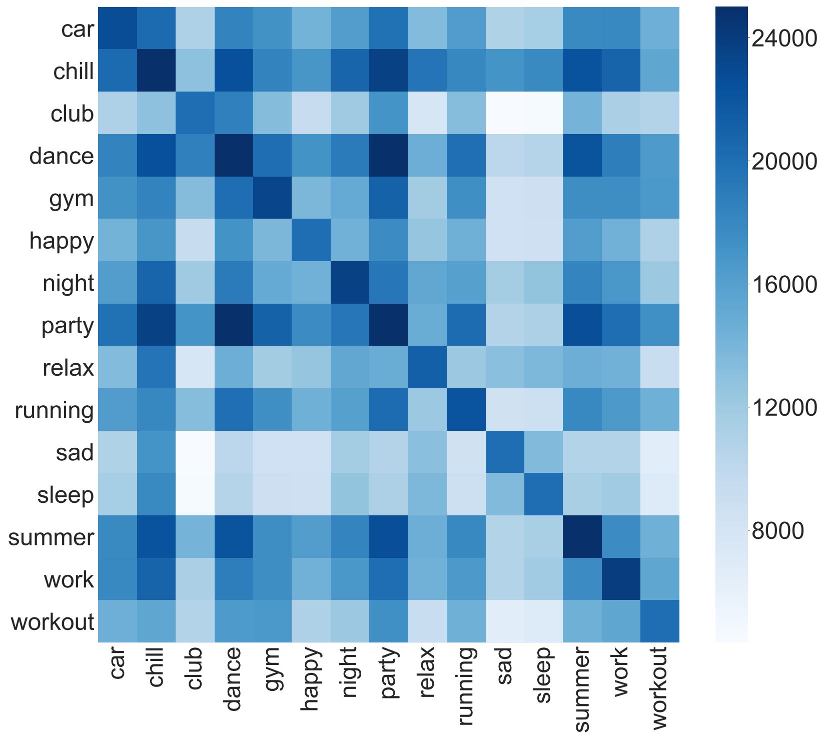


Figure 5.3 – Tracks co-occurences between contexts

related to these contexts to be rather calm and soothing. We also find that contexts such as “club", “dance", and “party" often co-occur together, having most likely associated energetic tracks. We also observe that “chill" and “car” often co-occurs with all the other contexts. This indicates that certain contexts are potentially user-specific and would require additional data about users and music listening cases, apart from audio, for being inferred. It also hints that certain keywords, like “chill”, are being used is a generic manner to describe different situations, with no uniform influence on the listening preferences.

5.4 Multi-Context Audio Auto-tagging Model

Our goals are to predict contexts for a track given its audio content and to assess to what extent this is possible. There has been a number of approaches proposed to auto-tag tracks using audio content. The most recent, best-performing approaches rely on Convolutional Neural Networks (CNNs) applied to the melspectrograms of the input audio [CFS16a, PPNP+18], as explained in Chapter 2. We selected one of the previously proposed and commonly used models by Choi [CFS16a], which is a multi-layer convolutional neural network applied to the melspectrograms.

We trained the network with an input size of 646 frames × 96 mel bands, representing 30 seconds from each track cropped after 30 seconds from the start of the track to match the Deezer preview samples, for reproducibility purposes. The output corresponds to the 15 context tags. We applied a batch normalization on the input melspectrograms followed by 4 pairs of convolutional and max pooling layers. Each convolutional layer has a fixed filter size (3×3) and (32,64,128,256) filters respectively followed by a ReLu activation function.

Table 5.2 – Results of the CNN model on our context-annotated dataset.

	HL ↓	AUC ↑	Recall ↑	Precision ↑	F1 ↑	TN Rate ↑
Car	0.39	0.65	0.58	0.58	0.58	0.63
Chill	0.27	0.71	0.93	0.75	0.83	0.29
Club	0.24	0.84	0.64	0.74	0.68	0.85
Dance	0.26	0.8	0.83	0.79	0.81	0.58
Gym	0.34	0.72	0.74	0.62	0.67	0.6
Happy	0.34	0.7	0.46	0.61	0.53	0.8
Night	0.44	0.58	0.54	0.54	0.54	0.58
Party	0.26	0.77	0.86	0.76	0.81	0.53
Relax	0.31	0.74	0.66	0.63	0.65	0.7
Running	0.38	0.68	0.64	0.58	0.61	0.61
Sad	0.22	0.85	0.74	0.71	0.73	0.8
Sleep	0.23	0.84	0.73	0.71	0.72	0.8
Summer	0.37	0.67	0.74	0.63	0.68	0.51
Work	0.41	0.62	0.61	0.57	0.59	0.57
Workout	0.3	0.77	0.62	0.63	0.62	0.75
Average	0.32	0.73	0.66	0.69	0.67	0.64

We used max pooling filter of size (2× 2). We pass the flattened output of the last CNN layer to a fully connected layer with 256 hidden nodes with ReLu activation function and apply a dropout with 0.3 ratio for regularization.

Finally, we pass the output to the final layer of 15 output nodes and a Sigmoid activation function. Initially, we used binary cross entropy as a loss function optimized with Adadelta and a learning rate initialized to 0.1 with an exponential decay every 1000 iterations. We stopped the training after 10 epochs of no improvement on the validation set and retrieved the model with the best validation loss.

The initial results showed that the model can predict certain contexts fairly well, while others are harder to predict. Table 5.2 gives the performance of the model on the different contexts with standard multi-label classification evaluation metrics [TK07]. We find that certain contexts such as “club", “party", “sad", and “sleep" are easier to predict, while contexts such as “car", “work", and “night" are harder to predict. These results confirm the intuition that certain contexts could be more related to the audio characteristics and hence could be inferred from it, such as energetic dance music for “party" and calming soothing music for “sleep". However, for other contexts, the audio does not appear sufficient and the music style which people tend to listen to in a car or at work seems to widely vary. These contexts would potentially need additional information about the user in order to be predicted correctly in a personalized manner. Additionally, in Figure 5.4, we can see that the model's output has similar co-occurrence patterns between contexts as the original co-occurrences in the dataset. This means that using the audio content, the model was able to learn the similarities between these different contexts.

One drawback of this method is that we do not have explicit negative samples for each label. Hence, it is challenging to fairly evaluate and train the model with missing negative labels. In this work, we mainly focus on the recall because we are confident in the positive labels and would prefer to correctly predict all of them. However, since a classifier that predicts all labels for any given track would give perfect recall, it is important to ensure a balance with the true negative rate and the precision as well. As the missing negative labels are still used in training, they would lead to falsely train the model on false negatives. Missing negative labels is a known problem in the research community that had

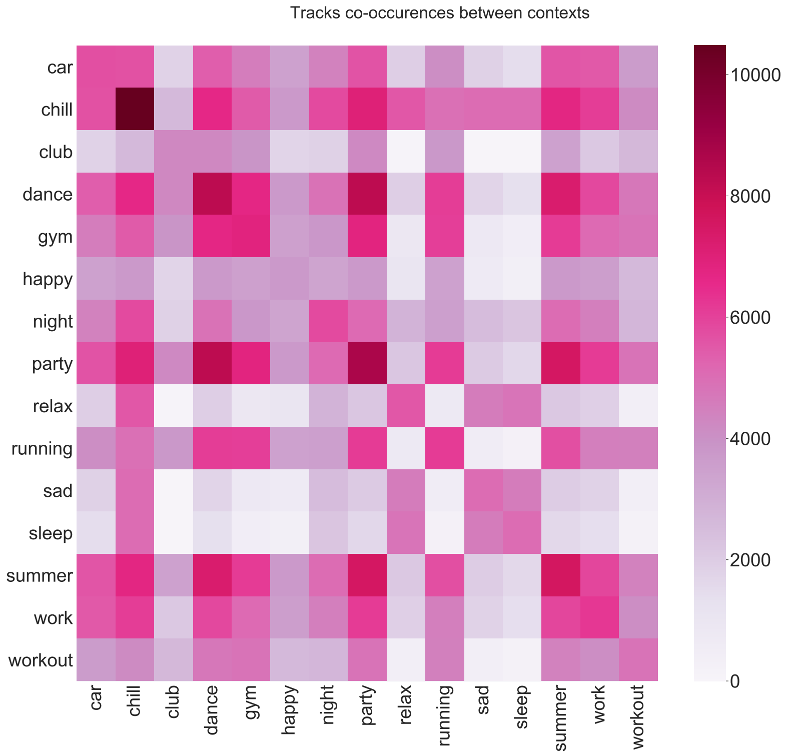


Figure 5.4 – Output co-occurrences between contexts from the trained model

received much attention [WLW+14, CSWZ08, EN08, YJKD14, PTC+, BK14, BJJ11]. To counteract this, we propose to modify the loss function as presented in the next section.

Sample-level Weighted Cross Entropy

We propose to modify the binary cross entropy loss to account for the confidence in the missing labels. This can be done by adding weighting factors to our loss function. Weighting the cross entropy function has been previously proposed in the literature [LGG+17a, PSK+16]. However, to our knowledge none of the previous approaches are applying the weight per sample for each of the positive and negative labels independently. In our proposed approach, we apply a confidence-based weight per sample for each of the positive and negative labels. We hypothesise that using these weights can improve our model performance in predicting the correct label by giving less weight to samples with low confidence in their label.

Formally, let $\mathbb{X} = \mathbb{R}^d$ denote the d-dimensional space for the instances, $\mathbb{Y} = \{0,1\}^m$ denote the label space marking the absence or presence of each of the m context classes for each instance. The task of multi-label classification is to estimate a classifier $f : \mathbb{X} \mapsto \mathbb{Y}$ using the labelled dataset $D = \{(\mathbf{x_i}, \mathbf{y_i}) | 1 \leq i \leq n\}$.

We can describe our classifier as $\hat{\mathbf{y}}_\mathbf{i} = f(\mathbf{x_i}; \theta)$, which tries to estimate the labels $\hat{\mathbf{y}}_i$ for the given sample $\mathbf{x_i}$, while θ represents the trainable parameters of the model. The model parameters are trained by optimizing a loss function $J(D, \theta)$ that describes how the model is performing over the training examples. In multi-label classification, it is common to use the binary cross entropy loss:

$$CE(\mathbf{x_i}, \mathbf{y_i}) = -\sum_{c=1}^{m} y_{i,c} \log(f_c(\mathbf{x_i})) + (1 - y_{i,c}) \log(1 - f_c(\mathbf{x_i})) \tag{5.1}$$

where $y_{i,c}$ is the c^{th} label in $\mathbf{y_i}$ and $f_c(\mathbf{x_i})$ is the output of the f_c classifier corresponding to the c^{th} label.

The cross entropy is made of two terms, one is active when the label is positive while the second is zero, and vice versa. We propose to modify each term to add a weighting factor, one relative to the confidence in the positive label and a second one relative to the expectation of a negative label for each sample.

$$CE_{proposed}(\mathbf{x_i}, \mathbf{y_i}) = -\sum_{c=1}^{m} \omega_{i,c} y_{i,c} \log(f_c(\mathbf{x_i})) + \bar{\omega}_{i,c}(1 - y_{i,c}) \log(1 - f_c(\mathbf{x_i})) \tag{5.2}$$

where $\omega_{i,c}$ represents the confidence in the positive label, while $\bar{\omega}_{i,c}$ represents the confidence in the negative label.

Regarding the case of context classification using audio content, our dataset contains several tracks that were labelled with a certain context based on their appearance in playlists that are relevant to that context. Our confidence in the context class increases as the number of playlists that the track appeared in increases. However, a negative label indicates that the track was not included in any of our selected playlists, which is not a conclusive way to label the track as a negative example for that context. However, we observed that there are certain correlations between contexts. Hence, we propose to use this information in estimating the probability of a true or a false negative label while training our model.

Regarding the negative weights, we define the confidence as:

$$\bar{\omega}_{i,c} = P(y_{i,c} = 0 | \mathbf{y_i}) \tag{5.3}$$

which corresponds to the probability of having a negative label for the c^{th} label given the vector of labels $\mathbf{y_i}$ for the point $\mathbf{x_i}$. This probability can be estimated from the ground-truth label matrix based on labels correlation. When estimating the weight, it is possible to either ignore the zeros in the labels $\mathbf{y_i}$ since we have lower confidence about them, referred to as `ignore zeros`, or we can condition on the whole label vector including the zeros, referred to as `exact match`. We experimented with both negative weight schemes.

Regarding the positive weights, we propose using TF-IDF [R+03]:

$$\omega_{i,c} = \frac{n_{i,c}}{Nc} * \log\left(\frac{m}{\bar{n}_i}\right) \tag{5.4}$$

where $n_{i,c}$ is the number of times track $\mathbf{x_i}$ appeared in playlists from context class y_c. N_c is the total number of tracks that appeared in playlists of context class y_c. $\bar{n}_i$ is the number of context classes x_i is labelled with. The TF-IDF values are naturally very small, hence, we normalize the values with unit-mean unit-variance. We interpret the positive weights as a priority rank to learn predicting important samples first, i.e. the ones with high TF-IDF. Since there are no missing labels in the positive samples, we normalize it to have a mean of 1.

Table 5.3 – Classification results for models trained with different weighting schemes computed with macro averaging

	HL	AUC	Recall	Precision	F1	TN ratio
No Weights	0.32	0.73	0.69	0.66	0.67	0.64
Negative Weights (exact match)	0.32	0.73	0.77	0.63	0.69	0.55
Negative Weights (ignore zeros)	0.39	0.72	0.94	0.56	0.7	0.27
Both Weights (exact match)	0.32	0.73	0.66	0.67	0.66	0.67
Both Weights (ignore zeros)	0.37	0.73	0.91	0.59	0.71	0.33

Evaluation Results

Table 5.3 shows the results of using different weighting schemes for training the model. We observe that using specific weighting schemes improves the results compared to using the non-weighted loss. It leads to a higher recall with a varying drop in the precision and true negative rate. We find that using the negative weights with ignoring the zeros gives the best results in terms of improving the recall, without pushing the model to outputting all ones. Hence, using the zeros when computing the co-occurrences of labels, even if some of them are missing, leads to better estimation of true and missing labels. We also found that using the TF-IDF weighting scheme for the positive samples does not lead to much improvement in the classification results, which is not surprising as there are no missing positive labels in this dataset.

The goal is to have the highest recall with the least drop in precision. However, the missing labels in the ground-truth makes it challenging to objectively evaluate the performance. It is possible that the drop in the precision and true negative rate is due to missing labels in the ground-truth that were regarded as a false prediction while it is a false ground-truth. Our interpretation is that using the weights is useful for correctly predicting more positive samples. However, the balance between the recall and precision is subject to the problem and the use case of the classifier. While the problem of evaluating a model with missing labels is still an open issue, using the sample-level weighting in the loss function seems promising, especially in cases where detecting true positives is prioritized.

Given the shortcomings in evaluating the proposed weighting scheme using a dataset with missing labels, and the potential of such scheme in various other settings, we take a detour to conduct an external evaluation of our method on complete datasets manipulated by us to create artificial missing labels. This evaluation methodology is a standard way in evaluating an approach targeting missing labels. Hence, in the following, we will further extend our study of the proposed weighted loss following the common evaluation procedure targeting missing labels.

5.5 External Evaluation of Confidence-based Weighted Loss for Multi-label Classification with Missing Labels

Before proceeding to evaluating our proposed weighted loss, we first give an overview of the progress in the problem of multi-label classification with missing labels (MLML) and their shortcomings that we address. Multi-label classification [GV15, TKV09] is a common task in various research fields, such as audio auto-tagging [BMEM11], image annotation [BLSB04], text categorization [McC99], and video annotation [QHR+07]. As explained in Chapter 2, multi-label classification is concerned with the problem of predicting multiple correct labels for each input instance. A relevant problem to multi-label classification, which we are facing in our collected dataset, is missing labels. Collecting a multi-label dataset is a challenging and demanding task that is less scalable than collecting a single-label dataset [DRK+14]. This is because collecting a consistent and complete list of labels for every sample requires significant effort.

It is shown in [ZBH+16] that learning with corrupted labels can lead to very poor generalization performances. Various strategies in dataset collection, such as crowdsourcing platforms like Amazon Mechanical Turk [3] or web services like reCAPTCHA [4], lead to datasets with a set of well-labelled positive samples and a set of missing negative labels. The set of missing labels is often not known. Hence, this problem of MLML is different from the problem of partial labels [DMM19], where the position of the missing labels is known but its value is unknown, and noisy labels [Vah17] where a set of both positive and negative labels are corrupted.

Most of the previous approaches relied on exploiting the correlation between labels to predict the missing negative labels [XNH+18, BK14, CSWZ08, WLW+14]. However, the state-of-the-art approaches in MLML [HYG+19, HQZ+19] are not easily usable in cases where a pre-trained model is used. They either rely on jointly learning the correlations between the labels along with the model parameters, require prior extraction of manually engineered features for the task [HYG+19], or assume the location of the missing labels is known but the value is missing [HQZ+19], which are not applicable to our setup. Furthermore, these methods do not allow to fine-tune a pre-trained model on a dataset with missing labels. This is limiting because it has been shown that fine-tuning a pre-trained architecture is useful and, in most cases, gives superior results to models trained from scratch [TSG+16, KSL19]. Multiple domains exploit existing pre-trained models especially when access to large annotated data is challenging, such as medical image classification [ZSZ+17, GBL+18, MCCL+17], or when access to resources and computation power to fully train a complex model are scarce.

Unlike these methods, our proposed approach is scalable and usable to fine-tune a pre-trained model. To train our model, the weighted loss function accounts for the confidence in the labels. Weighted loss functions is a common approach for different problems, e.g. to solve class imbalance [WSBT11], to focus on samples that are harder to predict [LGG+17b], or to solve a similar problem of partial labels [DMM19]. However, to our knowledge, this is the first attempt to use a per sample per label weighted loss for missing labels where the missing labels are unknown.

---

[3] https://www.mturk.com/
[4] https://www.google.com/recaptcha/

Experiments

To validate the advantage of using the weighted cross entropy, we compare between the performance of the same model trained with original cross entropy and with the weighted cross entropy across different ratios of artificially created missing labels. This requires finding multi-label datasets with complete labels, i.e. fully labelled with no missing labels. Given our requirements of large well-labelled multi-label dataset, and the sub-bar state of current audio-related datasets we examined, we seek datasets from the image domain. Specifically, we use the proposed loss on the problem of image classification with multiple labels. Image classification is a popular problem with multiple approaches proposed to it and a vast repertoire of pre-trained models on large datasets. We apply one of the commonly used pre-trained models, which is inception-resnet v2 [SIVA17], on two different large datasets: MSCOCO [LMB+14] and NUS-WIDE [CTH+09]. We use two different schemes for computing the weights:

1. Setting the weights for the missing labels to zero and one otherwise (by using our knowledge of which labels are missing, which is not the case in most real-world datasets) referred to as `ignore missing weighted cross entropy (IM-WCE)`;
2. Estimating them using label co-occurrences as explained earlier, referred to as `correlation-based weighted cross entropy (CB-WCE)`.

Our experiments can be reproduced through our published code [5].

Datasets

The experiment requires a strongly labelled multi-label dataset with no missing labels at the start. Hence, we decided to work with MSCOCO [6] [LMB+14] which is commonly used in multi-label classification with and without missing label [HHSC18, DMM19, LSL17, WYM+16]. The dataset is originally intended for image segmentation, but is also usable for image classification. We use the 2017 version of the dataset. The dataset contains ∼122k images and 80 classes. However, after filtering out the samples with less than 4 labels, the total number of images drops to ∼33k images.

The second dataset is NUS-WIDE [7] [CTH+09], which is another image classification dataset that is suitable for this problem and also commonly used in the multi-label classification studies along with MSCOCO. The dataset contain ∼270k images and 81 classes. However, the number of images drops to ∼24k images after filtering out samples with less than 3 labels per sample. We reduced the threshold to 3 labels for this dataset because it has less labels co-occurrences compared to MSCOCO.

An important part of the experiment is creating missing labels in the training dataset. We propose to create missing labels with different ratios. We follow a similar procedure to [DMM19]. We hide the labels randomly as a ratio of the complete labels per image, i.e. we hide $x_i = r * n_i$ labels for each image, where r is the ratio of labels to hide, n_i is the total number of positive labels of the image i, and x_i is the corresponding number of labels to hide in this image. We use ratios of 0.0, 0.25, 0.5, and 0.75 missing labels to complete labels.

We propose to use a pre-trained classification model for the task of image classification that needs to be fine-tuned to a different dataset with missing labels. Previous papers

---

[5] `https://github.com/karimmibrahim/Sample-level-weighted-loss.git`

[6] `http://cocodataset.org`

[7] `https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUS-WIDE.html`

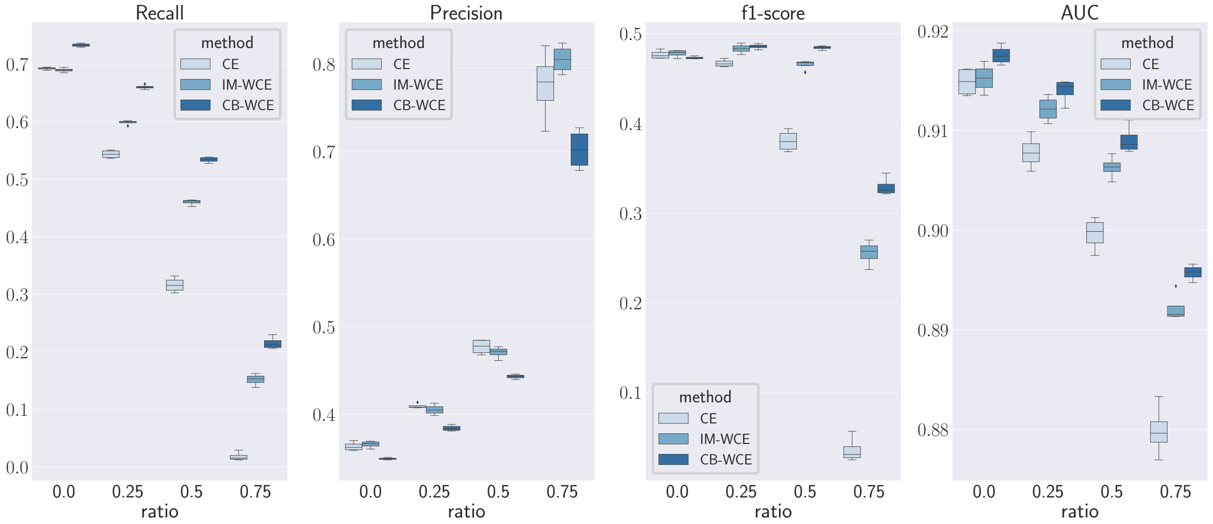


Figure 5.5 – Results of the weighted cross entropy loss and original cross entropy on the MSCOCO dataset with different ratios of missing labels

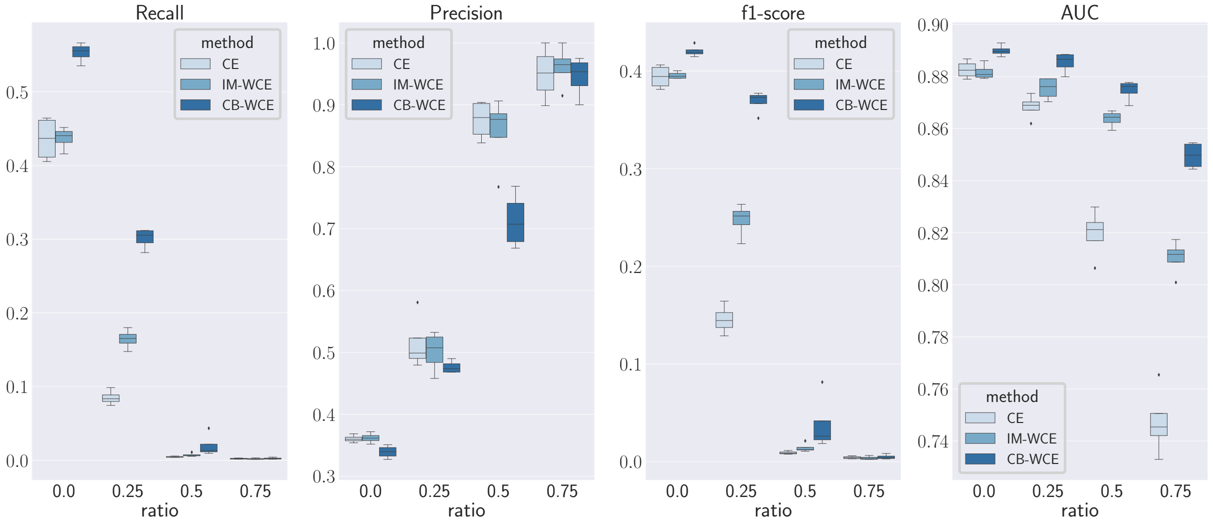


Figure 5.6 – Results of the weighted cross entropy loss and original cross entropy on the NUS-WIDE dataset with different ratios of missing labels

on multi-label classification used models as VGG16 [LSL17] and resnset-101 [DMM19]. The exact architecture of the model is not the focus of this work and would not have a significant effect on the comparison between the two losses. Hence, we used the inception-resnet v2 [SIVA17], which is one of the best performing models in image classification, pretrained on the ImageNet dataset [DDS+09] through the TensorFlow pre-trained models library[8].

We perform a 4-fold cross-validation, each with the aforementioned ratios of missing labels in the training sets and no missing labels in the test sets. We evaluate the model performance using standard multi-label classification metrics: Precision, Recall, f1-score and AUC [ZZ13], all computed with 'micro' averaging to account for the large number of classes with few samples [SL09]. The effect of the missing labels is specifically prominent in predicting the positive labels correctly. As the ratio of missing labels increases, the models learn to predict all zeros. Hence, the selected metrics are useful in evaluating the model's performance particularly in these cases.

---

8. https://github.com/tensorflow/models/tree/master/research/slim

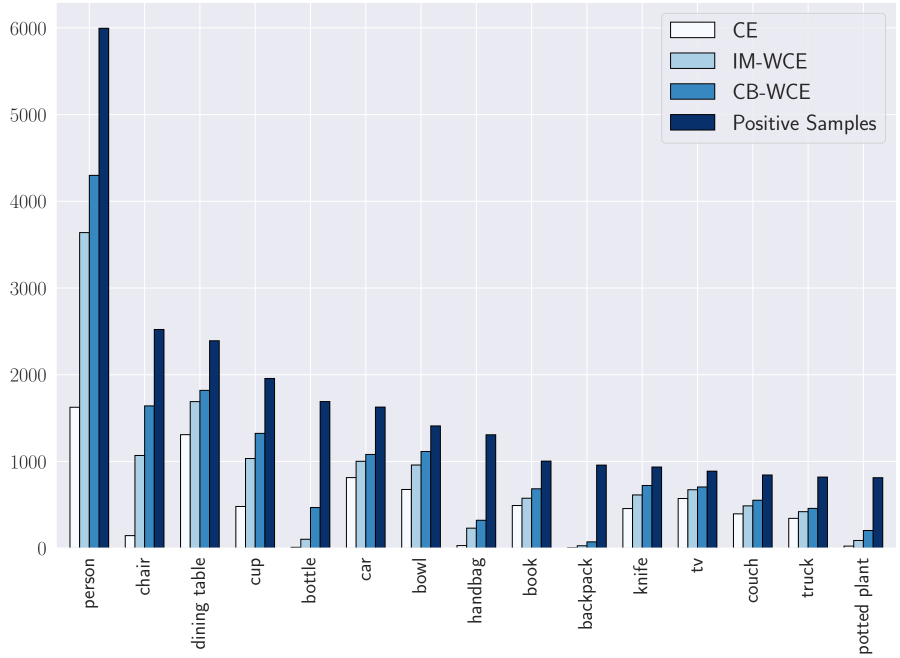


Figure 5.7 – Comparison of the correctly predicted positive samples between the different methods

Evaluation Results

Figure 5.5 shows the results obtained when training with the 3 different losses on the MSCOCO dataset on different ratios of missing labels. It shows that using the weighted loss clearly improves the performance of the model in terms of recall, f1-score and ROC-AUC, with an expected decrease in the precision. The decrease in the precision is explained by the fact that the model trained with the unweighted loss is learning to predict mostly zeros and the few samples that are predicted as positive are more likely to be correct. This is evident when observing the recall and the f1-score results alongside the precision. Additionally, the improvement is larger as the ratio of missing labels increases. We can also notice the effect of missing labels on the performance of the model. The higher the ratio of the missing labels is the worse the model performs. Moreover, we find that using the correlation as weights generally gives better results in all cases even in the case where there are no missing labels. This is understandable since using the correlations in training a multi-label classifier leads the model to learn the underlying relationship between labels [TDS+09].

Similarly in Figure 5.6, we find the results of applying the different loss function on the NUS-WIDE dataset. We find a similar pattern in the performance of the model across different values of missing labels which shows the advantage of using the weighted loss function. However, as NUS-WIDE shows less correlation and co-occurrences between the labels on average compared to MSCOCO, the improvement is less impactful, yet evident.

Additionally, Figure 5.7 shows a comparison of the true positives of each of the methods for the 15 most frequent classes in the MSCOCO dataset with a ratio of 0.5 missing labels. It is evident that the correlation based (CB-WCE) gives superior results in all classes even compared to (IM-WCE). However, the improvement is particularly noticeable in certain classes, such as "bottle" and "chair", which we interpret such that some classes that are harder to learn becomes easier when emphasizing their co-occurrences with other classes.

Considering the evaluation results on these two different datasets, we can advise towards using the weighted loss for multi-label classification when missing labels are present. We experimented with using the correlations to weight the loss function and concluded an evident improvement across different evaluation metrics. While there are various proposed solutions for missing labels, our proposal is particularly more suitable to be used in the cases of fine-tuning a pre-trained model, or even in cases where a specific deep learning architecture is preferred to be used and a simple modification in the loss is needed to account for the missing labels, similar to our original use-case. Hence, while our first evaluation on our audio dataset were not conclusive, our additional evaluations with creating artificial missing labels provided insightful proof of its usability.

5.6 Conclusion

In this chapter, we presented our study on using music auto-taggers for contextual tags. We investigated a new reliable procedure of collecting and labeling a dataset of context-based music tracks, by exploiting playlists titles. We focused on studying 15 context tags that were selected based on the popularity in the Deezer catalogue. We observed patterns in the co-occurrences between contexts which encourage studying the relationship between contexts to disambiguate identical, similar, and different contexts on a larger scale. Additionally, we studied the use of audio content to predict tracks suitability for different contexts.

We trained a convolutional neural network to predict the suitable context for each track. One clear challenge in this problem is the missing negative labels, which affects both the training and evaluation of the model. We proposed using a confidence-based weight in training the model that reflects the probability that the missing label is a positive or negative label. Initial results show an improvement in predicting the true positive samples. However, we faced the challenge of evaluating this methodology with missing negative labels. Hence, we proceeded by testing our proposal on a complete dataset but with artificially created missing labels. The proposed approach shows a clear improvement compared to original unweighted loss.

Finally, given our observations, the next logical step is to consider the user and other information that are available on the tracks and the user. As seen earlier, some contexts are expected to be more uniform in the music style while others vary a lot. We might be able to perform better in tagging tracks with contextual tags if we consider the user profile in the process. Hence, in the following chapter we investigate how to integrate the user information in the auto-tagging procedure.

Chapter 6

Situational Tags and User Dependency

The work presented in this chapter has been published in the following paper:

Ibrahim, K. M., Epure, E. V., Peeters, G., Richard, G. "Should we consider the users in contextual music auto-tagging models?" *Proceedings of the International Society for Music Information Retrieval Conference*, Montreal, Canada, 2020.

6.1 Introduction

Describing tracks with contextual tags provides a mean to improve music exploration and playlist generation in a dynamic way, suitable for the frequent changes in the user context. However, certain tags largely depend on users and their listening preferences, in particular, the tags referring to the situation of music listening such as ‘running’ or ‘relaxing’ [SFU13]. In this chapter, our primary investigation is on the fact that some context classes could be very specific to the user and, subsequently, on how to adapt the music auto-taggers to account for that.

As explained earlier, previous studies showed that the context of the user has a clear influence on the user’s music selection [NH96c, GL11]. Hence, context is a primary focus for music streaming services in order to reach a personalized user experience [KR12]. However, in order to truly achieve a personalized experience,our auto-taggers need to consider each user differently, based on their varying preferences for a given situation. Traditional auto-taggers that rely only on the audio content without considering the case where tags depend on users, are not ideal for describing music with user-dependent tags like contexts. Additionally, their evaluation protocol should be also adapted to account for the different users.

Given the variation in performance we observed in chapter 5, we hypothesize that putting the user in the loop would improve the auto-taggers performance when predicting the right contextual tag of a track. However, we face certain challenges: 1) linking between the audio content, the listening situation, and now the user, while collecting a dataset. 2) Representing the users and integrating their profile in the classification process. 3) Properly evaluating the influence of including the user profile on the tagging quality.

In this chapter, we present the following contributions: 1) a dataset of ∼182K user/track pairs labelled with 10 of the most common context tags based on the users’ contextual preferences (presented in Section 6.2), which we make available for future research; 2) a new evaluation procedure for music tagging which takes into account that tags are subjective, i.e. user-specific (in Section 6.3); 3) an auto-tagging model using both audio content and user information to predict contextual tags (in Section 6.4). Our experiments in Section 6.5 clearly show the advantage of including the user information in predicting contextual tags compared to traditional audio-only-based auto-tagging models.

6.2 Dataset

To properly study the influence of including user information in context auto-tagging models, we need a dataset of tracks labelled with their contextual tags according to different users. For this purpose, we rely on the user-created context-related playlists. As explained, users often create playlists for specific contexts and the titles of these playlists may convey these contexts. Thus, similar to [PZS15] and our work in Chapter 5, we exploit the playlist titles to label tracks with their contextual use. Additionally, we put the users in the loop as playlist creators by explicitly including them in the dataset.

To retrieve contextual playlists, we follow the same procedure using the set of contextual keywords collected earlier. Similarly, to construct our dataset, we selected, out of all collected context-related keywords, 10 which were the most frequent keywords found in the playlist titles in the Deezer catalogue. As an additional filter compared to the balancing step in the previous dataset, we selected the keywords that shared a similar number of playlists to avoid any bias due to the popularity of some contexts, hence, we use 10 tags instead of 15. The contextual tags we finally selected are: *car*, *gym*, *happy*, *night*, *relax*, *running*, *sad*, *summer*, *work*, *workout*.

We collected all the public user playlists that included any of these 10 keywords in the parsed title and applied a series of filtering steps to consolidate the dataset, similar to our earlier procedure. Similarly, we removed all playlists that contained more than 100 tracks, to ensure that the playlists reflected a careful selection of context-related tracks, and not randomly added. We also removed all playlists where a single artist or album made up more than 25% of all tracks in the playlist, to ensure that the playlist was not intended for a specific artist. Finally, to properly study the effect of the user on the contextual use of a track, we only kept the tracks that were selected by at least 5 different users in at least 3 different contexts. Hence, our dataset reflects how user preferences change the contextual use of tracks. Finally, we tagged each sample, the track/user pair, with the contextual tag found in the corresponding playlist title.

6.2.1 Dataset Analysis

In Figure 6.1, by observing the distribution of contextual tags per track/user pairs in the dataset, we noticed that most of the pairs were assigned to a unique contextual tag. Let us remind that the log scale is used and a sample represents a user/track pair labelled with the contextual tags. It appears that the majority of users tended to associate a track with a single context. Out of ∼3 millions samples, ∼2.9 millions are labelled with a single context. Nonetheless, ascertaining if this observation is generally valid requires further empirical investigation. For this study though, and given the distribution of available data, we limited our final dataset to track/user pairs with single context tags, i.e. we excluded users that assigned the same track to multiple contexts.

Observing the distribution of contextual tags per tracks in Figure 6.2, we find that tracks often have multiple contexts associated with them. This shows that the suitability of

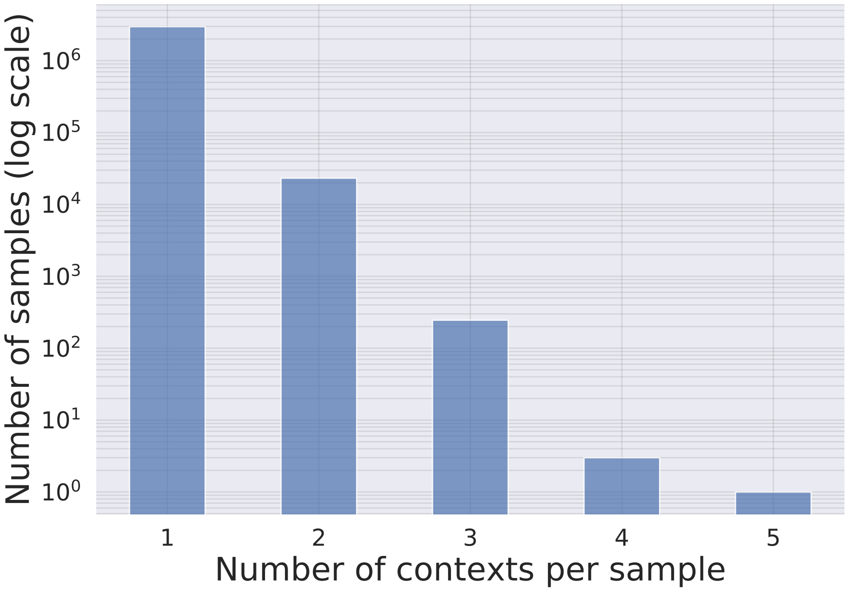


Figure 6.1 – Distribution of the number of contextual tags per sample (user/track pair) in the initial dataset.

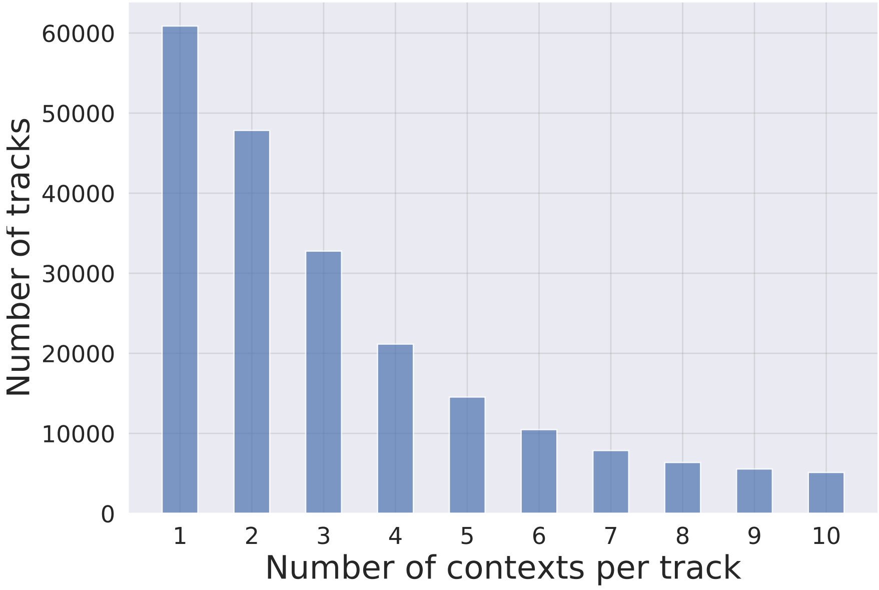


Figure 6.2 – Distribution of the number of contextual tags per track in the initial dataset.

Table 6.1 – Number of samples (track/user pairs), unique tracks and users in the train, validation and test datasets.

	#Samples	#Track	#Users
Train	102K	15K	40K
Validation	30K	4.4K	21K
Test	50K	7.5K	16K

a track for a specific context varies from user to user. However, as previously outlined, given the user, the track is most frequently associated with the same unique context.

The final dataset for this study contains ∼182K samples of user/tracks pairs made of ∼28K unique tracks and ∼75K unique users. We collected the dataset such that each context is equally represented, ensuring a ratio of $\sim \frac{1}{10}$ of all user/track pairs. We split our dataset in an iterative way to keep the balance between classes across subsets, while preventing any overlap between the users and minimising the overlap between tracks in these subsets [STV11]. The distribution of our final split dataset is shown in Table 6.1. The dataset is publicly available to the research community [1]

6.3 Proposed Evaluation Protocol

Previous studies on music auto-tagging [CFS16a, PPNP+18] performed the evaluation in a multi-label classification setup, therefore focusing on assessing the correctness of the tags associated with each track. This is suitable for datasets and tags that are only music-dependent. However, in the case of tags that are also user-dependent, the previous evaluation procedures are limiting. Hence, we further derive an evaluation centered around user satisfaction.

6.3.1 User Satisfaction-focused Evaluation

The purpose of our study is to measure the influence of leveraging the user information on the quality of the prediction of contextual tags. Consequently, we are interested in measuring the potential satisfaction of each user when predicting contexts, instead of relying on a general evaluation approach that could be biased by highly active users or by the popularity of certain tags. Hence, we propose to compute the model performance by considering each user independently. To assess the satisfaction of each user, the evaluation metrics are computed by considering only the contextual tags specific to a user. Then, to assess the overall user satisfaction, we average the per-user results yielded by each model.

Formally, let $\mathbb{U}$ denote a finite set of users in the test set, $G_u = \{0,1\}^{n_u \times m_u}$ denote the groundtruth matrix for user u, n_u is the number of tracks associated with the user u, and m_u is the number of contextual tags employed by the user. Similarly, $P_u = \{0,1\}^{n_u \times m_u}$ denotes the matrix outputted by the model for all active tracks and contextual tags for the given user u. First, we compute each user-aware metric, hereby denoted by S, for a given user u as:

$$S_u = f(G_u, P_u) \tag{6.1}$$

where f is the evaluation function. In our evaluation, we use standard classification metrics such as the area under the receiver operating characteristic curve (AUC), recall,

---

[1] https://doi.org/10.5281/zenodo.3961560

precision, and f1-score [HS15]. While the protocol is defined for the general case of multi-label setting, in our current work, given the dataset, it is applied to the case of single-label. Then, we compute the final metrics, by averaging over all users in the test set:

$$S_{\mathbb{U}} = \frac{1}{N} \sum_{u \in \mathbb{U}} S_u, \text{ where } N = |\mathbb{U}|. \tag{6.2}$$

6.3.2 Multi-label Classification Evaluation

In this work, we develop a system that takes both the audio and the user information as input. As seen in Section 6.2.1, for a given track and user, there is a single groundtruth context to be predicted. The problem is said to be single-label. However, if we want to compare this system with a system that only takes audio as input, we need to consider during training various possible groundtruth contextual tags for a track, each from a different user. Then, the problem becomes multi-label. The comparison of the two systems is therefore not straightforward. Indeed, for the user-agnostic case, we can train a multi-label system, i.e. a system with a set of sigmoid output activations optimized with a sum of binary cross entropy, and estimate it either as single-label by taking the output with the largest likelihood, or as multi-label by selecting all outputs with a likelihood above a fixed threshold. For these reasons, in the current evaluation, we consider the following scenarios:

1. Multi-output / multi-groundtruth (MO-MG): This is the classical multi-label evaluation where the model outputs several predictions and each track is associated with several groundtruths. This evaluation is however independent of the user.
2. Multi-output / single-groundtruth (MO-SG): In this scenario, a model trained as multi-label (such as a user-agnostic model) is still allowed to output several predictions. However, since the groundtruth is associated with a given user, there is a single groundtruth. The obtained results are then over-optimistic because the model has several chances to obtain the correct groundtruth.
3. Single-output / single-groundtruth (SO-SG): this is the case that is directly comparable to our single-output user-aware auto-tagging model. As opposed to the MO-SG scenario, models trained as multi-label are now forced to output a single prediction, the most likely contextual tag. This prevents them from being over-optimistic as they only have one chance to obtain the correct groundtruth, as does the single-label model too.

6.4 Audio+User-based Situational Music Auto-tagger

We propose to build a user-aware auto-tagging system. Given that contextual tags are interpreted differently by different users, we hypothesize that considering the user profile in training a personalized user-aware contextual auto-tagging model may help. For this, we propose to add to the system, along with the audio input, a user input. We study the effectiveness of representing the user via “user embeddings”, obtained from user listening history.

Traditional Audio-based Auto-tagger

We use the same prevalent audio-based auto-tagging model proposed by Choi et al [CFS16a] used earlier. As explained, the model is a multi-layer convolutional neural

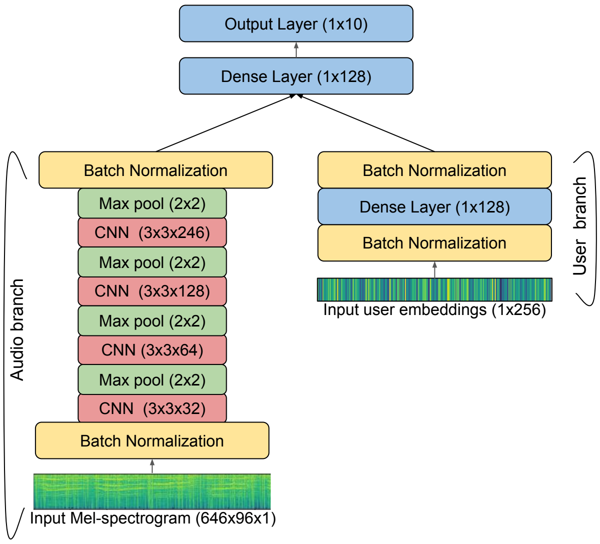


Figure 6.3 – Architecture of the Audio+User-based model.

network. The input to the network is the pre-processed Mel-Spectrogram of the music track. This multi-label classification model predicts, for a given track, the set of all possible tags.

We trained the network with the Mel-spectrogram as an input of size 646 frames x 96 mel bands, which corresponds to the snippet from 30 to 60 seconds for each track. The output is the predictions for the 10 contextual tags. The input Mel-Spectrograms is passed to a batch normalization layer then to 4 pairs of convolutional and max pooling layers. The convolutional layers have a fixed filter size of (3x3) and (32, 64, 128, 256) filters respectively, each followed by a ReLu activation function. The max pooling filters have a size (2x2) each. The flattened output of the last CNN layer is passed to a fully connected layer with 256 hidden units with ReLU activation function. We apply a dropout with 0.3 ratio for regularization. Finally, we pass the output to the final layer of 10 output units each with a Sigmoid activation function. The loss function is the sum of the binary cross entropy optimized with Adadelta and a learning rate initialized to 0.1 with an exponential decay every 1000 iterations. We applied early stopping after 10 epochs in case of no improvement on the validation set, and kept the model with the best validation loss.

Proposed Audio+User-based Auto-tagger

The Audio+User model that we propose is an extension of the Audio-based auto-tagger described above. Our model has two branches, one for the audio input and one for the user embeddings input. The audio branch is identical to the one described above, i.e. 4 pairs of convolutional and max pooling layers with ReLu activation. The input to the user branch is the user embedding of size 256. We apply batch normalization to it followed by a fully connected layer with 128 units and Relu activation. We concatenate the output of the audio branch and the user branch after applying batch normalization to each. We pass the concatenated output to a fully connected layer with 256 hidden units with ReLu activation function and apply a dropout with 0.3 ratio for regularization. The final layer is made of 10 output units with a Softmax activation function. We train the model with minimizing the categorical cross entropy using the same configuration as in the previous model, described in Section 6.4. We present the overall architecture of the complete model in Figure 6.3.

Table 6.2 – Results of the audio-based model (multi-label outputs) on the user-agnostic dataset (multiple groundtruth), MO-MG scenario.

	AUC	Recall	Precision	f1-score
car	0.56	0.96	0.47	0.63
gym	0.71	0.87	0.58	0.7
happy	0.58	0.87	0.37	0.52
night	0.59	0.97	0.48	0.64
relax	0.77	0.8	0.61	0.69
running	0.65	0.91	0.56	0.69
sad	0.77	0.72	0.54	0.61
summer	0.6	0.97	0.61	0.75
work	0.53	0.99	0.47	0.64
workout	0.75	0.84	0.52	0.64
average	0.65	0.89	0.52	0.65

User Embeddings

The user embeddings are computed by applying implicit alternating least squares matrix factorization (ALS) [KBV09, HZKC16] on the users/tracks interactions matrix. The matrix represents the user listening count of the tracks available, with an exponential decay applied based on the lapse since the last listening, i.e. the more recent and frequent a track is listened to, the higher the interaction value. The user embedding is represented as a 256-dimensions vector.

However, the user listening histories are proprietary and represent sensitive data. Additionally, the detailed derivation of the embeddings is an internal procedure at Deezer for the recommendation algorithm. Hence, in order to allow the reproducibility of the current work, we directly release the pre-computed embeddings for the anonymized users present in our dataset.

6.5 Evaluation Results

We evaluate the two models according to the evaluation protocol proposed in Section 6.3. First, we evaluate the audio based model with the 3 scenarios: MO-MG, MO-SG, SO-SG. Then, we evaluate the User+Audio model in the SO-SG scenario. Last, we perform the user satisfaction-based evaluation on both models for the SO-SG scenario. In all evaluation protocols, the metrics were macro-averaged.

Audio-based Multi-output Multi-groundtruth

Table 6.2 shows the results of the audio-based multi-label classification model on our collected dataset without considering the user. The results are consistent with our previous studies on context auto-tagging in chapter 5 on the first collected dataset. They show that certain contexts are easier to predict using only the audio input. These are general contexts with similar music style preferences by different users, e.g. ‘gym’ and ‘party’. By contrast, other contexts are harder to predict from audio only as users listen to more personalized music, e.g. ‘work’ and ‘car’. In consequence, we hypothesize that the variance of the AUC scores across contexts is related to the context dependency on

Table 6.3 – Results of the audio-based model (multi-label outputs) on the user-based dataset (single ground-truth), MO-SG scenario

	AUC	Recall	Precision	f1-score
car	0.54	0.87	0.09	0.17
gym	0.66	0.6	0.18	0.27
happy	0.57	0.67	0.08	0.14
night	0.57	0.6	0.11	0.19
relax	0.74	0.53	0.25	0.34
running	0.6	0.57	0.15	0.23
sad	0.75	0.52	0.21	0.3
summer	0.58	0.78	0.17	0.29
work	0.52	0.55	0.09	0.15
workout	0.71	0.41	0.17	0.24
average	0.62	0.61	0.15	0.23

users. Precisely, some contexts could depend more on users than others, making the latter harder to classify without considering the user information.

Audio-based Multi-output Single-groundtruth

Table 6.3 shows the results of the same audio-based multi-label classification model which we now evaluate considering the user. The same audio track will now be presented several times to the system, i.e. for each user who has annotated this track. While the groundtruth is now single-label and will change for each user, the system will output the same estimated tags independently of the user, i.e. the system does not consider the user as input. We observe a sharp decrease in the precision of the model due to false positive predictions for each user. Indeed, since the output of the system is multi-label, it will output several labels for each track, many of them will not correspond to the current user. The high recall of the model shows that it often predicts the right contextual use for many users. However, it also predicts wrong contexts for many other users. That is due to the limitation of the model which predicts all suitable contexts for all users.

Audio-based Single-output Single-groundtruth

Table 6.4 shows the results of the same audio-based multi-label classification model when restricted to a single prediction per track. While this is not the real-world case of using the audio-based model, it allows a direct comparison to the single-label User+Audio based model. In this case, we see a sharp drop in the recall due to the limitation of a single prediction per track.

Audio+User Single-output Single-groundtruth

Table 6.5 shows the results for the proposed Audio+User model. Comparing these results with the ones presented in Table 6.4, we observe that the model is performing better than the audio-based model for almost all metrics and labels. The f1-score almost doubles when adding the user information. Additionally, for certain labels as *car, happy, running, sad, summer, work*, the influence of adding the user information is obvious compared to all cases of audio-based evaluation when comparing the AUC values. This is consistent with

Table 6.4 – Results of the audio-based model (forced to single-label output) on the user-based dataset (single ground-truth), SO-SG scenario

	AUC	Recall	Precision	f1-score
car	0.54	0	0.03	0.001
gym	0.66	0.44	0.17	0.24
happy	0.57	0	0	0
night	0.57	0.004	0.14	0.007
relax	0.74	0.6	0.23	0.33
running	0.6	0.05	0.15	0.07
sad	0.75	0.003	0.16	0.006
summer	0.58	0.36	0.18	0.25
work	0.52	0	0.2	0
workout	0.71	0.13	0.18	0.15
average	0.62	0.16	0.14	0.11

Table 6.5 – Results of the audio+user model (single-label output) on the user-based dataset (single ground-truth), SO-SG scenario.

	AUC	Recall	Precision	f1-score
car	0.61	0.12	0.13	0.13
gym	0.71	0.16	0.24	0.19
happy	0.64	0.22	0.12	0.16
night	0.61	0.03	0.14	0.05
relax	0.76	0.41	0.29	0.34
running	0.69	0.26	0.22	0.24
sad	0.83	0.5	0.33	0.4
summer	0.65	0.2	0.3	0.24
work	0.58	0.03	0.12	0.04
workout	0.75	0.37	0.2	0.26
average	0.68	0.23	0.21	0.2

our hypothesis that for certain labels the influence of user preferences is much stronger than for other labels.

User Satisfaction-focused Scenario

Finally, we assess the user satisfaction by evaluating the performance of the two models on each user independently. We replace the AUC metric with accuracy because AUC is not defined in the case of certain users where a specific label is positive for all samples. Table 6.6 shows the average performance of each model when computed per user. In this case, we observe how the Audio+User model satisfies the users more on average in terms of all evaluation metrics. By investigating the recall and precision, we noticed that our model results in a larger number of true positives, i.e. predicting the correct context for each user, and a lower number of false positives, i.e. less predictions of the wrong contextual tags for each user. The audio-based model is prone to a higher false positives due to predicting the most probable context for a given track regardless of the user. To sum up, including the user information in the model has successfully proven to improve the estimation of the right contextual usages of tracks.

Table 6.6 – Comparison of user-based evaluation for the two models

	Accuracy	Recall	Precision	f1-score
Audio	0.21	0.204	0.243	0.216
Audio+User	0.254	0.246	0.295	0.26

6.6 Conclusion

Predicting the contextual use of music tracks is a challenging problem with multiple factors affecting the user preferences. Through our study in this chapter, we showed that including the user information, represented as user embeddings based on the listening history, improves the model's capability of predicting the suitable context for a given user and track. This is an important result towards building context-aware recommendation systems that are user-personalized, without requiring the exploitation of extensive user private data such as location tracking [CS14b]. However, there is still large room for improvement to successfully build such systems.

Our current model relies on using the audio content, which is suitable for the cold-start problem of recommending new tracks [VEzD$^+$17, CYJL16]. However, constructing representative user embeddings requires active users in order to properly infer the listening preferences. Future work could investigate the impact of using different types of user information, such as demographics [FS16], which could be suitable for the user cold-start or less active users too.

Additionally, we focused on the case of a single contextual tag for each user and track pair. In practice, a user could listen to the same track in multiple contexts, i.e. tag prediction would be modelled a multi-label classification problem at the user level. Future studies could further investigate this more complex case of adding the user information in the multi-label settings, given a proper procedure for collecting such a dataset.

Finally, while we have proven the advantage of our system on finding personalized suitable tracks for a given listening situation, it can properly be exploited if the listening situation can be also inferred on the run. Hence, in the next chapter, we focus on the possibility of inferring the active listening situation using available device data. Then, we can evaluate our system in a real-world use-case of recommending situational sessions.

Chapter 7

Leveraging Music Auto-tagging and User-Service Interactions for Contextual Music Recommendation

The work presented in this chapter is being prepared for publication in the following paper:

Ibrahim, K. M., Epure, E. V., Peeters, G., Richard, G. "Audio Autotagging as Proxy for Contextual Music Recommendation" *Manuscript in preparation*

7.1 Introduction

So far we explored how to exploit music auto-taggers for the task of predicting the potential listening situation. Additionally, we have managed to extend the auto-tagger model to be user-aware in order to give personalized predictions. However, tagging tracks with their listening situations can only be fully exploited if we can also infer when this situation is actually being experienced by the user. This is particularly challenging task, given the limited information we have about the users compared to the various potential situations to be inferred. Hence, in this chapter we investigate to what extent we can achieve this task given the data we have access to. Furthermore, we join together all the different parts of the system and put it to test in a real-world evaluation scenario. But first, we reiterate on the advantages of approaching the problem in this manner compared to traditional context-aware recommendations.

State-of-the-art contextual recommender systems use various techniques to embed the contextual information in the recommendation process. One of the most common approaches are sequence-based models aiming at predicting the items in the next session using available contextual information [HHM+20, KZL19, BCJ+18]. However, these approaches are lacking in interpretability and serve as a hit-or-miss, with little room for user involvement. Some of the domain-specific approaches, e.g. location-aware [KRS13] or activity-aware [WRW12], could provide interpretable recommendations. However, they are too specific in their range of applicability and do not fit the industrial needs. Interpretable recommendations are increasingly becoming a top priority both for the service and the users [CPC19, GBY+18]. Hence, our goal is to approach the problem in a manner that addresses the need for interpretable recommendations, for any potential listening situation, by exploiting music auto-taggers.

As explained earlier, music is highly complex and is often challenging to be analyzed and described in human readable terms. This missing link between the content of the music and a set of semantic descriptors is referred to as the "semantic gap". There are several ways to bridge this gap in order to extract useful attributes that can describe the music [CS08]. One very common way, which is massively used when searching for music, is the intended use-case, which can be referred to as the listening situation. By observing the user created playlists in online music streaming services, we find thousands made for a specific situation, e.g. an activity such as running. Hence, it would be useful to extract and link such semantic descriptors automatically to the content of the music tracks.

By addressing this semantic gap between the audio content and the intended listening situation, we provide a pathway for interpretable recommendations based on the situation. Previous recommendation systems have been in development to model the several influencing factors simultaneously and provide instant relevant recommendations for all possible sessions, situational or not. While those systems are still in development to reach their full potential in modeling this complex problem, we propose isolating the situational use-cases through a specialized framework to facilitate timely interpretable recommendations. In this sense, our proposed framework does not intend to replace the recommendation systems in development, but to add a layer of additional filtering for the specific case of situational sessions.

In this chapter, we propose a system that aims at disambiguating the listening situation. The listening situation for our system is an activity, location, or time that is influencing the listener's preferences. It is worth noting that we excluded users emotional situations, such as mood, from the scope of this part. This is mainly justified by the significant challenge of predicting a user mood from limited interactions with the service.

The process of music streaming from the perspective of our proposed approach can be found in Figure 7.1. We find that the music service is informed of the users, their track history, plus their past and current interactions with the service, i.e. the device and time data sent during an active session. However, the service is unaware of the influencing listening situation. Our goal is to utilize the available information for the service to infer the listening situation and the suitable tracks for inferred situation. We propose an approach that aims at inferring the potential situations from the user interactions in near real time, while labelling tracks with their potential listening situation in the background using music auto-taggers. Both systems are user-aware, to personalize the predictions of both potential situations and the music preferences in this situation.

In our approach, we give particular attention to the compatibility with the existing streaming services and their recommender systems, by seeking an ad-hoc approach to filter the pool of potential tracks which can be swiftly pushed into production. We seek to develop a system that provides the following features: 1) The listening situation is predicted in short time. 2) The predictions are user personalized. 3) The system makes use of only basic data available in online streaming sessions. 4) It uses proactive user verification, to verify the current situation of the user. The motivation behind these features can be attributed to the need for fast continuous recommendations. Furthermore, pushing the recommendations to the user (proactivity) while leaving the last step of activation to the user would significantly reduce ill-timed recommendations by the system.

We achieve this by splitting the problem in two main blocks: 1) a slow computationally-intensive auto-tagging of music with its intended situational use. 2) A fast and light situation prediction to rank the potential situations for a given user based on the transmitted data from the device to the service. This modular design allows for independent progression/personalization in each of the framework modules as seen fit by the service.

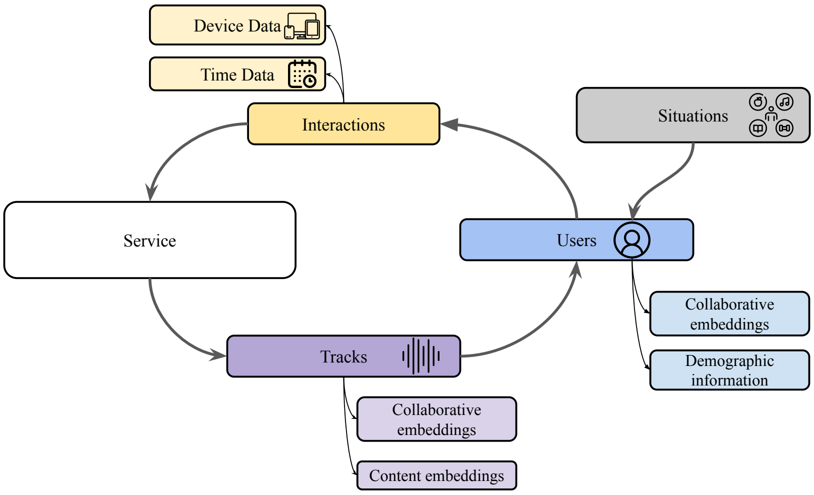


Figure 7.1 – The available data to online music streaming services. The service is informed of the users, their interactions, and their track history. However, the service is unaware of the influencing listening situation.

Finally, existing and future recommender systems can be deployed on top of our framework but on the reduced tracks pool.

Our contributions in this chapter are 1) a framework design to filter tracks with their intended listening situation using a specialized auto-tagger 2) a large dataset of tracks, device data, and user embeddings labelled with their situational use through a rigorous labelling pipeline, which is also made public for future research 3) an extended evaluation of each of the framework blocks, in addition to the complete system in a recommendation setting.

7.2 Proposed Framework

Our proposed approach satisfies the previously mentioned requirements through a two-branch design that achieves two objectives: 1) The user-aware music auto-tagger trained to tag a given user/track pair with a situational-use tag, as presented in Chapter 6. 2) A user-aware situation predictor to rank the potential situations for a given user based on the transmitted data from the device to the service. The overall architecture of the proposed system is shown in Figure 7.2. The intended use-case of this framework is as follows:

1. Background computation: the music auto-tagger would be running to auto-tag every track with its situational use for every given user. For efficiency, the auto-tagger could run only on a subset of the tracks for a given user, potentially the tracks in the given user library.
2. Real-time computation: the situation predictor is continuously running to predict the potential situation for each user when they access the suggestions section. The situation predictor makes available the top K predicted situations to those active users.
3. Retrieval: if the user selects one of the top K presented situations, a playlist made of tracks tagged with the selected situation according to the given user is played to this user. The track selection can be further achieved using traditional sequence-based retrieval methods applied on the reduced track pool, i.e. the tagged tracks.

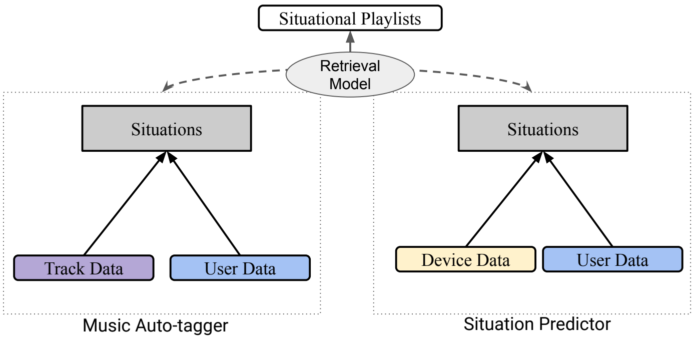


Figure 7.2 – Overview of the framework to generate a situational playlist. The left side (Music Auto-tagger) tags each track/user pair with a situational tag. The right side (Situation Predictor) ranks the potential situations for device/user pair to be presented to the user.

Given the related work explained in Part I and the various categories to describe a recommendation system, we place our proposed approach as a user-aware content-based proactive situational system. User-awareness means the system provides personalized recommendations for each user. Content-based means it uses the audio content of the track to label it. Proactive means the system is continuously predicting the user situation and pushing the suggestions to the user. However, in our system, the activation is left to the user. Finally, situational means the system is centered around giving recommendations related to the listening situation. While the listening situation is not always the dominant factor influencing the user preferences, our system makes available a “pathway” for the user when the situation is the main influence.

7.2.1 User-aware Situational Music Auto-tagger

The first branch in the system is responsible for predicting the intended listening situation for a track according to a given user. As described earlier, music auto-taggers are a useful tool for labeling music with human-readable “semantic” tags which are trained to automatically predict the correct tags (genres/instruments/moods, etc.) by using the audio content, but given a well-labeled dataset.

Our auto-tagger, presented in Chapter 6, is trained to tag tracks with potential situations, during which, a given user is likely to listen to this track. We will reiterate on our implementation of this system in Section 7.3.2, after covering the dataset.

7.2.2 User-aware Situation Predictor

The second branch in the system is responsible for predicting the top K potential situations of a given user using the data sent by the user during an active session. The situation predictor is the branch that needs to perform in real-time to allow the whole system to be real-time. Additionally, in our implementation we relied on using only data available during an online streaming session and sent by the users. Similarly, our selected implementation will be elaborated in section 7.3.2 after investigating available data.

7.2.3 Training Data

Finding available data is one of the most challenging tasks in the domain of context-aware recommendations. As described earlier, the influencing situation is a latent variable that we can only infer from the observed data sent by the user. Hence, in collecting a well-labelled dataset, we need a carefully designed procedure for data collection to ensure high quality groundtruth labels.

Dataset Collection Procedure

We rely on a modified version of the previously used method for labelling streaming sessions with a situational tag by using playlist titles. As seen earlier, users create playlists with a common theme according to their uses. One common theme of these playlists is the listening situation. In various cases, the title of the playlist clearly states the situation by including a “keyword” describing this situation. However, in this setup we rely on the users who actively listened to these playlists instead of the creators which were used in the previous chapter. This allows us to also retrieve the device data of those users while they were actively listening to the playlist.

We retrieve all the listening streams that came from within these playlists at Deezer. Each stream includes: a track, a user, a playlist with a situational keyword in the title, along with the device data sent during this stream. We tag each stream sample, the track/user/device data, with the situation tag found in the corresponding playlist title. This step results in a collection of these streams tagged with the corresponding situational keyword from the title of the source playlist, which can be used in training our system. Note that a track/user/device triplet has a joint tag, none of them are tagged individually. The playlist titles, while prone to errors and false positives, provide an appropriate proxy for labelling those listening streams that is consistent with similar previous work [PZS15, IEPR20]. We further analyse the collected labels in section 7.2.3 for a sanity check.

Collected Dataset

For our experiments, we collected a dataset using the previous procedure. However, we needed to consider certain factors in this subset to ensure high quality data. First, we selected the month of August 2019 for inspection, because this period had more stable use patterns, before the Covid-19 pandemic. Additionally, we had access to data from two locations: France and Brazil. These two locations were provided because they have the most active users in Deezer, while being in two distinctive time zones and seasons. This allows us to perform our study on diverse data with different sources and patterns.

The number of relevant situations to study is not defined. Hence, we collected 3 different subsets with an increasing number of situations. This also allows us to observe how the system performs as the complexity of the problem increases. The situations were selected given their frequency, counted as the number of corresponding playlists in the service’s catalogue. Our subsets are split into: 4, 8, and 12 situations. Each split was sampled such that, there is an equal number of streams for each situation. The number of streams/unique users/unique tracks found in each subset is shown in Table 7.1. We will be using those situations as independent tags without merging similar activities and places. This is due to the challenge in defining and measuring this similarity such that we can confidently assume them identical. The set of situations selected in our dataset are (constituting the 4, 8, 12 situations respectively):

`work, gym, party, sleep | morning, run, night, dance | car, train, relax, club`

Finally, once all the streams are retrieved, we proceed to fetch and compute the data needed to train our model. As described earlier, our system makes use of 3 different types of data:

Table 7.1 – Number of streams, users, and tracks in each of the 3 data subsets with 4, 8, and 12 situations

Situations	Streams	Users	Tracks
4	334K	14K	33K
8	635K	28K	56K
12	886K	79K	62K

1. Track Data: For each track included in one of the streams, we retrieve a 30 seconds snippet of the track, corresponding to the snippet made available by the API from Deezer for reproduciblilty purposes. As we will be using a Mel-Spectrograms based auto-tagger, we compute the Mel-Spectrograms of these snippets as 96 mel bands × 646 frames.
2. Device data: we collect only basic data sent by the device to the service and selected only what we deemed relevant to the situation prediction. The data is: the time stamp (in local time), day of the week, device used (mobile/desktop/tablet), and network used (mobile/WiFi/LAN/plane). Additionally, we extend the time/day data with circular representation of the time and day similar to the one used in [HRS10]. The final feature vector representing device data is made of 8 features: `device-type`, `network-type`, `linear-time`, `linear-day`, `circular-time X`, `circular-time Y`, `circular-day X`, `circular-day Y`.
3. User data: representing the users can be achieved through various versatile techniques. Consistent with our previous criteria, we choose to represent the users using the basic data available during streaming. However, given the nature of each of our branches, the user is represented differently in each branch:
   (a) Music Auto-tagger branch: Similar to our previous work on user-aware music auto-tagging, we used the users' listening history to derive user embeddings that encode their listening preferences. We compute these embeddings through matrix factorization of the user/track interactions matrix to compute a 128-d embedding vector per user [LS99]. The constructed matrix uses all the tracks available in the catalogue to model the user preferences, i.e. it is not computed exclusively to tracks included in our dataset. The computed embeddings is published with the dataset for reproducibility.
   (b) Situation predictor: we used the basic demographic data of the user recorded during registration. For each user, this data is composed of:
   `|age, country, gender|`. While this data is prone to errors and short from fully representing the users, it is consistent with our requirements of using basic always-available data. Additionally, demographics were shown to be reflective of the users' listening habits [KSKR17].

Finally, we split our dataset in an iterative way [STV11], to keep the balance between classes, into 4 subsets, which is used for a 4-fold cross-validation. These splits will be conditioned to either have no overlap between the users, between the tracks, or allow overlaps, which will be further employed for our evaluation protocols. The anonymized dataset, along with the splits, is published for future research[1]. The dataset includes the tracks, device data, anonymized users' data, computed user embeddings and the joint situational tag for each stream.

---

[1] https://zenodo.org/record/5552288

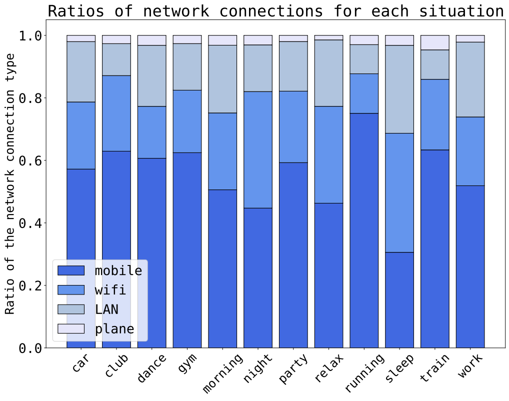


Figure 7.3 – Distribution of used network for different situations

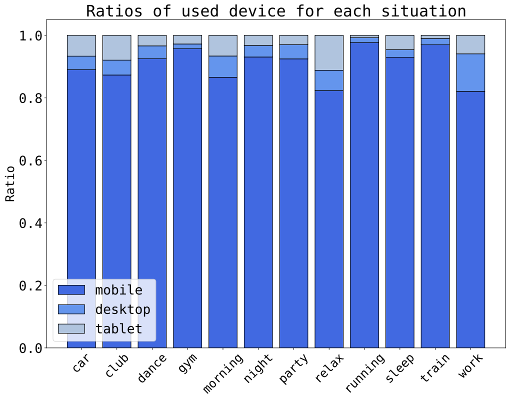


Figure 7.4 – Distribution of used device for different situations

Dataset Analysis

As a sanity check on the collected data, we plot the distribution of the situations across the different features. Figure 7.3 shows the ratio of the used network to connect to the service for all situations. The network connection can be: 1) “`mobile`”: a connection through cellular data, 2) “`wifi`”: a connection with a WiFi network, 3) “`LAN`”: a connection through wired Ethernet, or 4) “`plane`”: an offline stream from a device without a connection. We observe variations that correspond to what is expected from each situation, i.e. outdoors vs. indoors. However, we also find certain networks that do not match the expectations, e.g. LAN connections in a car situation, which represents noise in the dataset that can be a continuation of already existing sessions that moved indoors. Figure 7.4 represents the used device for each situation. The available device are “`mobile`”, “`desktop`” (e.g. a laptop), or “`tablet`”. We notice that most users overwhelmingly use mobile device in most cases, with small variations that also match expectations of indoor and outdoor situations. Finally, Figure 7.5 shows the distribution of all situations for each hour of the day. Similarly, we find predictable patterns for each situation ranging from night-related situation in the early hours that gradually progress as the time pass. These patterns support the hypothesis of using playlist titles as proxy for inferring the actual listening situation.

7.3 Problem Formulation

For simplicity, we consider the case of a single user. For a given user u, we define $\mathbf{e_u} \in \mathbb{R}^{128}$ as the user embeddings retrieved from the matrix factorization of the user-track matrix. $\mathbf{g_u} \in \mathbb{R}^3$ is the demographic data of the user, representing `|age,country,gender|`. At time t, we define the data received from the user's device as $\mathbf{d_u^{(t)}} \in \mathbb{R}^8$. For an audio track a, we define $\mathbf{e_a} \in \mathbb{R}^{256}$ as the track embeddings from applying a convolutional auto-tagger on the melspectrogram of track $\mathbf{A} \in \mathbb{R}^{96\times646}$. Finally, we define the user's listening situation $c \in C$, where $|C| = N$ is the number of predefined situations in the collected dataset. A summary of the used notation can be found in Table 7.2.

The output of the Situation Predictor model is a vector of probabilities $\in \mathbb{R}^N$, one for each situation in C given the user and the device data, such that

$$\mathbf{\hat{y}_{sp}^{(t)}} = P(c|\mathbf{d_u^{(t)}}, \mathbf{g_u}) \quad \forall c \in C \tag{7.1}$$

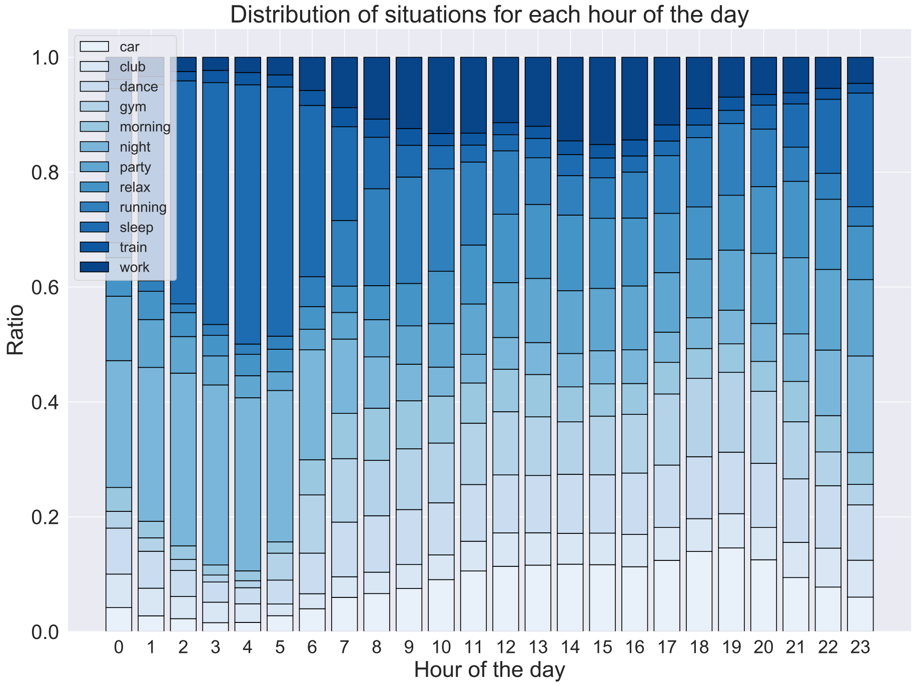


Figure 7.5 – Ratios of the different situations at each hour of the day

Symbol	Definition	Dimension
$\mathbf{e_u}$	The embeddings of an input user	$\mathbb{R}^{128}$
$\mathbf{g_u}$	The demographics of an input user	$\mathbb{R}^{3}$
$\mathbf{d_u}$	The device data of the input user	$\mathbb{R}^{8}$
$\mathbf{A}$	The Melspectrogram of an input track	$\mathbb{R}^{96\times646}$
$\mathbf{e_a}$	The track embeddings from the audio branch	$\mathbb{R}^{256}$

Table 7.2 – Summary of the used notation

The output of the user-aware music auto-tagger is again a vector of probabilities.

$$\hat{\mathbf{y}}_{\mathbf{at}} = P(c|\mathbf{e_a}, \mathbf{e_u}) \quad \forall c \in C \tag{7.2}$$

7.3.1 Inference

Background Computation:

By running the music auto-tagger on all the tracks for a given user to predict the probabilities of all the potential situations, we get a probabilities matrix $\hat{\mathbf{Y}}_{at} \in [0,1]^{N \times M}$, where N is the number of predefined situations and M is the number of tracks available. $\hat{\mathbf{Y}}_{at}$ is computed once and saved for the retrieval phase later.

Real-time Computation

Once a user interacts with the service and sends device data at time t, the situation predictor runs to predict the potential situations using $\mathbf{d_u^{(t)}}$ and $\mathbf{g_u}$. The predicted situation is the one with the highest probability. Let $n^* = \arg\max(\hat{\mathbf{y}}_{\mathbf{sp}})$, be the index of predicted situation.

Retrieval

Once the predicted situation is computed, we retrieve the tracks' probabilities of the predicted situation from $\hat{\mathbf{Y}}_{at}[n^*, :]$. Afterwards, the simplest approach would be to sort those tracks with this probability of fitting the predicted situation, and the top tracks are retrieved. Ideally, we would use those predictions to prefilter the tracks pool, then employ further recommendation algorithms to select the suitable tracks from the reduced pool of tracks.

7.3.2 Training

For a given user u, let the dataset of streams consisting of a sequence of track-device pairs be $< (\mathbf{A_u^{(1)}}, \mathbf{d_u^{(1)}}), \ldots, (\mathbf{A_u^{(K)}}, \mathbf{d_u^{(K)}}) >$, where K is the total number of streams for user u, where $\mathbf{A_u^{(i)}}$ is the melspectrogram of the streamed track at the i^{th} sample for $i \in \{1, \ldots, K\}$, $\mathbf{d_u^{(i)}}$ is the corresponding device data, and $\mathbf{Y} = \{0,1\}^{N \times K}$ is the corresponding groundtruth situations.

We can describe the situation predictor as $\hat{\mathbf{y}}_{\mathbf{sp}}^{\mathbf{(i)}} = f_{\theta_{sp}}(\mathbf{d_u^{(i)}}, \mathbf{g_u})$, which tries to estimate the probabilities $\hat{\mathbf{y}}_{\mathbf{sp}}^{\mathbf{(i)}}$ for the given sample $\mathbf{d_u^{(i)}}$, while $\theta_{\mathbf{sp}}$ represents the trainable parameters of the model. The model parameters are trained by optimizing a loss function $\mathcal{L}(\hat{\mathbf{y}}_{\mathbf{sp}}^{\mathbf{(i)}}, \mathbf{y^{(i)}}, \theta_{\mathbf{sp}})$. Given that the dataset is a single-label-multi-class set, we optimize after applying the soft-max function for all samples i.

Similarly, the music auto-tagger can be described as $\hat{\mathbf{y}}_{\mathbf{at}}^{\mathbf{(i)}} = f_{\theta_{at}}(\mathbf{A_u^{(i)}}, \mathbf{e_u})$, which tries to estimate the probabilities $\hat{y}_{at}^{(i)}$ for the given streamed track using its melspectrogram $\mathbf{A_u^{(i)}}$ as input, while $\theta_{\mathbf{at}}$ is the trainable parameters of the model. Similarly, the model is trained by minimizing the loss after applying soft-max $\mathcal{L}(\hat{\mathbf{y}}_{\mathbf{at}}^{\mathbf{(i)}}, \mathbf{y^{(i)}}, \theta_{\mathbf{at}})$ for all samples i.

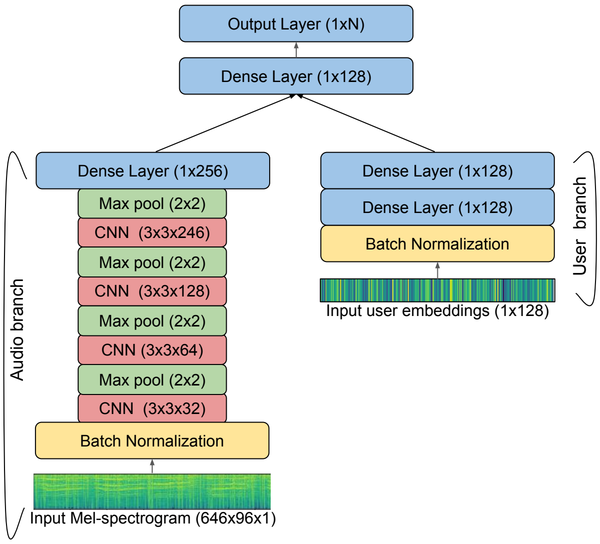


Figure 7.6 – Architecture of the User-Aware Music Auto-tagger.

Framework Implementation

The modular architecture of this framework allows for an effortless mix-and-match according to computation/data availability. In this section, we present the implementation we selected to present the experimentation results.

User-Aware Music Auto-tagger

We use almost the same architecture used in the previous chapter with few variations in the dimensions. The model operates on both the track and user data. The audio input is processed through a multi-layer convolutional neural network applied to its Mel-Spectrograms. The input Mel-Spectrograms is passed to a batch normalization layer then to 4 pairs of convolutional and max pooling layers. The convolutional layers have a fixed filter size of (3x3) and (32, 64, 128, 256) filters respectively, each followed by a ReLu activation function. The max pooling filters have a size (2x2) each. The flattened output of the last CNN layer is passed to a fully connected layer with 256 units with ReLU activation function. The output of the audio module $\mathbf{e_a}$ is a 256-d vector representing the extracted features from the input track. The full architecture of the used model can be found in Figure 7.6.

The user input is processed through a 2-layer feed-forward neural network with ReLu activations before being concatenated with the audio representation. We pass the concatenated output to a fully connected layer with with ReLu activation function and apply a dropout with 0.3 ratio for regularization. The final layer is made of N output units with a Softmax activation function, where N depends on the number of tags available in the selected subset. We train the model till convergence by minimizing the categorical cross entropy, optimized with Adam [KB14] and a learning rate initialized to 0.1 with an exponential decay every 1000 iterations.

Situation Predictor

In choosing a real-time “light” Situation Predictor model, we prioritize the computational complexity requirements over accuracy. The low dimensional input features already provide a strong case for the investigated models. For our implementation, we experimented with different classifiers: Decision Trees, K-Nearest Neighbors, and eXtreme Gradient Boosting (XGBoost) [CG16]. While all gave comparable results, we selected XGBoost to present for its consistent performance across splits and different evaluation scenarios. The selected model takes as input an 11-d feature vector (8 device features + 3 demographic features). Similar to the auto-tagger model, the output predictions depend on the number of situations in the dataset.

Reproducibility

The framework implementation along with the experimental results can be reproduced using the public sourcecode published with this thesis [2]. Additionally, the dataset, along with the splitting information, is shared both for reproducibility and for future studies. The dataset is composed of the track, user, and device triplets along with the corresponding groundtruth situation label. The precomputed and anonymized user embeddings, along with their demographic data, are also shared in the dataset. The 30-seconds track samples used for training and testing in our experiments are retrievable through the Deezer API [3].

7.4 Experimental Results

In this section, we first present our protocols to evaluate the system in different use cases and then discuss the experimental results obtained.

7.4.1 Evaluation Protocols

We approach the evaluation of this system from two different perspectives: 1) evaluating the system intelligence and its capability of learning and generalizing 2) evaluating the proposed system in a stable use-case with frequent users/tracks 3) evaluating the system in an actual retrieval task.

We simulate these scenarios through a different split criteria for the testset. Let the full set of streams in our collected dataset S, where each stream s has a user u and a track t. We will be referring to the training set as S_{train}, the testset as S_{test}, the set of unique users in training and testing as U_{train} and U_{test} respectively, and similarly the unique tracks in the splits as T_{train} and T_{test}.

In our evaluation, we use standard classification metrics such as accuracy, area under the receiver operating characteristic curve (AUC), precision, recall, and f1-score [HS15]. Additionally, as we will be evaluating the situation predictor capability of including the correct situation in the top k predictions, we use accuracy@k. It is important to emphasize that our splits have an equal number of samples for each situation, i.e. there is no bias due to skewness in the labels distribution.

Evaluating System Generalization

This evaluation aims at testing the Auto-taggers capability of extracting relevant features from the audio content and using it to predict its situational use for a given user, while

---

[2] `https://github.com/KarimMibrahim/Situational_Session_Generator`

[3] `https://developers.deezer.com/api`

personalizing the predictions based on the user listening history. The auto-tagger is evaluated on the case of either new tracks or new users to assess its generalization on new cases. Additionally, this evaluation assess the Situation Predictor's capability of learning the patterns of the users and how this knowledge generalizes to new users with new patterns. Note, we consider "new" to be unseen to the models during training but not new to the service, i.e. the users embeddings from their listening history are available during testing.

To evaluate the system intelligence and fit to the data, we restrict the evaluation splits to include either: 1) new users (cold-user case): $S_{test} = \{s|u \notin U_{train}, t \in T_{train}\}$ 2) new tracks (cold-track case): $S_{test} = \{s|u \in U_{train}, t \notin T_{train}\}$. We exclude the specific case of both new tracks and new users because splitting the data with only new user/track pairs in the testset is difficult and rare to find. Additionally, recommending a new track to a new user is not a common nor practical scenario to use for evaluating a system.

Evaluating the System for Session Generation

In this scenario, the goal is to evaluate how the system would perform in a regular use-case as described in section 7.2 (warm case): $S_{test} = \{s|u \in U_{train}, t \in T_{train}, s \notin S_{train}\}$. The regular use-case does not restrict the system to neither new users nor tracks. However, the testset is split to include exclusively new streams, i.e. (user/track) pairs. The evaluation of this regular use-case is relatively complex and includes several entwined evaluation criteria. The goal is to compare the overlap of generated sessions with groundtruth sessions.

We performs those evaluation protocols through:

1. Evaluating the auto-taggers accuracy in predicting the correct situation for a given track/user pair.
2. Evaluating the situation predictor at finding the correct situation in the top K ranked situations.
3. Evaluating the joint system through the overlap of correct predictions between the two branches for every stream in the testset. This accuracy can be interpreted as the ratio of existing streams that would have occurred in these sessions if the playlists were generated with this system instead.

Evaluating the System in a Recommendation Scenario

Finally, we evaluate our proposed approach in a recommendation scenario; we specifically only evaluate the Music Auto-Tagger Part and consider an ideal Situation Predictor. We start by explaining what is the recommendation scenario. For this we first explain the concept of a "session". A "session" is a continuous time span, i.e. without any breaks longer than 20 minutes, during which the user listens to a streaming service. It is defined by user and device information and the sequence of tracks the user has listened to. A "situational session" is a session, for which, tracks are coming from a playlist that is associated with a situation, such as "gym", using the method described earlier. The goal of a recommendation system is to predict the tracks of the next session given a description of previous sessions (user, device and list of tracks). For example, knowing the content of the sessions Z_1, Z_2 predict the tracks for the session Z_3 , therefore for the same user but at a later time. For this, a recommendation system will select a subset of tracks from a music catalogue to be part of the next session Z_3.

In our experiment, we will highlight the fact that pre-filtering this catalogue to the subset of tracks tagged with the same situation as Z_3 improves the recommendation, i.e. we consider only situational sessions as target. For this, we will estimate the situation of all pairs of track/user in the catalogue using our Music Auto-Tagging system. Knowing the situation of the next session Z_3 (we here consider that our Situation Predictor provides the correct situation), we can pre-filter the catalogue, i.e. only retain the subset of tracks with the predicted situation equal to "gym" for this user. To get the tracks of the next session, we then run the recommendation system either on the whole catalogue or our pre-filtered subset of tracks. We describe below the recommendation system (CoseRNN) we use. Given the list of tracks of the ground-truth next session Z_3, and the ones obtained by the recommendation system (a ranked list cut at K=10, 20, . . . , 100) we can compare both lists using Recall (the number of tracks in the ground-truth N list which also exist in the recommended list divided by N), Precision (same divided by K) and F-measure.

CoseRNN. The recommendation algorithm we used is the CoseRNN (Contextual and Sequential Recurrent Neural Network), a state-of-the-art model recently proposed [HHM+20]. CoseRNN aims at predicting the tracks of the next session Z_3 given information of the previous sessions (Z_1, Z_2). Each session is described by its user, list of tracks and device data. The tracks are embedded (using a modification of the Word2Vec algorithm) and the average of the embeddings of the tracks of a session represents the session embedding. The sequence of previous session embeddings is fed with user and device information as inputs to CoseRNN whose output is a new session embedding (the embedding of the next session). Tracks with the closest embeddings to this predicted session embedding are then selected for the next session. The closeness is defined by a cosine distance. The results are therefore a ranked list of tracks. For our experiment, we train this model using all our available sessions (whether situational or not). However, we only test the model on the situational sessions.

7.4.2 Evaluation Results

User-aware Music Auto-tagger Evaluation

The results for the music auto-tagger can be found in Table 7.3. As shown, the model can reach satisfying performance relative to the evaluation scenario. In terms of AUC, the model's fit for both new users and tracks in the cold user/track splits is not significantly impaired compared to the warm case. The performance decreases evidently as the problem gets harder with more situations to tag, though in some cases it increases given the increase of dataset size from additional situations. In terms of accuracy, the model's performance in the intended use-case, i.e. warm case, is satisfying. That is to say, the system can correctly tag around two thirds of the user/track listen streams with their correct situational use, when it has seen the user or the track before, but not jointly.

Note that this accuracy was computed by selecting the most probable situation from the predictions. While the high values of AUC suggest a threshold optimization is needed for each class, in real use-case we do not necessarily need a threshold. The prediction probability could be used directly to retrieve tracks, e.g. by ranking tracks with the prediction probabilities and include top ranked tracks in the generated sessions. However, this max-probability threshold is needed for further evaluations with the situation predictor and with the sequential retrieval model.

A more detailed per-situation evaluation is shown in Figure 7.7. We observe a varying performance across the different situations, with some being easier to predict than others. To further understand these variations, we plot the confusion matrix between the situations in Figure 7.8. The confusion matrix explain the variation in the results. Certain situations that are misclassified are often confused with similar situations, e.g. situation

Table 7.3 – Evaluation results of the Music Auto-tagger evaluated with AUC and accuracy in the three evaluation protocol splits (cold-user, cold-track, and warm case) and the three subsets of situations (4, 8, and 12). The results are shown as mean(std.).

# Sit	AUC			Accuracy		
	Cold User	Cold Track	Warm Case	Cold User	Cold Track	Warm Case
4	0.889 (.009)	0.873 (.013)	0.959 (.013)	69.72 (1.07)	63.77 (2.33)	83.75 (2.33)
8	0.815 (.005)	0.866 (.007)	0.945 (.007)	47.56 (0.53)	52.44 (2.31)	70.81 (1.45)
12	0.852 (.004)	0.824 (.012)	0.941 (.012)	52.68 (1.25)	37.61 (3.47)	69.14 (3.79)

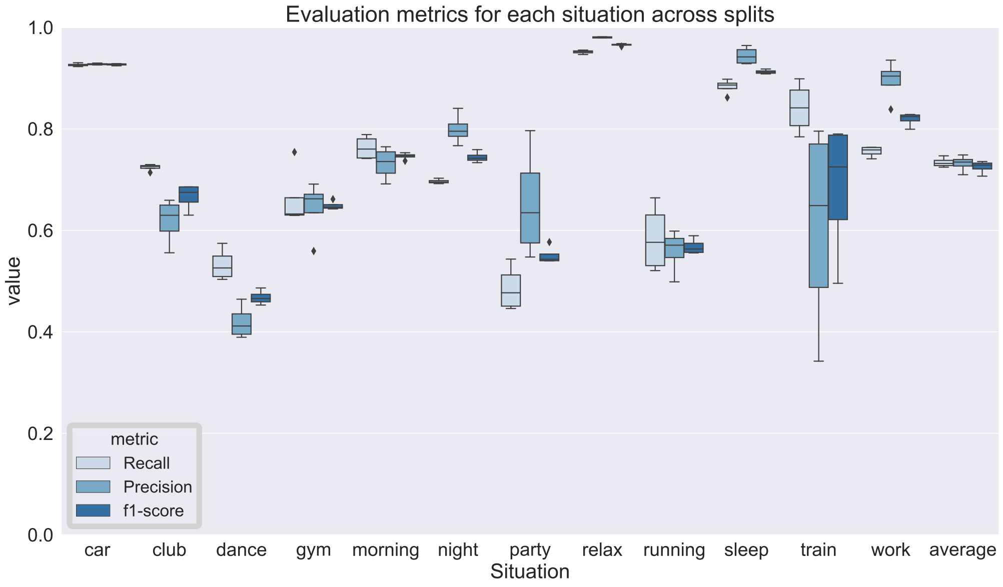


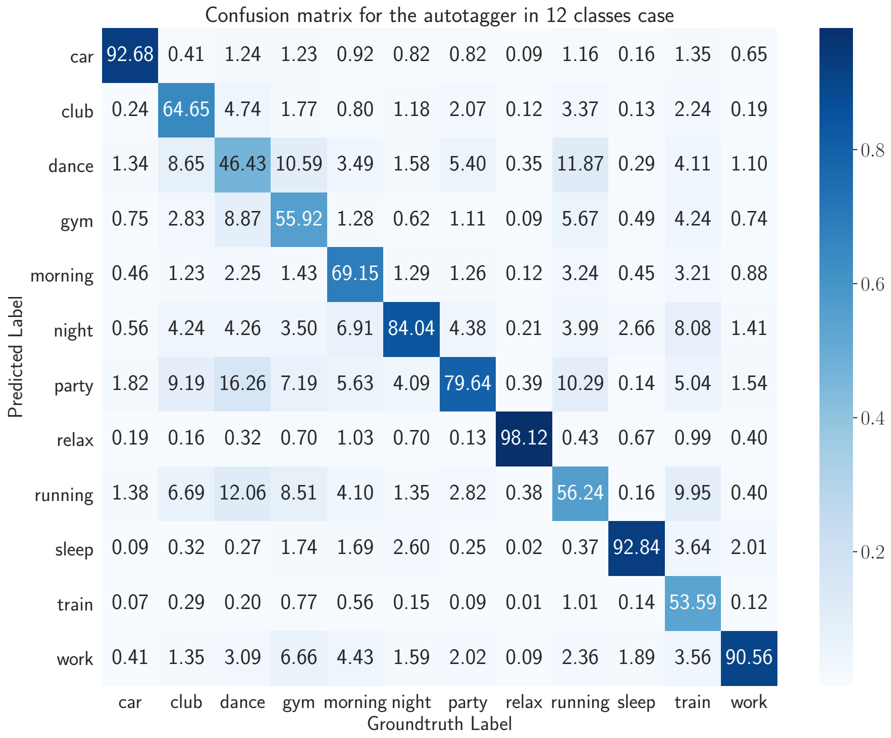
Figure 7.7 – Recall, precision, and f1-score computed for each situation in the 12 classes set in the warm case

with energetic music such as gym, running, dance or party. This confusion is indicative of the shortcomings of using keywords to express different situations without considering the intrinsic relationships between them. However, exploring these relations is still an active research problem with no complete model for categorizing the situations.

Situation Predictor Evaluation

The results for the situation predictor can be found in Table 7.4. We find that predicting the situation for new users becomes noticeably harder. In the case of 12 potential situations, the system was able to correctly predict the situation for only one fourth of the streams. However, when the system is allowed to make multiple guesses, the accuracy evidently increases. In the case where the user is to make the last decision, the system is able to include the correct situation in the top 3 suggestions 96%, 80%, and 68% in the cases of 4, 8, and 12 potential situations respectively. The choice of K, when evaluating with accuracy@k, can be obviously changed, and the performance will increase as K increases. We choose to display the results for K=3 since 3 is around the number of visible items in the carousels displayed by most streaming services on the suggestions screen on mobile devices [BSB20].

Additionally, Figure 7.9 shows the confusion matrix obtained in the 12 situations case. We observe that the confusion is mostly coherent with the statistic shown earlier of the



	car	club	dance	gym	morning	night	party	relax	running	sleep	train	work
car	92.68	0.41	1.24	1.23	0.92	0.82	0.82	0.09	1.16	0.16	1.35	0.65
club	0.24	64.65	4.74	1.77	0.80	1.18	2.07	0.12	3.37	0.13	2.24	0.19
dance	1.34	8.65	46.43	10.59	3.49	1.58	5.40	0.35	11.87	0.29	4.11	1.10
gym	0.75	2.83	8.87	55.92	1.28	0.62	1.11	0.09	5.67	0.49	4.24	0.74
morning	0.46	1.23	2.25	1.43	69.15	1.29	1.26	0.12	3.24	0.45	3.21	0.88
night	0.56	4.24	4.26	3.50	6.91	84.04	4.38	0.21	3.99	2.66	8.08	1.41
party	1.82	9.19	16.26	7.19	5.63	4.09	79.64	0.39	10.29	0.14	5.04	1.54
relax	0.19	0.16	0.32	0.70	1.03	0.70	0.13	98.12	0.43	0.67	0.99	0.40
running	1.38	6.69	12.06	8.51	4.10	1.35	2.82	0.38	56.24	0.16	9.95	0.40
sleep	0.09	0.32	0.27	1.74	1.69	2.60	0.25	0.02	0.37	92.84	3.64	2.01
train	0.07	0.29	0.20	0.77	0.56	0.15	0.09	0.01	1.01	0.14	53.59	0.12
work	0.41	1.35	3.09	6.66	4.43	1.59	2.02	0.09	2.36	1.89	3.56	90.56

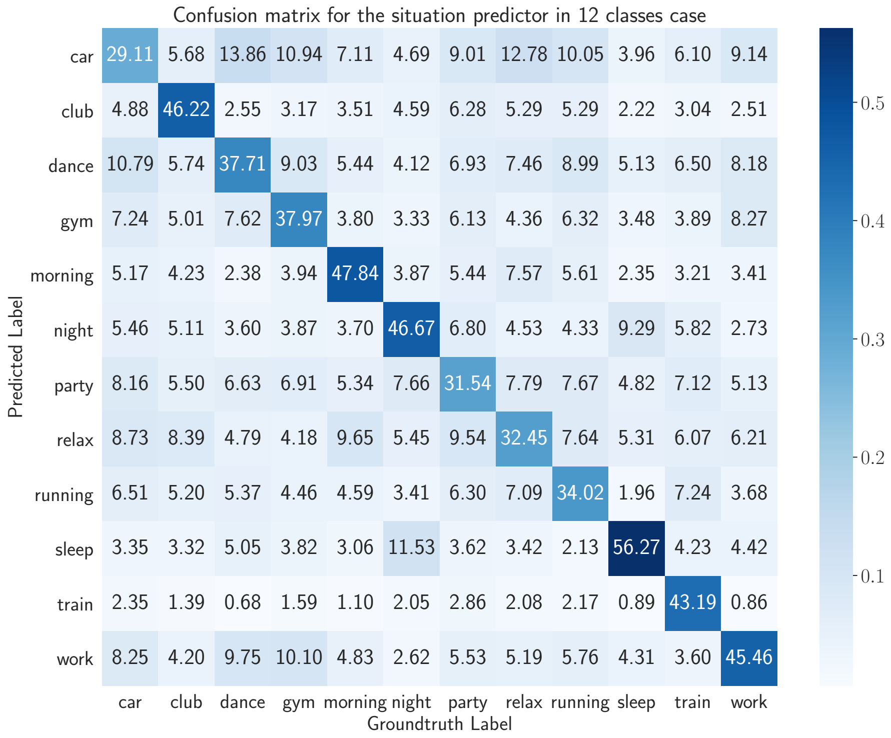
Figure 7.8 – Confusion Matrix of the auto-tagger in the case of 12 classes in the warm case

distribution of situations with the device data. Situations that are likely to originate with similar device data are harder to discriminate than the rest. For example, we observe a cluster of night-related situations including night, sleep, and relax situations. Similarly, outdoors situation are also often confused together. Discriminating those situations is hindered by the limited data available. However, the convenience of recommending top k situations provides an easy solution to further discriminate between these similar situations.

Finally, to evaluate the challenge in classifying situations from multiple sources, we compare between the evaluation results in each location separately. We compare between two different cases: 1) a model trained globally on the data from both locations but tested locally, 2) a model trained locally on each location independently and tested on the corresponding location. Table 7.5 shows the results for this evaluation setting. We find that training the models locally slightly improves the results, but not significantly. This sug-

Table 7.4 – Evaluation results of the Situation Predictor evaluated on accuracy and accuracy@3 in the three evaluation protocol splits (cold-user, cold-track, and warm case) and the three subsets of situations (4, 8, and 12). This system only rely on the user, so the cold track split is merged with the warm case. The results are shown as mean(std.).

No. of Situations	Accuracy		Accuracy @3	
	Cold User	Warm Case	Cold User	Warm Case
4	47.46 (0.98)	66.96 (0.39)	90.51 (0.31)	96.3 (0.1)
8	30.95 (0.89)	49.23 (0.16)	64.11 (1.42)	79.62 (0.13)
12	25.00 (0.29)	39.92 (0.13)	52.04 (0.61)	67.62 (0.21)



	car	club	dance	gym	morning	night	party	relax	running	sleep	train	work
car	29.11	5.68	13.86	10.94	7.11	4.69	9.01	12.78	10.05	3.96	6.10	9.14
club	4.88	46.22	2.55	3.17	3.51	4.59	6.28	5.29	5.29	2.22	3.04	2.51
dance	10.79	5.74	37.71	9.03	5.44	4.12	6.93	7.46	8.99	5.13	6.50	8.18
gym	7.24	5.01	7.62	37.97	3.80	3.33	6.13	4.36	6.32	3.48	3.89	8.27
morning	5.17	4.23	2.38	3.94	47.84	3.87	5.44	7.57	5.61	2.35	3.21	3.41
night	5.46	5.11	3.60	3.87	3.70	46.67	6.80	4.53	4.33	9.29	5.82	2.73
party	8.16	5.50	6.63	6.91	5.34	7.66	31.54	7.79	7.67	4.82	7.12	5.13
relax	8.73	8.39	4.79	4.18	9.65	5.45	9.54	32.45	7.64	5.31	6.07	6.21
running	6.51	5.20	5.37	4.46	4.59	3.41	6.30	7.09	34.02	1.96	7.24	3.68
sleep	3.35	3.32	5.05	3.82	3.06	11.53	3.62	3.42	2.13	56.27	4.23	4.42
train	2.35	1.39	0.68	1.59	1.10	2.05	2.86	2.08	2.17	0.89	43.19	0.86
work	8.25	4.20	9.75	10.10	4.83	2.62	5.53	5.19	5.76	4.31	3.60	45.46

Figure 7.9 – Confusion Matrix of the situation predictor in the case of 12 classes in the warm case

Table 7.5 – Evaluation results of the globally and locally trained models for each of the two locations in our dataset, France and Brazil, evaluated at each subset of situations in the warm case. The results are shown as mean(std.).

Situations	France		Brazil	
	Global Model	Local Model	Global Model	Local Model
4	53.4 (0.2)	55.1 (0.2)	59.2 (0.2)	61.7 (0.2)
8	35.8 (0.1)	36.8 (0.1)	45.9 (0.1)	48.2 (0.1)
12	27.6 (0.1)	28.2 (0.1)	39.6 (0.2)	41.8 (0.1)

gests that using the same one model for all available locations gives comparable results to using multiple local models. We also observe a clear distinction in the accuracy between the two locations, where Brazil scores higher than France in all cases. This is due to the larger number of users in our dataset who are in France, i.e. there are more users with more distinct patterns in the France case.

Joint Evaluation

The results for the joint system can be found in Table 7.6. As we can see, each variable in our evaluation influences the performance of the system. The most influential parameter is the number of potential situations. As the complexity increases, we find the accuracy of the model ranges from 16% in the case of 12 potential situations in the cold scenarios, up to 58% in the case of 4 potential situations with no new users or tracks. Additionally, we find the expected variation in performance between the cold cases and the warm case

Table 7.6 – The joint evaluation results of the Situation Predictor, Auto-tagger, and their overlapping predictions evaluated with accuracy in the three evaluation protocol splits (cold-user, cold-track, and warm case) and the three subsets of situations (4, 8, and 12). The results are shown as mean(std.).

Model	Cold Users	Cold Tracks	Warm Case
	4 Situations		
Situation Predictor	47.46 (0.98)	66.81 (0.35)	67.20 (0.26)
Auto-tagger	69.73 (1.07)	63.78 (2.33)	83.75 (2.33)
Overlap	36.22 (1.27)	44.60 (1.01)	58.92 (1.71)
	8 Situations		
Situation Predictor	30.95 (0.89)	49.13 (0.24)	49.35 (0.19)
Auto-tagger	47.56 (0.53)	52.44 (2.31)	70.81 (1.45)
Overlap	17.77 (0.49)	28.94 (1.24)	39.52 (1.27)
	12 Situations		
Situation Predictor	25.00 (0.29)	39.05 (0.31)	39.19 (0.14)
Auto-tagger	52.68 (1.25)	37.61 (3.47)	69.14 (3.79)
Overlap	16.19 (0.32)	18.75 (1.63)	31.26 (1.30)

of intended use. We observe how the drop in the performance of the situation predictor and auto-tagger, on new users/tracks, negatively affects the joint system performance.

However, in the harder evaluation case of generating a situational playlists with only 1 guess allowed out of 12, the proposed system would have been able to include at least a third of the actual listened tracks in those playlists, while pushing them to the user at the exact listened time.

Recommendation Evaluation

Finally, we evaluate the potential of auto-tagging tracks with situational tags in a recommendation scenario. We achieve this by employing a sequential session-aware method (CoseRNN) on its own and compare its performance with the case of (Prefiltering) the potential tracks. It is important to note that prefiltering the tracks pool prevents us from evaluating with certain ranking metrics that require reaching a recall of 1, e.g. mean average precision. Hence, we display the ranking evaluation metrics as is in Table 7.7. We observe a clear improvement across those metrics with increasing number of retrieved tracks, K. This demonstrates that the prefiltering is efficient in removing tracks less likely to be listened by the given user in each situation and facilitates the retrieval task using traditional sequential recommenders.

It is also important to emphasize that the proposed approach does not replace these traditional recommenders used in streaming services, but aims at enhancing them. This is achieved by isolating the situational listening sessions from non-situational ones. This separation is motivated by the strong variation in listening preferences between the two cases. The situational sessions can deviate from the user's "average" preferences and is often motivated by a specific purpose being sought through the music, e.g. energizing music for sports. This deviation can hinder the performance of traditional recommenders where this temporary change in preference is not revealed to the recommender. Hence, it is possible that this separation of situational sessions can further improve the development of recommenders in non-situational cases by allowing them to focus on one task instead of several tasks simultaneously.

Table 7.7 – Comparison of CoseRNN model with and without prefiltering using predicted tags from the auto-tagger, evaluated with Recall@K, Precision@K, and F1-score. Results are displayed as percentage.

K	Recall@K		Precision@K		F1-score@K	
	No filtering	Prefiltering	No filtering	Prefiltering	No filtering	Prefiltering
10	0.679	0.933	0.489	2.261	0.569	1.321
20	1.286	1.611	0.437	1.979	0.653	1.776
30	1.898	2.184	0.412	1.808	0.677	1.978
40	2.551	2.719	0.401	1.707	0.693	2.097
50	3.357	3.254	0.398	1.633	0.712	2.174
75	5.566	4.355	0.391	1.475	0.731	2.204
100	8.086	5.366	0.393	1.375	0.749	2.189

7.4.3 Discussion

The evaluation results give us various insights of the system. The first impression would be of potential towards a convenient feature to facilitate music streaming. It hints as well to a range of service-customization techniques that could be used to significantly improve the performance. Out of those techniques, we bring to the table customizing potentials situations. The appealing performance in the case of 4 situations directs us towards selecting the subset of potential situations based on the users, which could even be inserted by the users themselves.

Additionally, representing the listening situation as a tag attached to a track/user pair allows for a range of applications to be accessible. In our experiments, we show-cased one application where the tags were used to prefilter the recommended tracks based on the current listening situation. Furthermore, they could facilitate playlist generation, providing personalized results for a search query with a situational keyword, or playlist/session continuation based on the common situational tags in the recent tracks.

However, the evaluation raises unanswered questions in some aspects. The study is based on playlists with a “situational keyword”, and users who listened to them were being influenced by these situations. Although we tried to be thorough with our data collection, this was a strong assumption that we could not validate. This raises to question, if this is sufficient to collect quality data for training the models. Additionally, the brute keyword matching does not account for intrinsic relationships between those situations, e.g. sport-related situations could be clustered together. It also does not account for the multiple meanings of a certain keyword, e.g. “train” can be a place or an activity. This motivates future research to develop a categorical representation for these situations to better classify them.

But, are the users OK with their patterns being exploited?

This study will not be complete without a discussion of the privacy concerns for the users. Specifically, when it comes to using their usage patterns to predict their daily activities. While it is arguable that these recommendation systems are developed primarily to provide better experience for the users, there has been negative sentiment towards revealing personal information online [SBS11]. This personalization-privacy paradox has been in discussion for as long as personalized services [AK06]; a discussion that grew into an urge towards systems that reveal their intentions instead [CKS$^+$19].

Hence, we suggest that using the user involvement paradigm is a workaround in this paradox. Moreover, this feature can and should be optional, even to work properly. Users who do not have situational listening patterns should not be included in the system, similar to the case is our dataset used for training and testing. It is arguable that users are suspicious of background activities, but this transparent interpretable approach is potentially intuitive to the users for a better personalized experience. We are keen on finding out if this user-service ecosystem would be more approachable by the users than to ghost them and trespass with overreaching recommendations.

7.5 Conclusion

In this chapter, we explained our proposed approach for generating personalized situational sessions to online music streamers. This practical approach is developed to be used by online streaming services as an interactive feature for its frequent users. Through the proposed approach, users could access a set of predicted potential situations. When one is activated by the user, it automatically generates a playlist of tracks "likely" to be listened to by the user in this situation. This likelihood would be estimated using an auto-tagger trained on predicting the situational use of tracks, given a specific user and his/her listening history. This alternative approach of treating the context as a tag describing a listening situation provides a strong case for interpretable recommendations, since these tags can be easily communicated and understood by the user.

Our evaluations of this system showed promising results supporting the approach of isolating the situational sessions from the traditional recommendation task. We evaluated each of our system's blocks individually, combined, then in a recommendation scenario. The evaluation results indicated that the system is capable of learning personalized patterns for the users, which when employed for recommendation outperformed traditional recommendation.

Chapter 8

Conclusion and Future Work

8.1 Summary

In this thesis, we addressed the problem of tagging tracks with potential listening situations based on the audio content of the tracks. This unique setup of the problem is an alternative and interpretable approach towards providing contextual recommendation to music listeners. We incrementally investigated the use and design of music auto-taggers in order to annotate music with personalized situational tags. To achieve this, we have devised a semi-automatic pipeline for linking the listening situation to the listened tracks by exploiting playlist titles. Through our pipeline, we were able to collect large-scale datasets of tracks tagged with groundtruth situations and made them available to the community to foster future work on this problem. Additionally, throughout this work, we have given special attention to the adaptability of the designed system to real-world services.

We have tested a state-of-the-art auto-tagging model on our collected datasets and adapted it accordingly to the problem. This adaption entailed first addressing the challenge of missing negative labels in our datasets. Subsequently, it entailed changing the models to process user data simultaneously in order to give personalized tag predictions. We designed several evaluation protocols and validated the importance of considering the users in the task of situational auto-tagging. Finally, to put this approach into recommendation scenarios, it is essential to be able to predict the current active situation of a given user. We have exploited the available device data during a streaming session in order to infer the potential situations. Although this problem is significantly harder, given the limited available data, we have reached satisfactory results by allowing several simultaneous predictions. This was particularly useful in industrial cases where services often arrange their suggestions in a carousel with multiple options.

As explained in Chapter 1, we address four subjects in this thesis: Identifying relevant situations, applying music auto-taggers to situational tags, adapting auto-taggers to provide personalization, and automatically inferring active listening situations. Our main contributions in each task are described in the following paragraphs:

A dataset collection pipeline for linking situations, tracks, users, and device data: We have built on top of previous work on this topic to find a proxy to collect the required data for our studies. We first identified the potential keywords describing the listening situations that would serve as “tags” in our setup as explained in Chapter 5. We have then collected three datasets, first of their kind, that label a track/user/device data with a listening situation. We achieved this by exploiting playlists titles matched with the set of defined keywords/tags. We experimented with multiple filtering approaches to ensure high quality of the collected data.

Each of our datasets provide a specific attribute useful for studying each of our subsequent problems. The first dataset is a multi-context dataset of tracks that are frequently associated with these contexts. The second dataset is a user-aware dataset composed of tracks tagged with the context associated to them by the playlist creators, along with the creators embeddings. Finally, the third dataset focuses on retrieving the device data on which users listened to these playlists. This dataset provides a triplet of user/track/device tagged with listening situations retrieved from playlist titles. All our datasets are made public to encourage future work on this problem.

Adapting music auto-taggers to situational tags: Our second task is to study the usability of music auto-taggers in identifying the potential listening situations. Given that our evaluation results were comparable to the average performance of music auto-taggers on non-situational tags, this study has encouraged the pursuit of adapting auto-taggers to this specific task as shown in Chapter 5. However, the variation of the performance in-between different situations has further strengthened our assumption of user-dependency. But before exploring how to involve the users in the process, we had to overcome another problem associated with the nature of the data, the missing negative labels.

The procedure we used for collecting the dataset puts emphasis on the positive labels, i.e. our filtering steps are designed to ensure that only true positives are included. However, we have no proxy for validating negative labels. Hence, we have proceeded with deriving a confidence-based weighting scheme for the loss function used in training the model. This correlation-based weighting proved to improve the training stage of the model. We further verified these results by creating artificial missing labels on complete multi-label datasets, which clearly showed a significant improvement. This is particularly useful where the architecture of the model is predefined, such as our case, and an ad-hoc approach is required to address the missing labels.

Adapting auto-taggers to provide personalization: Auto-taggers have been largely used on tags that are solely content-based. However, this is not necessarily the case when it comes to situational tags. Hence, we adapted a state-of-the-art music auto-tagger to process the user information simultaneously with the track content. We achieved this by representing each users through an embedding vector derived from their previous listening history. Our proposed evaluation protocols, designed to measure user satisfaction, proved that auto-taggers can provide personalized and more accurate predictions compared to the original models as shown in chapter 6. Our user-aware music auto-tagger is the first attempt in developing personalized situational auto-taggers.

Joiny music auto-taggers with a situation prediction model for a situational recommendation framework Finally, to incorporate this approach to an into a real-time application, it is essential to predict the active listening situation. Hence, we have exploited the available device data to achieve this task. Even though the available data is quite limited, our trained models were able to achieve promising results as shown in Chapter 7. By combining the two models (the auto-tagger and situation predictor), we developed a framework to filter the set of potential tracks in real-time, based on the predicted situation, before deploying traditional recommendation algorithms and proposing the results to the user.

This framework provides an alternative pathway to recommending situational sessions, aside from the primary sequential recommendation system deployed by the service, which is interpretable through the tags. Our evaluation showed that prefiltering those situational sessions with the corresponding tags significantly improved the performance of the traditional recommendation algorithm in situational sessions. This approach to the problem is a first-time attempt into providing contextual recommendations through auto-tagging.

8.2 Limitations and Future Work

Our work has covered multiple topics where each has its own limitations and potential for future work. The most evident influence on our work would be the progress in developing auto-taggers. One can assume that obtaining complex music auto-taggers that are capable of extracting more musically-relevant features from the audio content would instantly improve their performance in the situational tagging case. This is bound to the overall future work regarding the auto-tagging task in MIR, i.e. not exclusive to the problem in hand. However, there is a number of potential steps to be taken regarding our approach to the problem.

General limitations In all our studies, we have made an assumption that dictates: any session associated with a situational playlist is indeed a situational session, and users listening to it are actively experiencing this situation. Even though this simplification significantly helped in collecting and labelling our datasets, it is a strong assumption that we have not validated. There is always a possibility that the users either like the playlist content without regarding its target situation, or it is a continuation of an earlier session which is no longer being listened to in the corresponding situation. This can impair the device data collected in the third dataset.

Furthermore, due to the dataset distribution, we had to shift the problem from a multi-label setup to a single-label one. This was motivated by the fact that the users included in the dataset rarely have playlists made for several situations. However, in reality, each user could have the same track associated with multiple situations. Our approach does not address this case and only predicts a single situation for a user/track pair. The simplest solution for this shortcoming would be training the models in a single-label setting, before replacing the prediction layer with a multi-label one, i.e. without normalizing the predicted probabilities. However, this is not guaranteed to achieve the full potential without fine-tuning on a multi-label groundtruth. Hence, the more elaborate solution would be to find a proper procedure for collecting user-aware datasets in a multi-label setting.

Additionally, personalized auto-taggers are far less explored and possess the potential to significantly improve through dedicated studies on personalization. In our work, we have used history-based embeddings that capture the global preferences of the user. However, the global preferences are not necessarily reflective of each user's preference in each of the possible situations. Hence, personalization can be further improved through a context-centered representation. One possible approach would be deriving embeddings based solely on the situational sessions, detected through their association with the titles of the listened playlists. Another possibility is through creating a user profile from a one-time questionnaire about their situational preferences.

Context hierarchy Our work has treated the context as a brute keyword describing an independent listening situation. However, this approach fails to address the interrelations of these listening situations. One can assume that sport-related or dance-related situations all share similar musical content. However, this is an oversimplification that should be addressed properly. Drawing a line between independent, related, and identical situations needs to be further investigated. One potential solution is to use a hierarchical representation of the situations adding more specificity as it grows. A starting point can be already available taxonomies. However, these taxonomies are more centered around the semantic similarity rather than the musical similarity. It is important to derive this contextual hierarchy in a way that reflects the musical similarity between the situations when being experienced.

Furthermore, this will again be challenged by the personalization step. It is arguable an identical situation for one user could mean two completely different situations for another. In order to properly achieve personalization, these differences should be also addressed in future work. One suggestion would be to adapt the auto-taggers to output learnable embeddings that represent several potential situations using the track content and the user, i.e. a personalized situational encoder. This representation can be further translated into comprehensible tags through an interpretation model that can hierarchically output the potential situations for a given user. However, this can only be achieved given a proper multi-context user-aware dataset along with a hierarchical representation of the situations.

Evaluation procedure A significant challenge when it comes to either context or recommendations is the evaluation stage. These multi-faceted problems are harder to evaluate as they are not fully observed due to the inability to judge the users subjective opinion. In our setup, the problem is considered a classification problem and is evaluated as so. However, the bigger picture is different. First of all, we are evaluating on a slice of data from the past. This does not actually reflects how this system would perform in a real-time setting. Hence, further evaluations with real-time users is highly encouraged. The most suitable approach would be to test this framework in an AB testing setup. Given our preliminary results, we can suggest proceeding with online testing. However, AB testing is costly as it is deployed through an actual service and could hinder the perception of the service. An alternative would be performing a smaller scale user evaluation/surveys on the quality of the predictions. A favorable result can then encourage for AB testing.

Beyond the audio content Finally, our work has solely focused on the audio content for disambiguating the potential situations. However, similar to other MIR tasks, other sources of data can provide additional information to achieve the task. For example, processing the lyrics has proved to be useful in various tasks, including auto-tagging targeting emotions [PFOT19]. Similarly, lyrics can be a decisive feature for a track which is sought after in specific situations, e.g. motivational lyrics can be appreciated in certain contexts. Additionally, user reviews for a specific artist or album could reflect the sentiment of the listeners towards the content, e.g. relaxing or energetic. Multi-modal approaches have been consistently proving to be superior to models that rely on a single source of data [SNA19]. Furthermore, the non-situational tags associated with the tracks could also provide a useful additional information that can be included in this specific tagging task.

To conclude, we have performed a number of studies aiming at facilitating contextual music recommendation through tagging. Our studies focused on lying a foundation for this alternative setup for future work. Our setup was motivated by applicability in the industry in a data-driven manner. Our results showed promising potential towards an interpretable contextual recommendation process. Finally, we have highlighted several potential future works that can further investigate sub-tasks from each of our studies.

Bibliography

[ABB14] Marharyta Aleksandrova, Armelle Brun, and Anne Boyer. What about Interpreting Features in Matrix Factorization-based Recommender Systems as Users? In *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, Santiago du Chili, Chile, September 2014.

[ADB$^+$99] Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *The International symposium on handheld and ubiquitous computing*, pages 304–307. Springer, 1999.

[ADNDS$^+$20] Vito Walter Anelli, Tommaso Di Noia, Eugenio Di Sciascio, Azzurra Ragone, and Joseph Trotta. Semantic interpretation of top-n recommendations. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[AEK00] Asim Ansari, Skander Essegaier, and Rajeev Kohli. Internet recommendation systems. *Journal of Marketing Research*, 37(3):363–375, 2000.

[AG19] Marie Al-Ghossein. *Context-aware recommender systems for real-world applications.* PhD thesis, Université Paris-Saclay (ComUE), 2019.

[Aga18] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[AHT08] Gediminas Adomavicius, Zan Huang, and Alexander Tuzhilin. Personalization and recommender systems. In *State-of-the-Art Decision-Making Tools in the Information-Intensive Age*, pages 55–107. INFORMS, 2008.

[AK06] Naveen Farag Awad and Mayuram S Krishnan. The personalization privacy paradox: an empirical evaluation of information transparency and the willingness to be profiled online for personalization. *MIS quarterly*, pages 13–28, 2006.

[Ake17] Imen Akermi. *A hybrid model for context-aware proactive recommendation.* PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2017.

[AKTV14] Ashton Anderson, Ravi Kumar, Andrew Tomkins, and Sergei Vassilvitskii. The dynamics of repeat consumption. In *Proceedings of the 23rd international conference on World wide web*, 2014.

[Ama13] Xavier Amatriain. Mining large streams of user data for personalized recommendations. *ACM SIGKDD Explorations Newsletter*, 14(2):37–48, 2013.

[AS11] Anupriya Ankolekar and Thomas Sandholm. Foxtrot: a soundtrack for where you are. In *Proceedings of Interacting with Sound Workshop: Exploring Context-Aware, Local and Social Audio Applications*, pages 26–31, 2011.

[AT11] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.

[BA09] Linas Baltrunas and Xavier Amatriain. Towards time-dependant recommendation based on implicit feedback. In *Workshop on context-aware recommender systems (CARS'09)*, pages 25–30, 2009.

[BBC97] Peter J Brown, John D Bovey, and Xian Chen. Context-aware applications: from the laboratory to the marketplace. *IEEE personal communications*, 4(5):58–64, 1997.

[BCJ+18] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H Chi. Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018.

[BCN18] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

[BHBF+10] Dmitry Bogdanov, Martín Haro Berois, Ferdinand Fuhrmann, Emilia Gómez Gutiérrez, Herrera Boyer, et al. Content-based music recommendation based on user preference examples. In *Anglade A, Baccigalupo C, Casagrande N, Celma Ò, Lamere P, editors. Workshop on Music Recommendation and Discovery 2010 (WOMRAD 2010); 2010 Sep 26; Barcelona, Spain. Aachen: CEUR Workshop Proceedings; 2010. p. 33-8.* CEUR Workshop Proceedings, 2010.

[BHK98] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 1998.

[BHK13] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *arXiv preprint arXiv:1301.7363*, 2013.

[Bis06] Christopher M Bishop. Pattern recognition. *Machine learning*, 128(9), 2006.

[BJJ11] Serhat Selcuk Bucak, Rong Jin, and Anil K. Jain. Multi-label learning with incomplete class assignments. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011.

[BK07] Robert M Bell and Yehuda Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Seventh IEEE international conference on data mining (ICDM 2007)*, pages 43–52, 2007.

[BK14] Wei Bi and James T Kwok. Multilabel classification with label correlations and missing labels. In *Proceedings of 28th AAAI Conference on Artificial Intelligence*, 2014.

[BKL+11] Linas Baltrunas, Marius Kaminskas, Bernd Ludwig, Omar Moling, Francesco Ricci, Aykan Aydin, Karl-Heinz Lüke, and Roland Schwaiger. Incarmusic: Context-aware music recommendations in a car. In *International conference on electronic commerce and web technologies*, pages 89–100. Springer, 2011.

[BL+07] James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, page 35. New York, NY, USA., 2007.

[BLSB04] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

[BMEM11] Thierry Bertin-Mahieux, Douglas Eck, and Michael Mandel. Automatic tagging of audio: The state-of-the-art. In *Machine audition: Principles, algorithms and systems*, pages 334–352. IGI Global, 2011.

[BMEML08] Thierry Bertin-Mahieux, Douglas Eck, Francois Maillet, and Paul Lamere. Autotagger: A model for predicting social tags from acoustic features on large music databases. *Journal of New Music Research*, 37(2):115–135, 2008.

[BOHG13] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowledge-based systems*, 46:109–132, 2013.

[Bou13a] Djallel Bouneffouf. The impact of situation clustering in contextual-bandit algorithm for context-aware recommender systems. *arXiv preprint arXiv:1304.3845*, 2013.

[Bou13b] Djallel Bouneffouf. Situation-aware approach to improve context-based recommender system. *arXiv preprint arXiv:1303.0481*, 2013.

[BR14] Linas Baltrunas and Francesco Ricci. Experimental evaluation of context-dependent collaborative filtering using item splitting. *User Modeling and User-Adapted Interaction*, 24(1-2):7–34, 2014.

[Bra97] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[BS97] Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[BSB20] Walid Bendada, Guillaume Salha, and Théo Bontempelli. Carousel personalization in music streaming apps with contextual bandits. In *Fourteenth ACM Conference on Recommender Systems*, pages 420–425, 2020.

[Bur02] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[Bur07] Robin Burke. Hybrid web recommender systems. *The adaptive web*, pages 377–408, 2007.

[BYRN$^+$99] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

[CAS16] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, 2016.

[CBF06] Sally Jo Cunningham, David Bainbridge, and Annette Falconer. 'more of an art than a science': Supporting the creation of playlists and mixes. In *International Society for Music Information Retrieval (ISMIR)*, 2006.

[CCG08] Stuart Cunningham, Stephen Caulder, and Vic Grout. Saturday night or fever? context-aware music playlists. *Proceedings of Audio Mostly*, 2008.

[CDC14] Pedro G Campos, Fernando Díez, and Iván Cantador. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, 24(1):67–119, 2014.

[CFCS18] Keunwoo Choi, György Fazekas, Kyunghyun Cho, and Mark Sandler. The effects of noisy labels on deep convolutional neural networks for music tagging. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(2):139–149, 2018.

[CFS16a] Keunwoo Choi, George Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks. *arXiv preprint arXiv:1606.00298*, 2016.

[CFS16b] Keunwoo Choi, George Fazekas, and Mark Sandler. Explaining deep convolutional neural networks on music classification. *arXiv preprint arXiv:1607.02444*, 2016.

[CFSC17] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. In *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[CFSC18] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. A comparison of audio signal preprocessing methods for deep neural networks on music tagging. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1870–1874, 2018.

[CG16] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM sigkdd international conference on knowledge discovery and data mining*, 2016.

[CHS06] Ò. Celma, Perfecto Herrera, and Xavier Serra. Bridging the music semantic gap. In *Workshop on Mastering the Gap: From Information Extraction to Semantic Representation*, volume 187, Budva, Montenegro, 2006. CEUR.

[CKS$^+$19] Tathagata Chakraborti, Anagha Kulkarni, Sarath Sreedharan, David E Smith, and Subbarao Kambhampati. Explicability? legibility? predictability? transparency? privacy? security? the emerging landscape of interpretable agent behavior. In *Proceedings of the international conference on automated planning and scheduling*, 2019.

[CKW05] Pedro Cano, Markus Koppenberger, and Nicolas Wack. Content-based music audio recommendation. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 211–212, 2005.

[CLSZ18] Ching-Wei Chen, Paul Lamere, Markus Schedl, and Hamed Zamani. Recsys challenge 2018: Automatic music playlist continuation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018.

[ÇM$^+$17] Erion Çano, Maurizio Morisio, et al. Music mood dataset creation based on last. fm tags. In *2017 International Conference on Artificial Intelligence and Applications, Vienna, Austria*, pages 15–26, 2017.

[CMZG09] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 621–630, 2009.

[Coo01] Colleen Cool. The concept of situation in information science. *Annual Review of Information Science and Technology (ARIST)*, 35:5–42, 2001.

[CPC19] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.

[CPVA10] Toni Cebrián, Marc Planagumà, Paulo Villegas, and Xavier Amatriain. Music recommendations with temporal context awareness. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 349–352, 2010.

[CS08] Òscar Celma and Xavier Serra. Foafing the music: Bridging the semantic gap in music recommendation. *Journal of Web Semantics*, 6(4):250–256, 2008.

[CS14a] Zhiyong Cheng and Jialie Shen. Just-for-me: an adaptive personalization system for location-aware social music recommendation. In *Proceedings of international conference on multimedia retrieval*, pages 185–192, 2014.

[CS14b] Zhiyong Cheng and Jialie Shen. Just-for-me: An adaptive personalization system for location-aware social music recommendation. In *Proceedings of International Conference on Multimedia Retrieval*, 2014.

[CSWZ08] Gang Chen, Yangqiu Song, Fei Wang, and Changshui Zhang. Semi-supervised multi-label learning by solving a sylvester equation. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, 2008.

[CTH$^+$09] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM international conference on image and video retrieval*, 2009.

[CYJL16] Szu-Yu Chou, Yi-Hsuan Yang, Jyh-Shing Roger Jang, and Yu-Ching Lin. Addressing cold start for next-song recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016.

[CZTR08] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*, pages 87–94, 2008.

[DBS11] Sander Dieleman, Philémon Brakel, and Benjamin Schrauwen. Audio-based music classification with a pretrained convolutional network. In *12th International Society for Music Information Retrieval Conference (ISMIR-2011)*, pages 669–674. University of Miami, 2011.

[DDS$^+$09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2009.

[Dey98] Anind K Dey. Context-aware computing: The cyberdesk project. In *Proceedings of the AAAI Spring Symposium on Intelligent Environments*, pages 51–54, 1998.

[Dey00] Anind Kumar Dey. *Providing architectural support for building context-aware applications*. Georgia Institute of Technology, 2000.

[DF13] Ricardo Dias and Manuel J Fonseca. Improving music recommendation in session-based collaborative filtering by using temporal context. In *IEEE 25th international conference on tools with artificial intelligence*, pages 783–788, 2013.

[DFC14] Ricardo Dias, Manuel J Fonseca, and Ricardo Cunha. A user-centered music recommendation approach for daily activities. In *CBRecSys@ RecSys*, pages 26–33, 2014.

[DGLM$^+$15] Marco De Gemmis, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. Semantics-aware content-based recommender systems. In *Recommender systems handbook*, pages 119–159. Springer, 2015.

[DK04] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[DMM19] Thibaut Durand, Nazanin Mehrasa, and Greg Mori. Learning a deep convnet for multi-label classification with partial labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[Dou04] Paul Dourish. What we talk about when we talk about context. *Personal and ubiquitous computing*, 8(1):19–30, 2004.

[DRK+14] Jia Deng, Olga Russakovsky, Jonathan Krause, Michael S Bernstein, Alex Berg, and Li Fei-Fei. Scalable multi-label annotation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3099–3102, 2014.

[DS14] Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6964–6968, 2014.

[DWTS16] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. Deep coevolutionary network: Embedding user and item features for recommendation. *arXiv preprint arXiv:1609.03675*, 2016.

[EN08] Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–220, 2008.

[Fis01] Gerhard Fischer. User modeling in human–computer interaction. *User modeling and user-adapted interaction*, 11(1):65–86, 2001.

[FS16] Bruce Ferwerda and Markus Schedl. Investigating the relationship between diversity in music consumption behavior and cultural dimensions: A cross-country analysis. In *Proceedings of the International Conference on User Modeling, Adaptation, and Personalization*, 2016.

[FYST19] Bruce Ferwerda, Emily Yang, Markus Schedl, and Marko Tkalcic. Personality and taxonomy preferences, and the influence of category choice on the user experience for music streaming services. *Multimedia tools and applications*, 78(14):20157–20190, 2019.

[GB10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[GBL+18] Mingchen Gao, Ulas Bagci, Le Lu, Aaron Wu, Mario Buty, Hoo-Chang Shin, Holger Roth, Georgios Z Papadakis, Adrien Depeursinge, Ronald M Summers, et al. Holistic classification of ct attenuation patterns for interstitial lung diseases via deep convolutional neural networks. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 6(1):1–6, 2018.

[GBY+18] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89, 2018.

[GD98] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636, 1998.

[GL11] Alinka E Greasley and Alexandra Lamont. Exploring engagement with music in everyday life using experience sampling methodology. *Musicae Scientiae*, 15(1):45–71, 2011.

[GL14] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[GPT11] Michele Gorgoglione, Umberto Panniello, and Alexander Tuzhilin. The effect of context-aware recommendations on customer purchasing behavior and trust. In *Proceedings of the fifth ACM conference on Recommender systems*, 2011.

[GPZ05] Tao Gu, Hung Keng Pung, and Da Qing Zhang. A service-oriented middleware for building context-aware services. *Journal of Network and computer applications*, 28(1):1–18, 2005.

[GS15a] Michael Gillhofer and Markus Schedl. Iron maiden while jogging, debussy for dinner? In *International Conference on Multimedia Modeling*, pages 380–391. Springer, 2015.

[GS15b] Asela Gunawardana and Guy Shani. Evaluating recommender systems. In *Recommender systems handbook*, pages 265–308. Springer, 2015.

[GSMDS12] Carlos Gomes, Daniel Schneider, Katia Moraes, and Jano De Souza. Crowdsourcing for music: Survey and taxonomy. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 832–839, 2012.

[GSS18] Fabian Greb, Wolff Schlotz, and Jochen Steffens. Personal and situational influences on the functions of music listening. *Psychology of Music*, 46(6):763–794, 2018.

[GUH15] Carlos A Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):1–19, 2015.

[GV15] Eva Gibaja and Sebastián Ventura. A tutorial on multilabel learning. *ACM Computing Surveys (CSUR)*, 47(3):52, 2015.

[HAIS$^+$17a] Khalid Haruna, Maizatul Akmar Ismail, Suhendroyono Suhendroyono, Damiasih Damiasih, Adi Pierewan, Haruna Chiroma, and Tutut Herawan. Context-aware recommender system: A review of recent developmental process and future research direction. *Applied Sciences*, 7(12), 2017.

[HAIS$^+$17b] Khalid Haruna, Maizatul Akmar Ismail, Suhendroyono Suhendroyono, Damiasih Damiasih, Adi Cilik Pierewan, Haruna Chiroma, and Tutut Herawan. Context-aware recommender system: A review of recent developmental process and future research direction. *Applied Sciences*, 7(12):1211, 2017.

[HC00] Conor Hayes and Pádraig Cunningham. Smart radio: Building music radio on the fly. *Expert Systems*, 2000:2–6, 2000.

[HF01] David B Hauver and James C French. Flycasting: using collaborative filtering to generate a playlist for online radio. In *Proceedings First International Conference on WEB Delivering of Music. WEDELMUSIC*, pages 123–130. IEEE, 2001.

[HHM$^+$20] Casper Hansen, Christian Hansen, Lucas Maystre, Rishabh Mehrotra, Brian Brost, Federico Tomasi, and Mounia Lalmas. Contextual and sequential user embeddings for large-scale music recommendation. In *Pro-*

ceedings of the Fourteenth ACM Conference on Recommender Systems, 2020.

[HHSC18] Mengying Hu, Hu Han, Shiguang Shan, and Xilin Chen. Multi-label learning from noisy labels with non-linear feature transformation. In *Proceedings of the Asian Conference on Computer Vision*, 2018.

[HKR00] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250, 2000.

[HMB12] Negar Hariri, Bamshad Mobasher, and Robin Burke. Context-aware music recommendation based on latent topic sequential patterns. In *Proceedings of the sixth ACM conference on Recommender systems*, 2012.

[HN99] David J Hargreaves and Adrian C North. The functions of music in everyday life: Redefining the social in music psychology. *Psychology of music*, 27(1):71–83, 1999.

[HQZ$^+$19] Jun Huang, Feng Qin, Xiao Zheng, Zekai Cheng, Zhixiang Yuan, Weigang Zhang, and Qingming Huang. Improving multi-label classification with missing labels by learning label-specific features. *Information Sciences*, 492:124–146, 2019.

[HRS10] Perfecto Herrera, Zuriñe Resa, and Mohamed Sordo. Rocking around the clock eight days a week: an exploration of temporal patterns of music listening. In *Proceedings of the 1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain*, 2010.

[HS15] Mohammad Hossin and MN Sulaiman. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):1, 2015.

[HSK09] Jong-yi Hong, Eui-ho Suh, and Sung-Jin Kim. Context-aware systems: A literature review and classification. *Expert Systems with applications*, 36(4):8509–8522, 2009.

[HYG$^+$19] Zhi-Fen He, Ming Yang, Yang Gao, Hui-Dong Liu, and Yilong Yin. Joint multi-label classification and label correlations with missing labels and feature selection. *Knowledge-Based Systems*, 163:145–158, 2019.

[HZ11] Neil Hurley and Mi Zhang. Novelty and diversity in top-n recommendation–analysis and evaluation. *ACM Transactions on Internet Technology (TOIT)*, 10(4):1–30, 2011.

[HZKC16] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016.

[IB04] Peter Ingwersen and Nick Belkin. Information retrieval in context-irix: workshop at sigir 2004-sheffield. In *ACM SIGIR Forum*, volume 38, pages 50–52, 2004.

[IEPR20] Karim Ibrahim, Elena Epure, Geoffroy Peeters, and Gael Richard. Should we consider the users in contextual music auto-tagging models? In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[JHM$^+$17] Andreas Jansson, Eric Humphrey, Nicola Montecchio, Rachel Bittner, Aparna Kumar, and Tillman Weyde. Singing voice separation with deep u-net convolutional networks. 2017.

[JK02] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[JMN+16] Kurt Jacobson, Vidhya Murali, Edward Newett, Brian Whitman, and Romain Yon. Music personalization at spotify. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016.

[JWK14] Gawesh Jawaheer, Peter Weller, and Patty Kostkova. Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 4(2):1–26, 2014.

[KABO10] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 79–86, 2010.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KB15] Yehuda Koren and Robert Bell. Advances in collaborative filtering. *Recommender systems handbook*, pages 77–118, 2015.

[KBV09] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[KDF+13] Ron Kohavi, Alex Deng, Brian Frasca, Toby Walker, Ya Xu, and Nils Pohlmann. Online controlled experiments at large scale. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1168–1176, 2013.

[KLN18] Taejun Kim, Jongpil Lee, and Juhan Nam. Sample-level cnn architectures for music auto-tagging using raw waveforms. In *Proceedings of the 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2018.

[KMG16] Young-Jun Ko, Lucas Maystre, and Matthias Grossglauser. Collaborative recurrent neural networks for dynamic recommender systems. In *Proceedings of the Asian Conference on Machine Learning*. PMLR, 2016.

[Kor09] Yehuda Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 447–456, 2009.

[KR11] Marius Kaminskas and Francesco Ricci. Location-adapted music recommendation using tags. In *International conference on user modeling, adaptation, and personalization*, pages 183–194. Springer, 2011.

[KR12] Marius Kaminskas and Francesco Ricci. Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review*, 6(2-3):89–119, 2012.

[KRS13] Marius Kaminskas, Francesco Ricci, and Markus Schedl. Location-aware music recommendation using auto-tagging and hybrid matching. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 17–24, 2013.

[KSKR17] Thomas Krismayer, Markus Schedl, Peter Knees, and Rick Rabiser. Prediction of user demographics from music listening habits. In *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, pages 1–7, 2017.

[KSL19] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[KZL19] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[Lam08] Paul Lamere. Social tagging and music information retrieval. *Journal of New Music Research*, 37:101 – 114, 2008.

[LBBH98] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LBD$^+$89] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[LCHL17] Wei-Po Lee, Chun-Ting Chen, Jhih-Yuan Huang, and Jhen-Yi Liang. A smartphone-based activity-aware system for music streaming recommendation. *Knowledge-Based Systems*, 131:70–82, 2017.

[LGG$^+$17a] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[LGG$^+$17b] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[LGLR15] Romain Lerallut, Diane Gasselin, and Nicolas Le Roux. Large-scale realtime product recommendation at criteo. In *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015.

[LHR09] Hao Liu, Jun Hu, and Matthias Rauterberg. Music playlist recommendation based on user heartbeat and music preference. In *2009 International Conference on Computer Technology and Development*, volume 1, pages 545–549. IEEE, 2009.

[LHZ03] Charles X Ling, Jin Huang, and Harry Zhang. Auc: a better measure than accuracy in comparing learning algorithms. In *Conference of the canadian society for computational studies of intelligence*, pages 329–341. Springer, 2003.

[LJV13] Simon Liljeström, Patrik N Juslin, and Daniel Västfjäll. Experimental evidence of the roles of music choice, social context, and listener personality in emotional reactions to music. *Psychology of Music*, 41(5):579–599, 2013.

[LMB$^+$14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European conference on computer vision*, 2014.

[LMJBVR$^+$21] Álvaro Lozano Murciego, Diego M Jiménez-Bravo, Adrián Valera Román, Juan F De Paz Santana, and María N Moreno-García. Context-aware recommender systems in the music domain: A systematic literature review. *Electronics*, 10(13):1555, 2021.

[LS99] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[LSEM12] Justin J Levandoski, Mohamed Sarwat, Ahmed Eldawy, and Mohamed F Mokbel. Lars: A location-aware recommender system. In *IEEE 28th international conference on data engineering*, pages 450–461, 2012.

[LSL17] Yuncheng Li, Yale Song, and Jiebo Luo. Improving pairwise ranking for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[LYC11] Yu-Ching Lin, Yi-Hsuan Yang, and Homer H Chen. Exploiting online music tags for music emotion classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 7(1):1–16, 2011.

[LYM04] Fang Liu, Clement Yu, and Weiyi Meng. Personalized web search for improving retrieval effectiveness. *IEEE Transactions on knowledge and data engineering*, 16(1):28–40, 2004.

[McC99] Andrew McCallum. Multi-label text classification with a mixture model trained by em. In *Proceedings of the AAAI workshop on Text Learning*, 1999.

[MCCL$^+$17] Jan Margeta, Antonio Criminisi, R Cabrera Lozoya, Daniel C Lee, and Nicholas Ayache. Fine-tuned convolutional neural nets for cardiac mri acquisition plane recognition. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 5(5):339–349, 2017.

[MDILA12] David Martín, Diego López De Ipiña, Carlos Lamsfus, and Aurkene Alzua. Situation-driven development: a methodology for the development of context-aware systems. In *International Conference on Ubiquitous Computing and Ambient Intelligence*, pages 241–248. Springer, 2012.

[Mey12] Frank Meyer. Recommender systems in industrial contexts. *arXiv preprint arXiv:1203.4487*, 2012.

[Mil95] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[MSD17] Saumitra Mishra, Bob L Sturm, and Simon Dixon. Local interpretable model-agnostic explanations for music content analysis. In *ISMIR*, pages 537–543, 2017.

[MSD18] Saumitra Mishra, Bob L Sturm, and Simon Dixon. Understanding a deep machine listening model through feature inversion. In *ISMIR*, pages 755–762, 2018.

[Mül15] Meinard Müller. *Fundamentals of music processing: Audio, analysis, algorithms, applications.* Springer, 2015.

[MvNL10] Bart Moens, Leon van Noorden, and Marc Leman. D-jogger: Syncing music with walking. In *7th Sound and Music Computing Conference*, pages 451–456. Universidad Pompeu Fabra, 2010.

[MYLK08] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940, 2008.

[Neb98] Claus Nebauer. Evaluation of convolutional neural networks for visual recognition. *IEEE transactions on neural networks*, 9(4):685–696, 1998.

[NH96a] Adrian C North and David J Hargreaves. The effects of music on responses to a dining area. *Journal of environmental psychology*, 16(1):55–64, 1996.

[NH96b] Adrian C North and David J Hargreaves. Responses to music in aerobic exercise and yogic relaxation classes. *British Journal of Psychology*, 87(4):535–547, 1996.

[NH96c] Adrian C North and David J Hargreaves. Situational influences on reported musical preference. *Psychomusicology: A Journal of Research in Music Cognition*, 15(1-2):30, 1996.

[NR04] Quang Nhat Nguyen and Francesco Ricci. User preferences initialization and integration in critique-based mobile recommender systems. In *Proceedings of the 5th International Workshop on Artificial Intelligence in Mobile Systems,(AIMS'04)*, 2004.

[ON15] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[PB07] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[PCL$^+$16] Roberto Pagano, Paolo Cremonesi, Martha Larson, Balázs Hidasi, Domonkos Tikk, Alexandros Karatzoglou, and Massimo Quadrana. The contextual turn: From context-aware to context-driven recommender systems. In *Proceedings of the 10th ACM conference on recommender systems*, pages 249–252, 2016.

[PFOT19] Loreto Parisi, Simone Francia, Silvio Olivastri, and Maria Stella Tavella. Exploiting synchronized lyrics and vocal features for music emotion detection. *arXiv preprint arXiv:1901.04831*, 2019.

[PG13] Nikolaos Polatidis and Christos K Georgiadis. Recommender systems: The importance of personalization in e-business environments. *International Journal of E-Entrepreneurship and Innovation (IJEEI)*, 4(4):32–46, 2013.

[PKA14] Yannis Panagakis, Constantine L Kotropoulos, and Gonzalo R Arce. Music genre classification via joint sparse low-rank representation of audio features. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1905–1917, 2014.

[PMB13] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR, 2013.

[PP$^+$19] Jordi Pons Puig et al. *Deep neural networks for music and audio tagging.* PhD thesis, Universitat Pompeu Fabra, 2019.

[PPNP$^+$18] Jordi Pons Puig, Oriol Nieto, Matthew Prockup, Erik M Schmidt, Andreas F Ehmann, and Xavier Serra. End-to-end learning for music audio tagging at scale. In *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR*, 2018.

[PPPD01] George Prassas, Katherine C Pramataris, Olga Papaemmanouil, and Georgios J Doukidis. A recommender system for online shopping based on past customer behaviour. In *Proceedings of the 14th BLED Electronic Commerce Conference, BLED*, volume 1, pages 766–782, 2001.

[PS17] Jordi Pons and Xavier Serra. Designing efficient architectures for modeling temporal features with convolutional neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2472–2476, 2017.

[PSG+17] Jordi Pons, Olga Slizovskaia, Rong Gong, Emilia Gómez, and Xavier Serra. Timbre analysis of music audio signals with convolutional neural networks. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 2744–2748. IEEE, 2017.

[PSK+16] Sankaran Panchapagesan, Ming Sun, Aparna Khare, Spyros Matsoukas, Arindam Mandal, Björn Hoffmeister, and Shiv Vitaladevuni. Multi-task learning and weighted cross-entropy for dnn-based keyword spotting. In *Proceddings of Interspeech*, 2016.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[PTC+] Olivier Petit, Nicolas Thome, Arnaud Charnoz, Alexandre Hostettler, and Luc Soler. Handling Missing Annotations for Semantic Segmentation with Deep ConvNets. Technical report.

[PTG08] Cosimo Palmisano, Alexander Tuzhilin, and Michele Gorgoglione. Using context to improve predictive modeling of customers in personalization applications. *IEEE transactions on knowledge and data engineering*, 20(11):1535–1549, 2008.

[PTG+09] Umberto Panniello, Alexander Tuzhilin, Michele Gorgoglione, Cosimo Palmisano, and Anto Pedone. Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems. In *Proceedings of the third ACM conference on Recommender systems*, pages 265–268, 2009.

[PTM+19] Minsu Park, Jennifer Thom, Sarah Mennicken, Henriette Cramer, and Michael Macy. Global music streaming data reveal diurnal and seasonal patterns of affective preference. *Nature human behaviour*, 3(3):230–236, 2019.

[PZS15] Martin Pichl, E. Zangerle, and G. Specht. Towards a Context-Aware Music Recommendation Approach: What is Hidden in the Playlist Name? In *Proceddings of International Conference on Data Mining Workshop (ICDMW)*, 2015.

[PZS16] Martin Pichl, Eva Zangerle, and Günther Specht. Understanding playlist creation on music streaming platforms. In *IEEE International Symposium on Multimedia (ISM)*, pages 475–480, 2016.

[QCJ18] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. Sequence-aware recommender systems. *ACM Computing Surveys (CSUR)*, 51(4):1–36, 2018.

[QHR+07] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, Tao Mei, and Hong-Jiang Zhang. Correlative multi-label video annotation. In *Proceedings of the 15th ACM international conference on Multimedia*, 2007.

[QM02] Luz M Quiroga and Javed Mostafa. An experiment in building profiles in information filtering: the role of context of user relevance feedback. *Information processing & management*, 38(5):671–694, 2002.

[R+03] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48, 2003.

[RCL$^+$19] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten De Rijke. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[RD19] Shaina Raza and Chen Ding. Progress in context-aware recommender systems—an overview. *Computer Science Review*, 31:84–97, 2019.

[RDN06] Mohammed Abhur Razzaque, Simon Dobson, and Paddy Nixon. Categorization and modelling of quality in context information. 2006.

[Ren12] Steffen Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):1–22, 2012.

[RGL11] Peter J Rentfrow, Lewis R Goldberg, and Daniel J Levitin. The structure of musical preferences: a five-factor model. *Journal of personality and social psychology*, 100(6):1139, 2011.

[RHW86] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[RIS$^+$94] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, 1994.

[RM51] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[RM06] Sasank Reddy and Jeff Mascia. Lifetrak: music in tune with your life. In *Proceedings of the 1st ACM international workshop on Human-centered multimedia*, pages 25–34, 2006.

[Ros58] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[RPM98] Nick S Ryan, Jason Pascoe, and David R Morse. Enhanced reality fieldwork: the context-aware archaeological assistant. In *Computer applications in archaeology*. Tempus Reparatum, 1998.

[RRS11] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

[RV97] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[RWP81] James A Russell, Lawrence M Ward, and Geraldine Pratt. Affective quality attributed to environments: A factor analytic study. *Environment and behavior*, 13(3):259–288, 1981.

[SB14] Alan Said and Alejandro Bellogín. Comparative recommender system evaluation: benchmarking recommendation frameworks. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 129–136, 2014.

[SBS11] Chris Sumner, Alison Byers, and Matthew Shearing. Determining personality traits & privacy concerns from facebook activity. *Black Hat Briefings*, 11(7):197–221, 2011.

[Sch19] Markus Schedl. Deep learning in music recommendation systems. *Frontiers in Applied Mathematics and Statistics*, 5:44, 2019.

[SCZZ19] Shiliang Sun, Zehui Cao, Han Zhu, and Jing Zhao. A survey of optimization methods from a machine learning perspective. *IEEE transactions on cybernetics*, 50(8):3668–3681, 2019.

[SFHS07] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.

[SFU13] Markus Schedl, Arthur Flexer, and Julián Urbano. The neglected user in music information retrieval research. *Journal of Intelligent Information Systems*, 41(3):523–539, 2013.

[SIVA17] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2017.

[SK09] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 2009.

[SKKR00] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, pages 158–167, 2000.

[SKR99] J Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166, 1999.

[SL09] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009.

[SLC$^+$17] V Subramaniyaswamy, R Logesh, M Chandrashekhar, Anirudh Challa, and V Vijayakumar. A personalised movie recommendation system based on collaborative filtering. *International Journal of High Performance Computing and Networking*, 10(1-2):54–63, 2017.

[Slo99] John A Sloboda. Everyday uses of music listening: A preliminary study. *Music, mind and science*, pages 354–369, 1999.

[SM95] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating “word of mouth”. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, 1995.

[SM13] Meenakshi Sharma and Sandeep Mann. A survey of recommender systems: approaches and limitations. *International Journal of Innovations in Engineering and Technology*, 2(2):8–14, 2013.

[Smi07] John R Smith. The real problem of bridging the “semantic gap”. In *International Workshop on Multimedia Content Analysis and Mining*, pages 16–17. Springer, 2007.

[SNA19] Federico Simonetta, Stavros Ntalampiras, and Federico Avanzini. Multimodal music information processing and retrieval: Survey and future challenges. In *2019 International Workshop on Multilayer Music Representation and Processing (MMRP)*, pages 10–18, 2019.

[SOI01] John A Sloboda, Susan A O’Neill, and Antonia Ivaldi. Functions of music in everyday life: An exploratory study using the experience sampling method. *Musicae scientiae*, 5(1):9–32, 2001.

[Son99] Diane H Sonnenwald. Evolving perspectives of human information behavior: Contexts, situations, social networks and information horizons. In *Exploring the contexts of information behavior: Proceedings of the Second International Conference in Information Needs*. Taylor Graham, 1999.

[SS14] Markus Schedl and Dominik Schnitzer. Location-aware music artist recommendation. In *International conference on multimedia modeling*, pages 205–213. Springer, 2014.

[SSH17] Carl Southall, Ryan Stables, and Jason Hockman. Automatic drum transcription for polyphonic recordings using soft attention mechanisms and convolutional neural networks. 2017.

[ST94] Bill N Schilit and Marvin M Theimer. Disseminating active map information to mobile hosts. *IEEE network*, 8(5):22–32, 1994.

[STV11] Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. On the stratification of multi-label data. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2011.

[SVN37] Stanley Smith Stevens, John Volkmann, and Edwin Broomell Newman. A scale for the measurement of the psychological magnitude pitch. *The journal of the acoustical society of america*, 8(3):185–190, 1937.

[SWH$^+$18] Guangxiao Song, Zhijie Wang, Fang Han, Shenyi Ding, and Muhammad Ather Iqbal. Music auto-tagging using deep recurrent neural networks. *Neurocomputing*, 292:104–110, 2018.

[SZC$^+$18] Markus Schedl, Hamed Zamani, Ching-Wei Chen, Yashar Deldjoo, and Mehdi Elahi. Current challenges and visions in music recommender systems research. *International Journal of Multimedia Information Retrieval*, 7(2):95–116, 2018.

[TBL08] Douglas Turnbull, Luke Barrington, and Gert RG Lanckriet. Five approaches to collecting tags for music. In *Ismir*, volume 8, pages 225–230, 2008.

[TBLY09] Douglas R Turnbull, Luke Barrington, Gert Lanckriet, and Mehrdad Yazdani. Combining audio content and social context for semantic music discovery. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 387–394, 2009.

[TDS$^+$09] Grigorios Tsoumakas, Anastasios Dimou, Eleftherios Spyromitros, Vasileios Mezaris, Ioannis Kompatsiaris, and Ioannis Vlahavas. Correlation-based pruning of stacked binary relevance models for multi-label learning. In *Proceedings of the 1st international workshop on learning from multi-label data*, 2009.

[TK07] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

[TKV09] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer, 2009.

[TSG$^+$16] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312, 2016.

[Vah17] Arash Vahdat. Toward robustness against label noise in training deep discriminative neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*, 2017.

[VDH$^+$18] Michail Vlachos, Celestine Dünner, Reinhard Heckel, Vassilios G Vassiliadis, Thomas Parnell, and Kubilay Atasu. Addressing interpretability and cold-start in matrix factorization for recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 31(7):1253–1266, 2018.

[VDODS13] Aäron Van Den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Neural Information Processing Systems Conference (NIPS 2013)*, volume 26. Neural Information Processing Systems Foundation (NIPS), 2013.

[VEzD$^+$17] Andreu Vall, Hamid Eghbal-zadeh, Matthias Dorfer, Markus Schedl, and Gerhard Widmer. Music playlist continuation by learning from hand-curated examples and song features: Alleviating the cold-start problem for rare and out-of-set songs. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*, 2017.

[WAB$^+$17] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, 2017.

[WCNS20] Minz Won, Sanghyuk Chun, Oriol Nieto, and Xavier Serrc. Data-driven harmonic filters for audio representation learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[WCS19] Minz Won, Sanghyuk Chun, and Xavier Serra. Toward interpretable music tagging with self-attention. *arXiv preprint arXiv:1906.04972*, 2019.

[WCW$^+$19] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z Sheng, Mehmet Orgun, and Defu Lian. A survey on session-based recommender systems. *arXiv preprint arXiv:1902.04864*, 2019.

[Web02] Peter Webster. Historical perspectives on technology and music. *Music Educators Journal*, 89(1):38–43, 2002.

[WLW$^+$14] Baoyuan Wu, Zhilei Liu, Shangfei Wang, Bao-Gang Hu, and Qiang Ji. Multi-label learning with missing labels. In *Proceedings of 22nd International Conference on Pattern Recognition*, 2014.

[WRW12] Xinxi Wang, David Rosenblum, and Ye Wang. Context-aware mobile music recommendation for daily activities. In *Proceedings of the 20th ACM international conference on Multimedia*, 2012.

[WSBT11] Byron C Wallace, Kevin Small, Carla E Brodley, and Thomas A Trikalinos. Class imbalance, redux. In *Proceedings of the IEEE 11th international conference on data mining*, 2011.

[WW14] Xinxi Wang and Ye Wang. Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 627–636, 2014.

[WYM$^+$16] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[Wys17] Lonce Wyse. Audio spectrogram representations for processing with convolutional neural networks. *arXiv preprint arXiv:1706.09559*, 2017.

[XNH$^+$18] Miao Xu, Gang Niu, Bo Han, Ivor W Tsang, Zhi-Hua Zhou, and Masashi Sugiyama. Matrix co-completion for multi-label classification with missing features and labels. *arXiv preprint arXiv:1805.09156*, 2018.

[YJKD14] Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit Dhillon. Large-scale multi-label learning with missing labels. In *International conference on machine learning*, 2014.

[YLLH17] Karthik Yadati, Cynthia CS Liem, Martha Larson, and Alan Hanjalic. On the automatic identification of music for common activities. In *Proceedings of the 2017 ACM on international conference on multimedia retrieval*, pages 192–200, 2017.

[YYLL11] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 325–334, 2011.

[ZBH$^+$16] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[Zei12] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[ZMB13] Yong Zheng, Bamshad Mobasher, and Robin D Burke. The role of emotions in context-aware recommendation. *Decisions@ RecSys*, 2013:21–28, 2013.

[ZPGS14] Eva Zangerle, Martin Pichl, Wolfgang Gassler, and Günther Specht. # nowplaying music dataset: Extracting listening behavior from twitter. In *Proceedings of the first international workshop on internet-scale multimedia management*, pages 21–26, 2014.

[ZSZ$^+$17] Zongwei Zhou, Jae Shin, Lei Zhang, Suryakanth Gurudu, Michael Gotway, and Jianming Liang. Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[ZTYS18] Shuai Zhang, Yi Tay, Lina Yao, and Aixin Sun. Next item recommendation with self-attention. *arXiv preprint arXiv:1808.06414*, 2018.

[ZZ13] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2013.

List of Figures

List of Tables

Titre : L'étiquetage Automatique Personnalisé comme Substitut à la Recommandation Musicale Contextuelle

Mots clés : Auto-tagging musical, Context-aware, Recommandation musicale

Résumé : La croissance exponentielle des services en ligne et des données des utilisateurs a changé la façon dont nous interagissons avec divers services, et la façon dont nous explorons et sélectionnons de nouveaux produits. Par conséquent, il existe un besoin croissant de méthodes permettant de recommander les articles appropriés pour chaque utilisateur. Dans le cas de la musique, il est plus important de recommander les bons éléments au bon moment. Il est bien connu que le contexte, c'est-à-dire la situation d'écoute des utilisateurs, influence fortement leurs préférences d'écoute. C'est pourquoi le développement de systèmes de recommandation fait l'objet d'une attention croissante. Les approches les plus récentes sont des modèles basés sur les séquences qui visent à prédire les pistes de la prochaine session en utilisant les informations contextuelles disponibles. Cependant, ces approches ne sont pas faciles à interpréter et ne permettent pas à l'utilisateur de s'impliquer. De plus, peu d'approches précédentes se sont concentrées sur l'étude de la manière dont le contenu audio est lié à ces influences situationnelles et, dans une moindre mesure, sur l'utilisation du contenu audio pour fournir des recommandations contextuelles. Par conséquent, ces approches souffrent à la fois d'un manque d'interprétabilité.

Dans cette thèse, nous étudions le potentiel de l'utilisation du contenu audio principalement pour désambiguïser les situations d'écoute, fournissant une voie pour des recommandations interprétables basées sur la situation.

Title : Personalized Audio Auto-tagging as Proxy for Contextual Music Recommendation

Keywords : Music Auto-tagging, Context-aware, Music Recommendation

Abstract : The exponential growth of online services and user data changed how we interact with various services, and how we explore and select new products. Hence, there is a growing need for methods to recommend the appropriate items for each user. In the case of music, it is more important to recommend the right items at the right moment. It has been well documented that the context, i.e. the listening situation of the users, strongly influences their listening preferences. Hence, there has been an increasing attention towards developing recommendation systems. State-of-the-art approaches are sequence-based models aiming at predicting the tracks in the next session using available contextual information. However, these approaches lack interpretability and serve as a hit-or-miss with no room for user involvement. Additionally, few previous approaches focused on studying how the audio content relates to these situational influences, and even to a less extent making use of the audio content in providing contextual recommendations. Hence, these approaches suffer from lack of interpretability.

In this dissertation, we study the potential of using the audio content primarily to disambiguate the listening situations, providing a pathway for interpretable recommendations based on the situation.

Institut Polytechnique de Paris
91120 Palaiseau, France