LEVERAGING USER-GENERATED CONTENT WITH LATENT SEMANTIC INDEXING FOR COLD-START PLAYLIST CONTINUATION

A dissertation submitted for the degree of

Doctor of Philosophy

Ali YÜREKLİ
Eskişehir, 2021

LEVERAGING USER-GENERATED CONTENT WITH LATENT SEMANTIC INDEXING FOR COLD-START PLAYLIST CONTINUATION

Ali YÜREKLİ

PhD Dissertation

Department of Computer Engineering

Supervisor: Prof. Dr. Cihan KALELİ

Eskişehir

Eskişehir Technical University

Institute of Graduate Programs

October 2021

This thesis has been supported by the project 20ADP172, which is approved by the SRP Committee.

ABSTRACT

LEVERAGING USER-GENERATED CONTENT WITH LATENT SEMANTIC INDEXING FOR COLD-START PLAYLIST CONTINUATION

Ali YÜREKLİ

Department of Computer Engineering

Eskişehir Technical University, Institute of Graduate Programs, October 2021

Supervisor: Prof. Dr. Cihan KALELİ

Music recommender systems, which are indispensable components of online music streaming services, aim to provide better music listening experience to their users by providing personalization in line with users' musical preferences. An important requirement in achieving this goal is the automatic continuation of playlists in a coherent manner. A music recommender system should identify the characteristics of a playlist, create a user profile based on these characteristics, and recommend music suitable for this profile. This requirement, which is quite challenging due to the huge size of music catalogs, becomes even more complex for new playlists that do not yet contain any tracks. In such cold-start cases, the use of additional information to reveal possible music themes in playlist is essential.

In this dissertation, playlist naming tendencies of users consuming music via the Internet are analyzed. Furthermore, it is investigated how the common attitude in playlist organization can be utilized in music recommender systems. Based on the findings obtained from a large dataset containing one million different playlists, a novel recommendation approach using latent semantic indexing is proposed to alleviate the cold-start problem in automatic playlist continuation. The experimental studies and results show that the proposed method can produce recommendations with higher accuracy than the state-of-the-art approaches in the literature for the addressed cold-start problem. The statistically significant improvements on a large, real-world dataset reveal that latent semantic indexing can effectively discover hidden relationships between user-generated titles and music preference.

Keywords: Music recommender systems, Automatic playlist continuation, Cold-start problem, Information retrieval, Latent semantic indexing.

ÖZET

SOĞUK-BAŞLANGIÇ ÇALMA LİSTELERİNİN SÜRDÜRÜLMESİ İÇİN KULLANICI TANIMLI İÇERİKLER ÜZERİNE ÖRTÜLÜ ANLAMSAL İNDEKSLEME

Ali YÜREKLİ

Bilgisayar Mühendisliği Anabilim Dalı

Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Ekim 2021

Danışman: Prof. Dr. Cihan KALELİ

Çevrimiçi müzik servislerinin vazgeçilmez bir unsuru olan müzik öneri sistemleri, kullanıcılara bireysel tercihleri doğrultusunda kişiselleştirme sağlayarak daha iyi bir müzik dinleme deneyimi sunmayı amaçlar. Bu hedefi gerçekleştirmede önemli bir gereksinim, çalma listelerinin bir ahenk içerisinde otomatik sürdürülmesidir. Bir müzik öneri sistemi, çalma listesinin karakteristiğini tespit ederek kullanıcı profili çıkarmalı ve buna uygun şarkıları öneri olarak sunmalıdır. Müzik kataloglarının devasa boyutları sebebiyle oldukça zorlayıcı olan bu gereksinim, henüz hiç şarkı içermeyen yeni listeler için daha da karmaşık hale gelmektedir. Literatürde soğuk-başlangıç olarak bilinen bu durumlarda çalma listelerindeki temaları ortaya çıkaracak ek bilgilere ihtiyaç duyulur.

Bu tez kapsamında, İnternet ortamında müzik dinleyen kullanıcıların çalma listeleri için ne tür isimlendirmeler tercih ettiği incelenmiş ve çalma listesi organizasyonundaki ortak tutumlardan müzik önerileri sistemlerinde nasıl faydalanılabileceği araştırılmıştır. Bir milyon farklı çalma listesi içeren büyük bir veri seti üzerinden elde edilen bulgulara dayanarak örtülü anlamsal indeksleme tabanlı yeni bir müzik öneri yaklaşımı öne sürülmüş ve bu yaklaşım doğrultusunda kurgulanan bilgi erişimi sistemi ile otomatik çalma listesi sürdürmede soğuk-başlangıç problemine çözüm aranmıştır. Yapılan deneysel çalışmalar, öne sürülen yöntemin soğuk-başlangıç problemine yönelik literatürdeki mevcut yaklaşımlardan daha yüksek doğrulukta öneriler üretebildiğini göstermektedir. Gerçek ve büyük veri üzerinde istatiksel olarak anlamlı bu iyileştirme, örtülü anlamsal indekslemenin çalma listeleri başlıkları ve kullanıcıların müzik zevkleri arasındaki gizli ilişkileri etkin bir şekilde keşfedebildiğini ortaya koymaktadır.

Anahtar Sözcükler: Müzik öneri sistemleri, Otomatik çalma listesi sürdürme, Soğuk-başlangıç problemi, Bilgi erişimi, Örtülü anlamsal indeksleme.

ACKNOWLEDGEMENTS

I would like to express my sincerest gratitude to my esteemed supervisor, Prof. Dr. Cihan KALELİ, who has supported me throughout my dissertation with his wisdom, patience, and unsurpassed knowledge. It has been an honor for me to work with him.

Furthermore, I would like to thank my respectful colleagues on my dissertation committee for their valuable contributions. I owe a lot to their guidance in both academic and personal levels.

I would love to present my deepest appreciation to my dear parents and sister, and all my beloved family members for their love, support, and understanding during my long journey. The moments we share bring us to this point. Thank you all.

Finally, a special gratitude goes to my lovely wife, Ece. No matter what life brings us, you have always been solid as a rock. Thank you, I am so glad that I have you in my life.

Ali YÜREKLİ

11/10/2021

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with "scientific plagiarism detection program" used by Eskişehir Technical University, and that "it does not have any plagiarism" whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

Ali YÜREKLİ

CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

GLOSSARY OF SYMBOLS AND ABBREVIATIONS

APC	:	Automatic Playlist Continuation
ATNN	:	Adaptive Thresholding Nearest Neighbors
CARS	:	Context-Aware Recommender Systems
CBF	:	Content-Based Filtering
CF	:	Collaborative Filtering
DCG	:	Discounted Cumulative Gain
GTF	:	Global Track Frequency
ICF	:	Inverse Cluster Frequency
IDCG	:	Ideal Discounted Cumulative Gain
IR	:	Information Retrieval
KNN	:	K-Nearest Neighbors
LOOCV	:	Leave-One-Out Cross-Validation
LSI	:	Latent Semantic Indexing
MPD	:	The Million Playlist Dataset
MRS	:	Music Recommender Systems
MWTF	:	Maximized Weighted Track Frequency
NDCG	:	Normalized Discounted Cumulative Gain
NER	:	Named Entity Recognition
NLP	:	Natural Language Processing
NTP	:	Normalized Title Proximity
SVD	:	Singular Value Decomposition
TF	:	Track Frequency
TF*ICF	:	Track Frequency * Inverse Cluster Frequency
TNN	:	Thresholding Nearest Neighbors
TP	:	Title Proximity
WTF	:	Weighted Track Frequency

1. INTRODUCTION

It is possible to invent a single machine which can be used to compute any computable sequence.

- Alan Turing

In today's modern information age, one of the most vital problems of the Internet users is to make the right decision among excessive choices. An individual, who performs various activities such as watching a movie, ordering food, or booking a holiday hotel on online platforms, is surrounded by too many options that she cannot manage on her own. In order to alleviate this overwhelming situation, which is also called the information overload (Bollen et al., 2010), recommender systems have been widely utilized since the last few decades.

A recommender system is an information filtering mechanism that aims to expedite the decision-making process for users by providing a fast, accurate, and easy access to interesting products (Schafer et al., 2001). In simple words, a recommender system builds up a profile for a target user and filters relevant items based on this personalized user profile. This working principle mainly relies on users' past behaviors such as purchased items, given ratings, and likes or dislikes.

Playing an undeniable role in our online journey, recommender systems have been used in a variety of areas. Accordingly, recommendation items can be of various types. Since the earlier traditional versions, the application areas of recommender systems have expanded to a wide spectrum covering many kinds of products and services, including, but not limited to, movies, TV series, books, news, music, games, food, job advertisements, social relations, and friendships (Aggarwal, 2016). Specifically, this dissertation considers music and its elements, such as tracks, albums, artists, and genres, as the primary recommendation objects and investigates the cold-start problem, which is a well-known and common issue across recommender systems (Bobadilla et al., 2012), from the perspective of music recommendation.

1.1. Music Recommender Systems

With the emergence of online music streaming platforms like Spotify[1], Apple Music[2], and Deezer[3], music listeners acquire immediate access to almost infinite music catalogs. As a plausible way of addressing the information overload, music recommender systems (MRS) provide guidance to users when navigating these large collections. Since music differs from traditional recommendation items (e.g., movies, books, and news) in various aspects, such as emotional connotation and sequential consumption (Schedl et al., 2015), MRS should be specialized in terms of the domain-specific requirements and constraints to fulfil music listeners' needs.

In order to handle the idiosyncratic particularities of music as a recommendation item, various approaches have been proposed in the field of MRS. Collaborative filtering (CF) techniques commonly used across recommender systems are domain-agnostic and can be easily applied to music ratings data (Slaney & White, 2007). However, huge music catalogs deepen and exacerbate the data sparsity problem encountered in CF techniques. A sparse user-item matrix results in a diminished CF recommendation performance. For this reason, the need for techniques other than CF has been increasing. Schedl et al. (2015) investigate renovative music recommendation approaches in the literature in three major groups, which are content-based recommendation, contextual recommendation, and hybrid recommendation.

In music recommendation, content-based filtering (CBF) techniques aim to extract semantic information describing music items from several resources such as audio signals, social tags, user demographics, and item metadata. Although this process requires a fair amount of domain knowledge, CBF techniques are highly effective in utilizing content-based descriptors to alleviate the data sparsity problem. Especially in cold-start scenarios lacking user preference data, CBF can be used to augment or replace traditional CF techniques (Bogdanov & Boyer, 2012).

Context is an important factor that strongly affects users' appreciation of music. In terms of personal music preference, both environmental context (e.g., location, weather, and time) and personal context (e.g., current activity, emotional state, mood, and culture) might be decisive and insightful (Schedl et al., 2015). Context-aware recommender

[1] https://www.spotify.com
[2] https://www.apple.com/apple-music
[3] https://www.deezer.com

systems (CARS) take these specific contextual situations of users into consideration to produce more relevant and accurate music recommendations. In a sense, CARS produce music recommendations by considering the immediate context of users as well as their long-term music preferences and tastes.

In general, hybrid approaches in recommender systems combine two or more techniques to improve recommendation performance and accuracy with fewer drawbacks of any employed method (Burke, 2002). In the field of music recommendation, the hybrid approaches mainly incorporate multiple similarity aspects into the recommendation process. During hybridization, the music descriptors are usually combined at three levels: CF with content information (Van Den Oord et al., 2013; McFee et al., 2011), CF with context information (Baltrunas et al., 2011), and CBF with context information (Wang et al., 2012).

1.2. The Grand Challenges of Music Recommender Systems

MRS can be considered as an emerging field in recommender systems research that still faces a variety of challenges. In a comprehensive study on future research directions of music recommendation, Schedl et al. (2018) identify three grand challenges of MRS as the cold-start problem, automatic continuation of music playlists, and fair evaluation of music recommendations.

1.2.1. The cold-start problem

In recommender systems, the cold-start problem is a common issue that prevents recommendation algorithms from working properly due to the lack of predictive data. New users or items, about which there is not enough information, suffer from this problem (Bobadilla et al., 2012). Although the problem might be a temporary condition (until adequate information is gathered about the target user or item), it remains an open research direction in the field of recommender systems.

As with other recommendation domains, the cold-start problem is also observed during music consumption in MRS. In general, the music catalogs are huge. This property, which offers a lot of different choices for users, implies high data sparsity for MRS. In a study comparing music and movie domains in terms of data sparsity, Dror et al. (2012) report that music ratings appear to be much sparse than movie ratings. Although several approaches, such as cross-domain recommendation, hybridization, CBF, and

active learning, have been proposed to address the cold-start problem (Schedl et al., 2018), it is still one of the grand challenges in MRS.

1.2.2. Automatic playlist continuation

One of the popular features of online music streaming services is the playlist concept. Almost all modern platforms allow their users to create and share their music playlists. According to the current 2021 reports[4] of the Spotify, which is one of the leader companies in the online music market, the platform maintains over 4 billion playlists in its catalog.

In simple words, a music playlist is a sequence of tracks. Generally, users tend to keep tracks to be listened to together in the same playlist during a music listening session. Thus, the content in a playlist is highly informing about a user's music preference. Automatic playlist continuation (APC) is the task of continuing existing playlists in a coherent manner. Nowadays, online music streaming services willing to appeal to more users (definitely a precise way to increase profit rates for companies) empower their MRS with the capability of APC. However, the way to an efficient APC is nontrivial, it requires well-understanding of playlist characteristics and user preference.

Obviously, identifying a set of tracks that fit a specific user profile (or a playlist) out of millions of options is a difficult task. Furthermore, there are many factors to consider when extending playlists, such as the order of seed tracks, user mood, personal preferences, and the environmental context. As a result of these several factors, APC can be considered as a complex challenge in the field of MRS.

1.2.3. Fair evaluation of music recommendations

The ultimate goal of MRS is to enhance user satisfaction (Bonnin & Jannach, 2015), which is a highly subjective and abstract notion that is hard to express mathematically. When a user is provided music recommendations of high-quality, the user's satisfaction with the online music streaming service is expected to increase naturally. At this point, the question of how to assess the quality of recommendations arises.

Accuracy measures, such as precision and recall, have been widely employed in the field of MRS to assess the quality of recommendations (Schedl et al., 2015). Commonly,

[4] https://newsroom.spotify.com/company-info

these measures calculate an error rate based on how much predictions and the real observations overlap and match with each other. However, this monolithic aspect might be misleading. Evaluating music recommendations according to some error rates is not a sufficient perspective on its own. For a fair evaluation of MRS, beyond-accuracy measures (Kaminskas & Bridge, 2016) should be taken into consideration as well as accuracy measures. This way, various different aspects such as novelty, diversity, and serendipity can also be subject to the evaluation of music recommendations.

Better evaluation of recommendations implies better understanding of users' music perception, which may potentially lead to considerable improvements in MRS. In line with the urgent need for metrics that melt accuracy measures and beyond-accuracy measures in the same pot, fair evaluation of music recommendations is still an open issue in the field of MRS.

1.3. Problem Definition

In its broadest definition, APC can be expressed as the task of proper extension of music playlists. The most significant information to accomplish this task is actually hidden in the playlist itself; seed tracks carry valuable signals about the intended purpose of the playlist owner (Logan, 2002). The state-of-the-art APC approaches are elegant models that are capable of identifying playlist characteristics through extensive analyses on seed tracks (Zamani et al., 2019).

Figure 1.1 depicts a typical APC process for an existing playlist with seed tracks (e.g., a warm-start playlist). Given a playlist with *n* tracks, it is possible to create a binary co-occurrence matrix where 1s represent the occurrence of a track in the given playlist and 0s indicate non-occurrence. This matrix resembles the famous user-item matrix in traditional CF systems. Analogously, while playlists represent users, tracks correspond to items. Therefore, it can be feasible to employ well-known CF techniques, such as matrix factorization (Koren et al., 2009), to detect similar playlists to the target playlist and choose frequent tracks among these playlists as final recommendations.

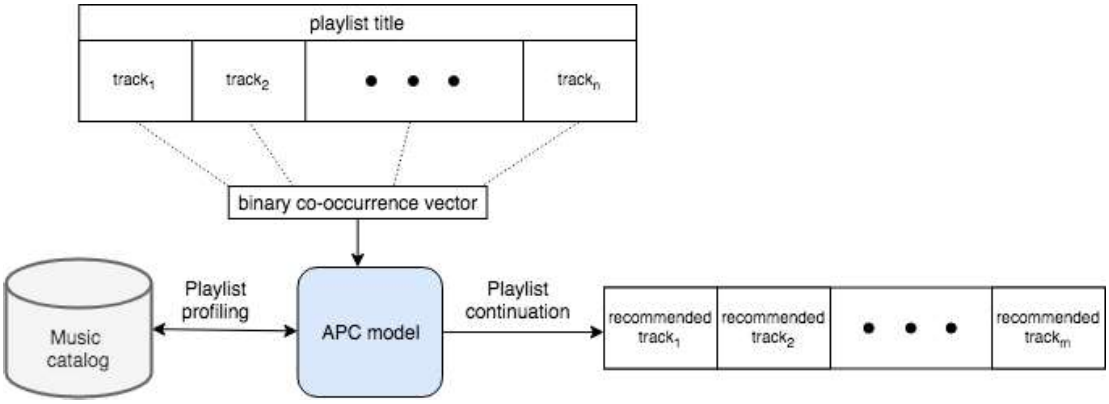

Figure 1.1. *An illustration of a typical APC process for a warm-start playlist*

Related to the data sparsity problem briefly introduced in Chapter 1.2, an extreme APC scenario is the continuation of new playlists with only title information. Unlike the example APC process illustrated in Figure 1.1, a binary co-occurrence vector cannot be obtained for a new playlist since it does not contain any tracks yet. Thus, traditional CF techniques are not applicable in this case. MRS should rely on alternative information sources, such as user-generated playlist titles and user demographics, to alleviate the resulting cold-start problem.

In this dissertation, the cold-start problem in APC is defined as follows. Each playlist essentially corresponds to a single user. In other words, it is unknown whether a user owns more than one playlist. Furthermore, there is no personal information about users, such as gender, age, and any other demographics. Under these circumstances, when a new playlist is created, it remains in a cold-start state until some tracks are added to the playlist. Until that time, the only information that can be used in this situation is the title of the playlist, which is given by the user at playlist creation. Cold-start APC seeks elegant ways to address the described problem.

1.4. Contributions

This dissertation focuses on improving the overall accuracy of recommender systems against a particular challenge, which is the automatic continuation of new playlists suffering from the cold-start problem. In order to compensate the lack of predictive data causing the problem, it is aimed to make maximum use of playlist titles as an auxiliary data source.

The main contributions of the dissertation can be summarized and listed as follows.

- In order to understand the usability and effectiveness of playlist titles in cold-start APC, extensive experiments are performed on a large dataset. The experiments are conducted on a total number of one million different playlists. The most preferred titles by users are identified and the correlation between common usage and significance in terms of MRS are investigated. Furthermore, titles that are of great importance in music recommendation are identified and the types of information hidden in these titles (e.g., genre, artist, and context) are revealed.
- Based on the findings attained from large scale analyses, a guideline to develop title-aware cold-start recommendation models is proposed. The key factors to improve the overall accuracy of cold-start recommendations are explained. In addition, a hypothetical title-aware recommender system is depicted by describing how an ideal cold-start APC approach should be. Such a system might be inspiring and enlightening for fellow researchers who are willing to work on the subject.
- An extreme problem in MRS is transformed into an information retrieval (IR) task. Upon this analogy, a multi-stage retrieval approach that utilizes user-generated titles to alleviate cold-start APC is proposed. The proposed architecture consists of four primary stages: *(i)* playlist clustering, *(ii)* latent semantics modeling, *(iii)* neighboring clusters retrieval, and *(iv)* track weighting. Several retrieval techniques are introduced to realize these stages.
- Hyperparameters of the proposed recommendation architecture are tuned to come up with an optimal final solution. This tuning phase consists of three major tasks: *(i)* determining the optimal number of dimensions for the latent semantics model, *(ii)* specifying the limits of adequate similarity between playlists, and *(iii)* determining the optimal ranking method to provide a final list of recommendations. Since all these tasks are correlated with each other, a full-factorial experimental methodology is followed.
- The overall recommendation accuracy of cold-start APC is improved. Comparisons with the state-of-the-art approaches show that the proposed architecture produces better recommendations than the existing methods in terms of three well-known evaluation metrics. Furthermore, t-tests are

applied on the experimental results. Accordingly, it is proven that the accuracy improvements attained by the proposed system are statistically significant with confidence levels up to 99%.

1.5. Organization of the Dissertation

The rest of the dissertation is organized as follows. In Chapter 2, a comprehensive definition of cold-start APC is given and existing studies addressing the problem are described. In Chapter 3, the usability and effectiveness of user-generated titles for music recommendations are investigated. The common attitude of users in playlist organization and the types of semantic information hidden in playlist titles are explored on a large scale. In Chapter 4, the multi-stage recommendation approach that leverages latent space item representations to alleviate the cold-start problem in APC is presented. Furthermore, statistically significant accuracy improvements attained by the proposed recommender system are shown. Finally, in Chapter 5, concluding remarks and some future research directions are discussed.

2. COLD-START APC: AN EXTREME RECOMMENDATION CASE

The cold-start problem is a critical situation that can be observed across all recommender systems, regardless of the recommendation domain. The absence of predictive data to discover personal preferences of users prevents algorithms from suggesting items; this obstacle causes the cold-start problem.

Typically, the cold-start problem occurs in two different forms: new user problem (Rashid et al., 2008) and new item problem (Park & Tuzhilin, 2008). In the new user problem, there is no personalization data for a user newly registered to a system, while in the new item problem, user feedback about an item newly added to a product catalog cannot be obtained. CF techniques suffer from both forms of the cold-start problem. On the other hand, CBF approaches that utilize content information of items only encounter the new user problem.

This dissertation deals with the cold-start problem encountered during the task of APC for new playlists. In the rest of this chapter, a broad definition of cold-start APC is given, and then current studies in the literature are presented. Afterward, explanations about the dataset and evaluation metrics to be used in the upcoming experimental studies are included.

2.1. Cold-Start APC

As previously mentioned in Chapter 1.2, two of the grand challenges of MRS are the harmonious continuation of music playlists and the cold-start problem caused by the absence or sparseness of predictive data. Cold-start APC can be expressed as an extreme recommendation case involving these two major challenges. In line with its complexity, cold-start APC requires elegant solutions that can incorporate alternative data sources into the recommendation process.

In a typical playlist creation scenario, a user initializes a listening session on a music streaming platform and saves this session as a playlist for further use (e.g., adding tracks, music listening, or updating the playlist). When a new playlist is created, MRS are not aware the characteristics of the playlist or the intended purpose of the user. Therefore, music recommendation process is in a stand-by state until adequate number of tracks are added to the playlist. During this time period, no recommendations can be provided to the active user, which is an undesirable cold-start APC situation in MRS.

In the scope of this dissertation, where playlists are accepted as users and tracks are considered as items, cold-start APC becomes an interesting case of the new user cold-start problem (Yürekli et al., 2021).

2.2. Related Work

Especially in CF techniques, integration of additional data is a necessity to alleviate the cold-start problem in recommender systems (Bobadilla et al., 2012). This dissertation aims to explore the usability of playlist titles for understanding the intended purpose of new playlists.

A number of approaches have been proposed to make use of playlist titles in music recommendation. Pichl et al. (2015) create contextual clusters using titles belonging to playlists from the Spotify platform. The idea is to link close clusters together. This way, latent patterns between playlists close to each other (e.g., playlists with titles “summer”, “summer 2015”, and “my summer songs”) can be revealed. The authors are able to observe at least 33% accuracy improvement when contextual information is integrated to a CF system. In order to handle the drawbacks of contextual clustering, the same authors use factorization machines to extract information from playlist titles (Pichl et al., 2017). In another work, Chung et al. (2017) use latent space representations of music tracks and words extracted from playlist titles in a music retrieval task. The authors propose that thematic inferences attained from music queries, such as “christmas” and “punk”, have a positive impact on recommendations.

The RecSys Challenge 2018 (Chen et al., 2018), the 2018 organization of the annual recommender systems competition[5], fosters the research on APC in cooperation with Spotify. Given a large music dataset, the participants are expected to develop APC models to extend music listening sessions in a coherent way. One of the 10 different categories in this competition requires elegant solutions to address the cold-start problem. Commonly, against this particular challenge, the participants prefer to propose a separate solution utilizing playlist titles apart from their warm-start APC models (Zamani et al., 2019). The next section presents an overview of the top-performing cold-start approaches in the competition.

[5] https://recsys.acm.org/recsys18/challenge

2.2.1. The state-of-the-art approaches in cold-start APC

The RecSys Challenge 2018 involves a special category that only deals with the cold-start problem. As an emerging topic in the field of MRS, studies in cold-start APC are limited with this competition. Therefore, the approaches that are successful in the RecSys Challenge 2018 can be considered as the state-of-the-art.

Yürekli et al. 2021 examine the state-of-the-art cold-start APC approaches under four main groups according to the employed techniques: *(i)* matrix factorization, *(ii)* neural networks, *(iii)* CBF, and *(iv)* nearest neighborhood search. A complete list of these approaches is presented in Table 2.1.

Table 2.1. *The state-of-the-art cold-start APC approaches in accordance with their primary techniques*

Matrix factorization	**Neural networks**	**CBF**	**Nearest neighborhood search**
Ferraro et al. (2018)	Kim et al. (2018)	Antenucci et al. (2018)	Kallumadi et al. (2018)
Ludewig et al. (2018)	Yang et al. (2018)	Faggioli et al. (2018)	Kelen et al. (2018)
Rubtsov et al. (2018)		Kaya & Bridge (2018)	Zhao et al. (2018)
Volkovs et al. (2018)			
Zhu et al. (2018)			

Frequent use of matrix factorization across recommender systems is also observed in cold-start APC approaches. Ludewig et al. (2018) employ matrix factorization as a component of a weighted hybrid recommendation architecture. The authors use matrix factorization with alternating least squares. Rubtsov et al. (2018) extracts frequent words from titles and build a matrix factorization model on top of these words. In another work, Volkovs et al. (2018) create latent factorization representations using one-hot encoding of titles and track frequencies. Zhu et al. (2018) propose a neighbor-based CF technique that uses n-gram representations of titles for factorization. In addition to these approaches, matrix factorization has been also preferred to integrate audio signals and genre annotations into the recommendation process (Ferraro et al., 2018). In brief, the common purpose of all the above-mentioned approaches is to decompose an interaction matrix into lower dimensions so that hidden relations between tracks and titles can be revealed.

In recent years, deep learning has a great impact on recommender systems research (Batmaz et al., 2019). In terms of cold-start APC, neural networks have been mainly

employed for character-level processing of user-generated titles. From a classification perspective, Yang et al. (2018) propose a character-level convolutional neural network that computes classification scores for tracks based on titles. In another work, Kim et al. (2018) propose a recurrent neural network architecture that uses character n-gram representations of playlist titles. Both of these two approaches are novel and interesting in terms of the way they handle sequential title terms and characters.

In order to retrieve the most likely tracks for a given new playlist, a number of approaches rely on several CBF techniques. Kaya & Bridge (2018) uses a rudimentary CBF technique that aggregates playlists around normalized title texts. Antenucci et al. (2018) propose a hybrid CBF model tokenizes titles during the preprocessing stage. Similarly, Faggioli et al. (2018) employs an additional stemming step when dealing with title inputs.

One of the common cold-start APC approaches is nearest neighborhood search. Kallumadi et al. (2018) utilize a query expansion technique from the IR domain to perform similarity search on metadata. Zhao et al. (2018) propose a contextual clustering technique and estimate similarity of these clusters. In another work, Kelen et al. (2018) use an effective neighborhood search algorithm to find relevant music tracks for a particular title. All these approaches mainly aggregate playlists around title clusters and search for close tracks.

Table 2.2. *Recommendation performances of the cold-start approaches in the RecSys Challenge 2018*

Cold-start rank	Approach	NDCG@500
1	**Ludewig et al. (2018)**	**0.2053**
2	**Volkovs et al. (2018)**	**0.2044**
3	**Faggioli et al. (2018)**	**0.2031**
4	Kelen et al. (2018)	0.2001
5	Antenucci et al. (2018)	0.1961
6	Kaya & Bridge (2018)	0.1959
7	Yang et al. (2018)	0.1925
8	Rubtsov et al. (2018)	0.1881
9	Zhao et al. (2018)	0.1817
10	Kallumadi et al. (2018)	0.1814
11	Zhu et al. (2018)	0.1812
12	Ferraro et al. (2018)	0.1786
13	Kim et al. (2018)	0.1750

Table 2.2 presents recommendation performances of the state-of-the-art approaches in the RecSys Challenge 2018. The approaches are ranked in terms of an evaluation metric employed in the competition (which is later introduced in Chapter 2.4). The first three best-performing approaches are highlighted in bold. As the table illustrates, matrix factorization approaches by Ludewig et al. (2018) and Volkovs et al. (2018) produce slightly more accurate recommendations than the others. Furthermore, these scores are directly taken from the overview paper of the competition (Zamani et al., 2019). It should also be noted that the experimental settings of the cold-start category requires recommendation of exactly 500 tracks for each of the given 1000 new playlists with only title information. Although participants are allowed and encouraged to use other public data sources, only Ferraro et al. (2018) consider giving a shot to seearch for similarities of audio features.

Surprisingly, the research in cold-start APC is limited with the scope of the RecSys Challenge 2018. Despite the fact that MRS research continues at a rapid pace, there is not much work specific to the cold-start problem in APC. The two possible reasons for this fact might be the complexity of the problem and the difficulty of gathering or integrating additional data sources.

2.3. The Million Playlist Dataset

In parallel with the widespread use of online music streaming services, datasets that can be used in MRS research have been both increasing in numbers and diversifying in content. The experimental works of this dissertation are performed on the Million Playlist Dataset (MPD), which is a recent dataset in the field of music recommendation (Chen et al., 2018). The MPD was originally prepared for the RecSys Challenge 2018, but was late made public.

The MPD is one of the largest public music datasets that contains one million user-generated playlists from Spotify. Each of these playlist instances comprises of a title, a track list with album and artist information, and various other metadata fields such as number of edits and total duration. Having more than 66 million listening records of more than 2 million unique tracks, the MPD can be considered as quite rich in content. This richness is also evident by the statistics presented in Table 2.3.

Table 2.3. *Basic properties of the MPD*

Property	Value
Number of playlists	1,000,000
Number of tracks	66,346,428
Number of unique tracks	2,262,292
Number of unique albums	734,684
Number of unique artists	295,860
Number of titles	1,000,000
Number of unique titles	92,944
Number of unique normalized titles	17,381
Average playlist length	66.35

2.3.1. An overview of playlist titles in the MPD

A playlist title, which is a short description defined by the owner of the corresponding playlist, might contain valuable information to understand playlist characteristics (Pichl et al., 2015). Such information can be utilized to generate recommendations for new playlists suffering from the cold-start problem. Therefore, it would be appropriate to mention a brief overview of the playlist titles in the MPD.

Every playlist instance in the MPD has a user-generated title. In other terms, the dataset contains one million playlist titles. When these short texts are analyzed in terms of their uniqueness, it is seen that there exist 92,944 unique titles in the dataset. Since users commonly prefer similar expressions when naming their music listening sessions (Yürekli et al., 2021), it is possible to further reduce this number by applying some elementary transformations. The steps of such a normalization scheme can be defined as follows:

i. Lowercasing all letters
ii. Eliminating all whitespaces
iii. Removing predefined special characters

The naïve title normalization technique described above results in a total number of 17,381 unique normalized titles. Naturally, users prefer some titles (e.g., “country”, “rap”, and “party”) more often than the others. In Figure 2.1, some of the most frequent normalized titles in the MPD are presented for a better illustration.

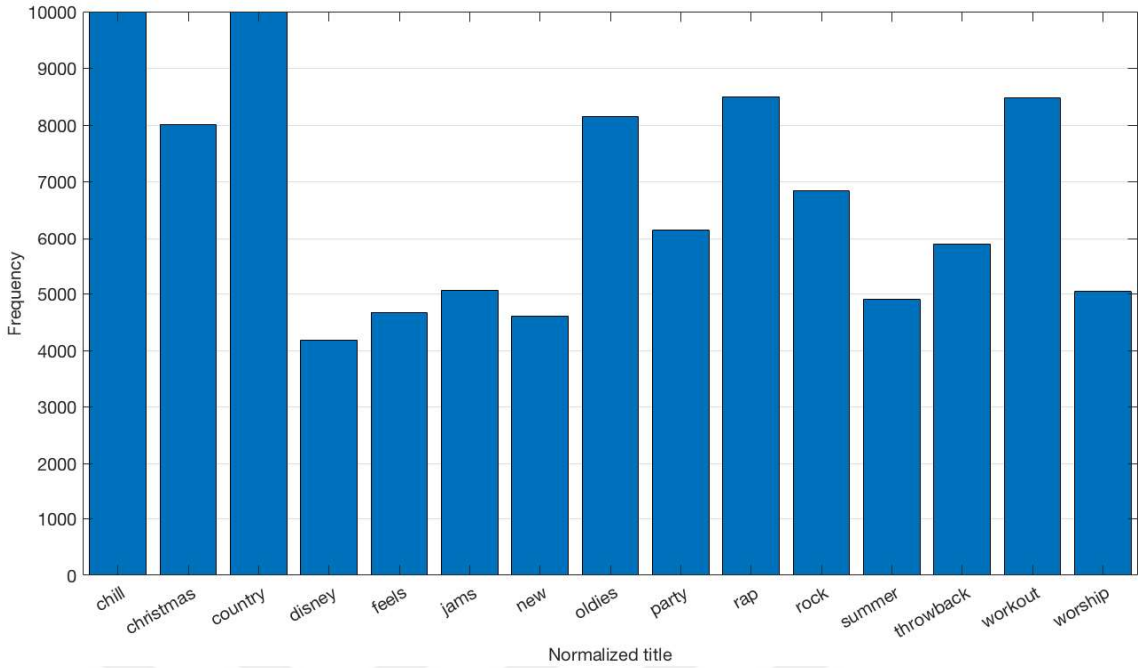

Figure 2.1. *Distribution of the most frequent normalized titles in the MPD*

The abovementioned properties of the MPD, especially the rich variety of content and titles, make it appropriate for a comprehensive research on novel approaches to the cold-start problem in APC. Therefore, this dissertation conducts extensive experiments on this dataset.

2.4. Evaluation Criteria

Assessing the quality of recommendations is a critical task across recommender systems. Especially in a subjective field such as music recommendation, this task becomes quite challenging (Schedl et al., 2018). In this dissertation, three well-known evaluation metrics are employed for a fair assessment of recommendation accuracy in cold-start APC. Assuming that G and R denote the ground truth and recommendations for a target playlist, respectively, the employed metrics can be expressed as follows.

i. **R-precision** assesses recommendation performance by rewarding the accurate predictions within the length of a target playlist (Zamani et al., 2019). R-precision is computed as given in Eq. (2.1). The metric was used for the first time in this form in the RecSys Challenge 2018.

$$\text{R-precision} = \frac{\left|G \cap R_{1:|G|}\right|}{|G|} \tag{2.1}$$

ii. **NDCG (Normalized Discounted Cumulative Gain)** is a cumulated gain-based metric that is mostly used to evaluate the effectiveness of IR systems (Järvelin & Kekäläinen, 2002). In recommender systems domain, NDCG assesses the ranking quality of recommendations and is calculated by dividing Discounted Cumulative Gain (DCG) by Ideal Discounted Cumulative Gain (IDCG). Assuming that rel_i is the relevancy label for a track ranked at position i for a target playlist, then NDCG is computed as follows.

$$\text{DCG} = rel_1 + \sum_{i=2}^{|R|} \frac{rel_i}{\log_2(i+1)} \tag{2.2}$$

$$\text{IDCG} = 1 + \sum_{i=2}^{|G|} \frac{1}{\log_2(i+1)} \tag{2.3}$$

$$\text{NDCG} = \frac{DCG}{IDCG} \tag{2.4}$$

iii. **Recall** is a common IR metric that evaluates model ability in terms of finding relevant items and calculated as given in Eq. (2.5). Since this metric is set-based, it does not take recommendation order into account.

$$\text{Recall} = \frac{|G \cap R|}{|G|} \tag{2.5}$$

For all the three evaluation metrics, higher scores indicate better recommendation accuracy. In addition, while R-precision and NDCG also consider the actual order of recommendations, there is no such criterion in Recall since it is a set-based evaluation metric.

3. EXPLORING THE HIDDEN CONTENT IN PLAYLIST TITLES

A common approach to the cold-start problem in recommender systems is the integration of additional data into the recommendation model (Bobadilla et al., 2012), which might compensate the lack of predictive data. This chapter investigates the effectiveness of user-generated titles as an auxiliary data source for cold-start APC. Firstly, three naïve recommendation models employed to continue cold-start playlists are introduced. Then, the methodology that enables testing on the entire dataset at a large scale is explained. Finally, the experimental results, corresponding analyses, and insights are presented.

3.1. Naïve Cold-Start Recommendation Models

In order to investigate the effectiveness of user-generated titles in cold-start APC and explore the hidden patterns in this content, three naïve recommendation models are proposed in line with the following strategies.

i. Not using any title information
ii. Utilizing titles directly, without any preprocessing
iii. Utilizing titles with their syntactic similarities

3.1.1. Global track frequency

A simple solution to mitigate the cold-start problem for new playlists might be recommending popular music among other playlists. Global track frequency (GTF) is the approach that realizes this rudimentary idea by calculating track frequencies in the entire MPD. The top tracks by frequency form the list of recommendations for cold-start playlists. The score of a track *t* is computed as given in Eq. (3.1).

$$score_{GTF}(t) = \sum_{p \in P} (t,p) \quad where \ \ (t,p) = \begin{cases} 1 & if \ t \in p \\ 0 & otherwise \end{cases} \tag{3.1}$$

where *P* denotes the set of all playlists in the dataset.

GTF does not utilize any title information; the same list of tracks is recommended to all cold-start instances. In fact, GTF can be considered as a complementary technique where title-based approaches fail to generate adequate number of recommendations due to unusual title content (Ferraro et al., 2018; Ludewig et al., 2018). Nevertheless, GTF provides an accuracy baseline to illustrate the usability of user-generated titles in cold-start approaches.

3.1.2. Title proximity

Based on the point of origin that playlists with the same titles may contain similar music preferences (Schedl et al., 2018), it is possible to develop recommendation models that reduce the track search space to playlists matching with the title of a given cold-start instance. Title proximity (TP) is the approach that computes track frequencies among playlists matching with a certain title. Denoting the set of all playlists in the MPD as P and the target playlist as c, the score of a track t is calculated as shown in Eq. (3.2).

$$score_{TP}(t) = \sum_{p \in P} (t, p, c) \quad where \quad (t, p, c) = \begin{cases} 1 & if\ t \in p \wedge title_p = title_c \\ 0 & otherwise \end{cases} \tag{3.2}$$

3.1.3. Normalized title proximity

Compared to formal languages, the language of user-oriented media is less standardized; it is very common not to conform to the rules of spelling, grammar, and punctuation in social media conversations (Clark & Araki, 2011). A similar situation is observed when users name their music listening sessions. Often playlist titles express the same thing, although they are spelled in many different ways. As an example of this fact, various title spellings denoting "summer 2017" in the MPD are presented in Figure 3.1.

Figure 3.1. *Sample title spellings for "summer 2017" in the MPD*

Normalized title proximity (NTP) is the recommendation model that aims to benefit from syntactic similarities of title texts. Employing the naïve title normalization technique, titles are transformed into more homogenous forms. The calculation of track scores is almost identical to the calculation procedure in TP. The only difference is that

NPT looks for the matchings of normalized playlist titles. Eq. (3.3) shows how the score of a track *t* is computed.

$$score_{NTP}(t) = \sum_{p \in P} (t, p, c) \;\; where \;\; (t, p, c) = \begin{cases} 1 \; if \; t \in p \wedge norm_p = norm_c \\ 0 \;\; otherwise \end{cases} \quad (3.3)$$

where *P* is the set of all playlists in the dataset, *c* is the target playlist, and *norm* denotes a normalized title.

The preprocessing technique applied on playlist titles has a direct impact on NTP. In some cold-start APC approaches (Faggioli et al., 2018; Ferraro et al., 2018; Kaya & Bridge, 2018), the naïve title normalization has been used successfully. Therefore, NTP also employs the same technique for convenience. As previously presented in Table 2.3, the naïve title normalization results in a total number of 17,381 unique normalized titles when applied on the entire MPD.

3.2. Testing at a Large Scale

For a comprehensive analysis to understand user dynamics in naming music sessions, it is required to perform extensive experiments on a large scale. In terms of data size and diversity, the MPD can be as considered as a suitable music dataset to meet this requirement. On the other hand, scalability might be a possible concern to process large datasets (Magnusson et al., 2020) like the MPD.

All playlists in the MPD have a title and at least one track. Therefore, it is possible to transform any playlist into a cold-start instance. Since the dataset includes one million playlists, each of the naïve recommendation models introduced in Chapter 3.1 can be tested with a total number of one million cold-start instances. This dissertation employs leave-one-out cross-validation (LOOCV), which is a method for the generalization performance of a model (Vehtari & Ojanen, 2012), to perform tests on such a large scale. As illustrated in Figure 3.2, a naïve recommendation model is trained using all the observations in the MPD. At the testing phase, a hold-out observation invokes a recommendation request that causes replication of the model. Simultaneously, the replica is modified to forget the learned representations of the test instance. Final recommendations are produced via the replica model. Then, the replica is disposed, which leaves the system in a stand-by state until a new recommendation request is invoked.

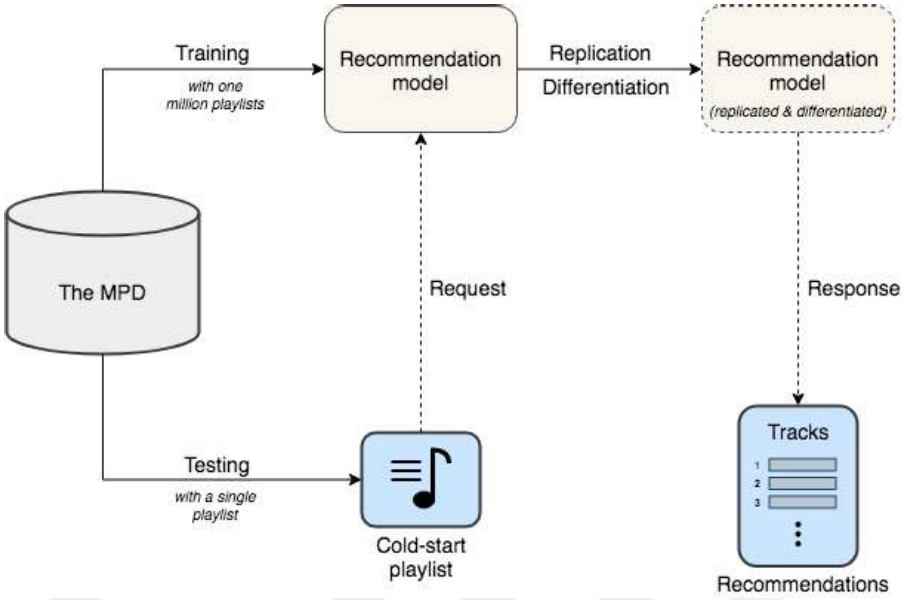

Figure 3.2. *LOOCV method to perform tests on a large scale*

Replication and differentiation of a naïve recommendation model is much faster than training the model from scratch. The process can be repeated for all observations in the MPD, which makes extensive experiments on the entire dataset possible.

3.3. Experimental Evaluation

Employing LOOCV testing methodology with each naïve recommendation model (i.e., GTF, TP, and NTP), a set of extensive experiments are conducted on the MPD. The number of recommendations to be produced for each test instance is set to 500 as in the RecSys Challenge 2018 (Chen et al., 2018; Zamani et al., 2019). During the experiments, a total of 1.5 billion (3 models * 1,000,000 test instances * 500 recommendations) cold-start recommendations are produced.

The main objectives of the performed experiments and corresponding analyses are to enlighten the following research questions:

- *Research question 1: To what extent can the music preference in a playlist be predicted from its title?*
- *Research question 2: What types of titles can accurately predict the music preference in a playlist?*
- *Research question 3: What is the relationship between the frequent preference of a title and its significance in terms of recommendation?*

3.3.1. On the effectiveness of titles as an auxiliary data source

Accuracy assessment of the naïve recommendation models, which make use of user-generated information at different levels, is a good starting point to investigate the usability of playlist titles as an auxiliary data source to mitigate the cold-start problem in APC. Table 3.1 presents the overall recommendation performances of the employed models on one million test instances. According to the attained scores, NTP performs the best on each evaluation metric (0.1012 as R-precision, 0.2121 as NDCG, and 0.2936 as Recall). TP also produces comparable recommendations close to the best results. On the other hand, GTF achieves incomparably low accuracy (0.0274 as R-precision, 0.0819 as NDCG, and 0.1354 as Recall) with the other two models. Clearly, incorporating user-generated titles into cold-start recommendation process has a positive impact on the overall accuracy.

Table 3.1. *Overall accuracy of the naïve recommendation models on one million test instances*

Model	R-precision@500	NDCG@500	Recall@500
GTF	0.0274	0.0819	0.1354
TP	0.0902	0.1852	0.252
NTP	**0.1012**	**0.2121**	**0.2936**

Although NTP appears to be the best performing approach among the three naïve recommendation models, there might be some instances where GTF or TP provides more accurate recommendations. In order to examine such cases, each of the one million test playlists is evaluated on its own. In Figure 3.3, where the blue and yellow bars show the superiority of models over each other, and the red bars represent ties, pairwise superiority comparisons of the naïve recommendation models are presented.

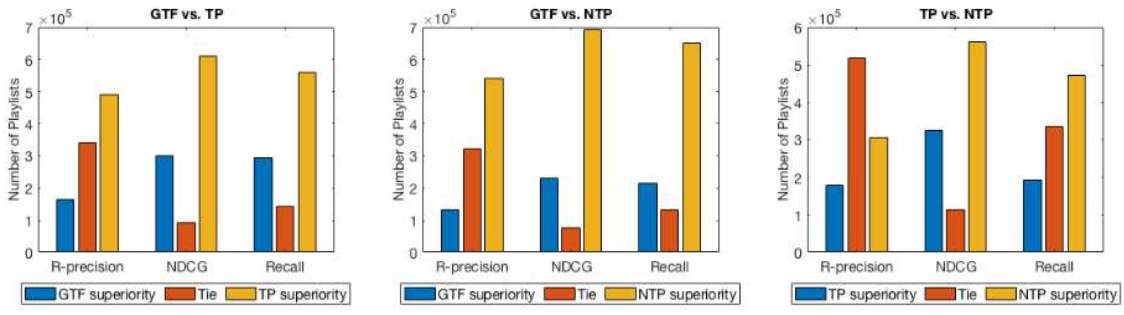

Figure 3.3. *Pairwise superiority comparisons of the naïve recommendation models*

According to Figure 3.3, the percentage of test playlists in which GTF and TP outperforms NTP in terms of NDCG (23.1% and 32.3%, respectively) is substantial. Similar observations also apply to R-precision and Recall metrics. Therefore, selective strategies behaving differently under different circumstances should be developed to improve the overall recommendation accuracy of cold-start APC approaches (Yürekli et al., 2021)

Another aspect of cold-start recommendation performance is the ability of a model to retrieve adequate number of items from the music catalogue as recommended tracks. As presented in Table 3.2, GTF is an independent model that can produce any number of recommendations for any playlist. On the other hand, such particularity does not apply to TP and NTP models. The direct title matching approach in TP generates an average of 439.04 recommendations, which can satisfy the required number of recommended tracks for only 80.61% of the test instances. In addition, no recommendations can be produced for 4.34% of the playlists. Normalizing titles in NTP helps to improve title coverage. Thus, the average number of recommendations increases to 492.53, the percentage of playlists with sufficient recommendations rises to 96.56%, and the percentage of playlists without any recommendations drops to 0.1%.

Table 3.2. *Model abilities to produce adequate number of recommendations*

Model	Number of average recommendations	Test instances satisfying 500 recommendations (%)	Test instances not provided with any recommendations (%)
GTF	**500**	**100**	**0**
TP	439.04	80.61	4.34
NTP	492.53	96.56	0.1

Since users can prefer any meaningful or meaningless expressions as title terms, title-based recommendation approaches have the vulnerability of not producing adequate or any recommendations for cold-start playlists with rare titles. For such cases, robust techniques such as GTF should be preferred as complementary approaches (Kaya & Bridge, 2018).

An additional criterion to compare the naïve recommendation models is a model's capability to extend a cold-start playlist with 100% accuracy. If recommendations to a playlist provide a score of 1.0 in terms of R-precision, NDCG, and Recall, this implies

that the playlist is continued with 100% recommendation accuracy. The expected ability is obviously difficult to meet, especially in a cold-start scenario. However, it is still worth investigating. As presented in Table 3.3, while NTP extends 1717 playlists with 100% accuracy, this number decreases slightly to 1612 in TP. On the other hand, GTF cannot meet the requirement for any playlists; it does not have such a capability. Eventually, title aggregation in NTP positively affects title-based cold-start APC models in terms of the ability to provide recommendations with 100% accuracy.

Table 3.3. *Model abilities to provide recommendations with 100% accuracy*

Model	Number of playlists with 100% recommendation accuracy
GTF	0
TP	1613
NTP	1717

In conclusion, the first research question given in Chapter 3.3 can be addressed as follows. In the absence of significant music preference indicators such as seed tracks (Kelen et al., 2018; Logan, 2002), the limited information in playlist titles is better than nothing; it is possible to build intent recommendation models based on an auxiliary dataset of playlist titles. In addition, as titles are utilized more effectively, the overall recommendation accuracy slightly increases. On the other hand, title-based models should be supported by alternative recommendation strategies capable of dealing rare and exceptional cases where titles do not provide any useable information.

3.3.2. Hidden content in playlist titles: an accuracy perspective

Preceding experimental results show that NTP is the model achieving highest recommendation accuracy among the employed models. In a sense, the approach applied in NTP can also be considered as clustering playlists around certain terms. All tracks belonging to playlists that match a normalized title actually form a cluster by converging around that title. In this perspective, the information contained in the titles of clusters with high recommendation accuracy becomes interesting.

This section focuses on revealing the hidden content within clusters that provide considerably high accuracy from the 17,381 clusters generated by the NTP model. After all clusters are ranked according to their average NDCG scores, the top-1000 accurate

clusters are chosen as the region of interest for further analyses. Table 3.4 presents playlist and track coverage statistics of the chosen clusters. As the table suggests, informative clusters are rare; the top-1000 accurate clusters can only cover about 3.5% of the whole dataset. In addition, track coverage is also low, only 9% of the unique tracks are included in informative clusters. Therefore, it is essential to analyze and understand what kind of information these clusters contain in their titles.

Table 3.4. *Playlist and track coverage statistics of the top-1000 accurate clusters*

Property	**Value**	
Number of playlists covered	34,229	
Number of unique tracks covered	201,987	
	Minimum	**Average**
R-precision@500	0.172	0.544
NDCG@500	0.563	0.714
Recall@500	0.452	0.762

When the titles of the top-1000 accurate clusters are investigated in terms of their content, it is seen that these titles point out some specific themes with a common music sense. Thereby, recommendation algorithms can produce accurate recommendations by utilizing such valuable information. The themes recognized in the top-1000 accurate clusters can be listed as follows.

i. *Artists:* The title of an artist-themed playlist may directly include the name of the corresponding artist (e.g., “ariana grande”, “drake”, and “eminem”). In such a case, it is a good option to recommend music by this artist. Furthermore, recommendations can also be made from artists who have similar music style with the target artist.
ii. *Soundtracks:* A title involving the name of a movie or a TV show (e.g., “la la land”, “sons of anarchy”, and “mamma mia!”) is a strong signal for recommender systems to continue the corresponding playlist with music belonging to the soundtracks of that production.
iii. *Albums:* Similar to an artist-themed playlist, a playlist may include tracks from a specific album of an artist. When a title contains such a signal (e.g., “18 months” and “eminem – curtain call”), the target playlist should be continued with music tracks from that album. It is notable that album-

themed playlists might contain both artist name and album name in their titles.

iv. *Compilations:* Some albums and playlists published by music organizations (e.g., “punk goes pop” and “100 hits of the 80’s”) are shared music listening sessions among users. Recognizing a compilation through its name in title enables MRS to continue these playlists accurately.

v. *Record labels:* Some titles contain names of record labels (e.g., “motown”, “ovoxo”, and “two steps from hell”) that publish music recordings. When such a title is recognized, it is appropriate to recommend music correlated to the corresponding record label.

vi. *Genres:* A music genre is a precise way to narrow down possible options to continue music playlists. “latin trap”, “bachata”, and “country” are some terms belonging to the titles of genre-themed playlists. When used with an additional information such as a time-period (e.g., “90’s alternative music“ and “2000’s r&b”), genres becomes much more feasible in terms of MRS.

vii. *Musicals:* In a musical-themed playlist, the name of the musical (e.g., “dear evan hansen” and “hamilton”) is likely to appear in the title text. Capturing such an information paves the way to continue these playlists in a coherent way.

viii. *Festivals:* When a recommender system encounters the name of a music festival in a playlist title, it should recommend music from that festival to the target playlist. It is notable that music festivals are often named with the year in which they are organized (e.g., “bonnaroo 2017” and “outside lands 2015”).

ix. *Video games:* In today’s entertainment world, video games aim to impress gamers not only with their scenario, game play, and graphics, but also with their music. Similar to the soundtracks, the name of a video game (e.g., “the last of us” and “life is strange”) indicates that the target playlist should be continued with music from that game.

x. *Time references:* Some titles include a time-reference that usually refers to a decade or a year. When combined with another theme such as a genre (e.g., “90’s favorite rock”, “reggaeton 2017”, and “country throwbacks”), time references become strong indicators of music preference.

The list presented above briefly explains the themes discovered in the titles of the top-1000 accurate clusters. Although some titles such as “grammy style” and “warriors” cannot be affiliated with any meaningful topic, almost all clusters are associated with one of the ten themes given above. In Figure 3.4, the distribution of these themes is presented to emphasize which themes are more common than the others.

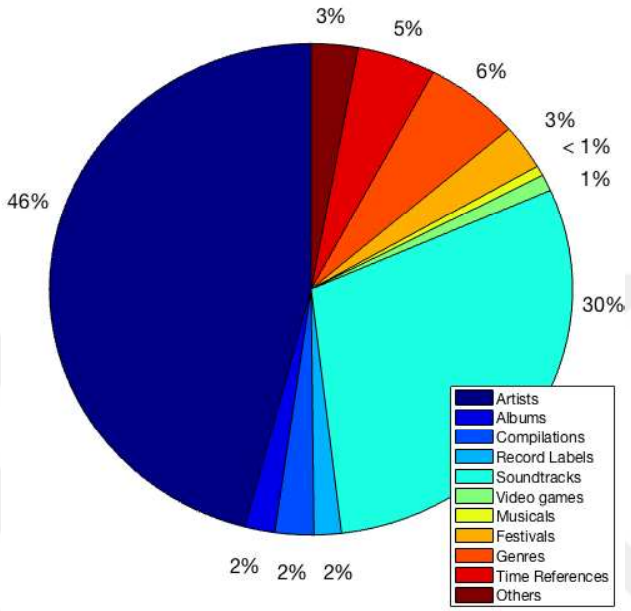

Figure 3.4. *The distribution of themes identified in the titles of the top-1000 accurate clusters*

As illustrated in Figure 3.4, 46% of the informative clusters fall into the category of artist-themed playlists. Soundtracks follow this category with a rate of 30%. These two common categories are followed by genres and time references with rates of 6% and 5%, respectively. For the rest of the clusters, the thematic distributions are highly close to each other. Therefore, it can be concluded that among the themes that provide high cold-start recommendation accuracy, the artist and soundtrack categories stand out due to their frequent preference by users.

Figure 3.5 presents an alternative visualization for the identified themes. Some similar themes in this figure are merged under a conjoint class for a better illustration. The figure also shows prominent items in the themes (with larger font sizes). For examples, “drake”, “eminem", and “kanye west” are common titles belonging to the artist theme. Similarly, “disney” and “guardians of the galaxy” are frequent entities among soundtracks.

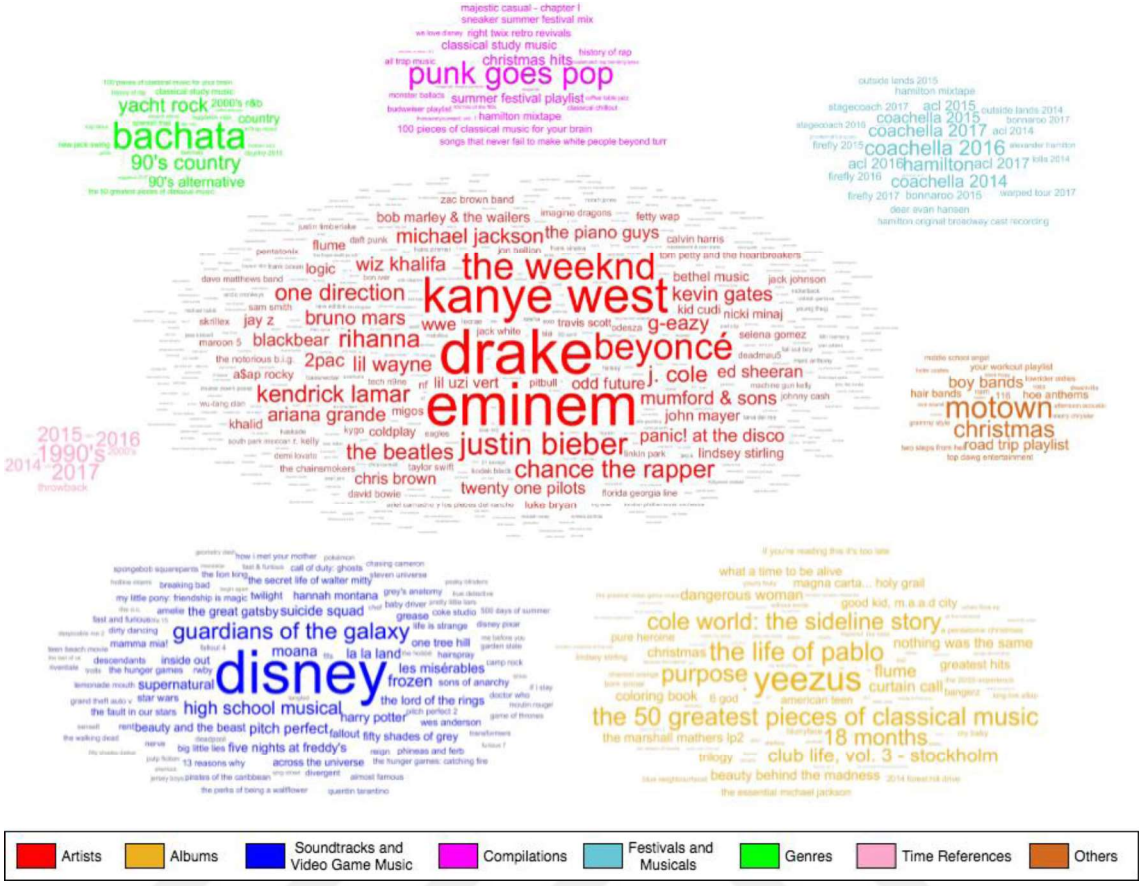

Figure 3.5. *An alternative visualization of themes identified in the top-1000 accurate clusters*

An important observation about artist-themed playlists is the fact that users often prefer to name this kind of playlists with artist aliases. An artist alias can be a stage name, a nickname, or an abbreviation. For example, “kung fu kenny”, “king kenny”, “k dot”, and “kendrick” are some terms pointing out Kendrick Lamar[6], who is an American rapper. Similarly, in another example, Drake[7], a Canadian musician, is commonly mentioned with terms such as “champagne papi”, “drizzy”, “drizzy drake”, and “draking”.

The importance of artist aliases in terms of music recommendation is a subject to be clarified. Figure 3.6 presents samples of alias usage for three different artists. For all the three cases, the NDCG scores of clusters with artist names and the clusters with artist aliases are both high and very close to each other. It is clear that an artist alias is as significant as the name of that artist in cold-start music recommendation.

[6] https://en.wikipedia.org/wiki/Kendrick_Lamar
[7] https://en.wikipedia.org/wiki/Drake_(musician)

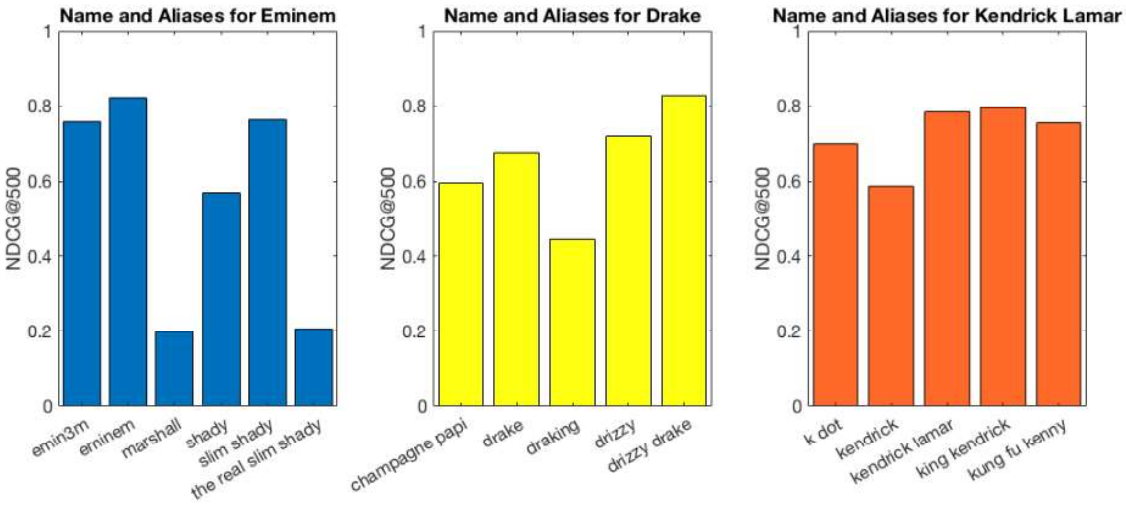

Figure 3.6. *Three cases of artist alias usage in artist-themed playlists*

To sum up, this section clarifies the second research question given in Chapter 3.3 as follows. Accurate cold-start recommendations can be generated when titles point out specific themes carrying a common music sense. The content of an informative title can be of many different forms, such as artists, soundtracks, albums, genres, and time references. Among these, artists and soundtracks are the prominent categories in terms of frequent preference.

3.3.3. Hidden content in playlist titles: a preference perspective

A criterion to consider when discovering the hidden information in user-generated content is the common preference of titles by users. Despite its high preferability, the significance and usability of a frequent title in a recommendation algorithm might be still questionable (Pichl et al., 2015).

In order to identify the most common playlist titles, the coverage of playlist clusters is examined. Based on the assumption that a cluster including at least 1000 playlists is dense, a comparison of dense and non-dense clusters is given in Table 3.5. Interestingly, only 113 of 17,381 clusters (i.e., the number of unique normalized titles in the MPD) are dense. Yet, this seemingly small proportion covers actually 28.65% of the entire MPD. The remaining 17,268 clusters cover the rest of the dataset.

Table 3.5. *A comparison of playlist clusters in terms of coverage*

Level	Total clusters	MPD coverage (%)
Clusters having at least 1000 playlists	113	~ 28.65
Clusters having less than 1000 playlists	17,268	~ 71.35

In terms of thematic information contained in the titles, 113 dense clusters fall into five major categories as genres (e.g., “rock”, “jazz”, “salsa”), time references (e.g., “90’s”, “halloween”, and “throwbacks”), activities (e.g., “driving”, “road trip”, “work”), mood and emotions (e.g., “chill vibes”, “happy”, “sad”), and generic terms. A complete list of the most common clusters and their categories is presented in Table 3.6.

Table 3.6. *The most common titles and their thematic categories*

Category	**Normalized titles of clusters**			
Genres	• acoustic	• alt	• alternative	• christian
	• classic	• classic rock	• classical	• classics
	• country	• dance	• disney	• dubstep
	• edm	• gospel	• hip hop	• indie
	• jazz	• love songs	• metal	• musicals
	• party	• pop	• r&b	• rap
	• reggae	• reggaeton	• rock	• salsa
	• slow	• trap	• worship	
Time references	• 2015	• 2016	• 2017	• 80’s
	• 80s	• 90’s	• 90s	• christmas
	• fall	• halloween	• new	• old
	• old school	• oldies	• oldies but goodies	• summer
	• summer 17	• summer 2015	• summer 2017	• summer 2017
	• tbt	• throwback	• throwbacks	
Activities	• beach	• car	• car jams	• drive
	• driving	• gym	• pregame	• road trip
	• roadtrip	• run	• running	• shower
	• sleep	• study	• wedding	• work
	• work out	• workout		
Mood and emotions	• <3	• calm	• chill	• chill vibes
	• chillin	• feel good	• feels	• fire
	• fun	• happy	• hype	• lit
	• litty	• love	• mellow	• mood
	• pump up	• relax	• sad	• turn up
Generic terms	• bangers	• beats	• good	• good music
	• good songs	• good stuff	• good vibes	• idk
	• jam	• jams	• jamz	• mix
	• music	• my music	• my songs	• playlist
	• random	• stuff	• vibes	

In Figure 3.7, recommendation performances of the most common titles in terms of NDCG are presented. Titles pointing out a music genre or a certain time period have reasonable cold-start recommendation accuracy. Interestingly, terms with past references (e.g., “throwbacks”, “oldies”, and “90’s”) are more effective indicators than terms with recent content (e.g., “summer 2017”, “new”, and “2016”). Furthermore, activities, user

mood, and emotions are less informative about the intended purpose of playlists, which makes them limited in cold-start APC solutions. In addition, generic terms like “good vibes”, “my music”, “jams”, and “random” do not contain any useful information in terms of MRS.

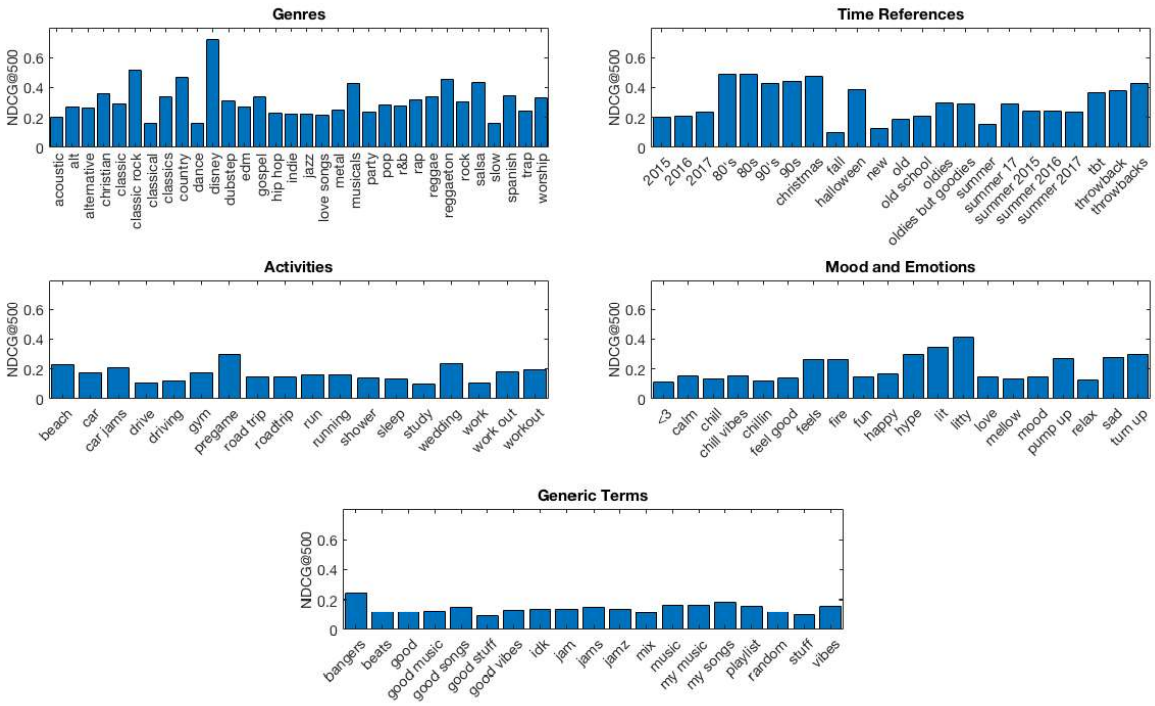

Figure 3.7. *Recommendation accuracy of the most common clusters in terms of NDCG*

In response to the third research question in Chapter 3.3, it can be said that the correlation between common preference of a title and its significance in recommendation is quite weak. Expressions that do not indicate a common music sense may not be useful in MRS, even if they are widely preferred as playlist titles by music listeners.

3.4. Key Factors for Title-Aware Music Recommendation

A title-aware recommendation approach is essential to effectively utilize user-generated titles as an auxiliary data source to alleviate the cold-start problem in APC. Based on the performed analyses and corresponding findings, key factors to develop such title-aware models can be defined as follows:

i. Music listeners commonly express same or similar terms using different spellings. Even the naïve normalization technique, a simple approach involving some elementary preprocessing steps, can obtain 17,381 unique normalized titles among a total number of one million different playlists. A better normalization might result in better recommendation performance. Therefore, elegant methods employing natural language processing (NLP) principles should be developed to homogenize user-generated content. At this point, it should be noted that preprocessing is prone to information loss. Some special characters carrying potential signals such as emojis might be lost during normalization.

ii. Capturing the hidden information in a title may broaden horizons in music recommendation. For example, a system capable of detecting an artist directly from a title (e.g., recognizing Michael Jackson[8] from a new playlist having "king of pop" as its title) can provide recommended tracks by this artist. Therefore, a recognition procedure, such as named entity recognition (NER) should be considered as an alternative feature of MRS.

iii. In order to go beyond syntactic similarities of titles, it is required to measure inter-cluster similarities of playlist clusters. For example, one can easily see that "spanish rock songs" and "canciones de rock en español" are two strings actually pointing out Spanish rock music. However, a computer system without language translation cannot understand this resemblance. Another example can be the contextual similarity between "gym" and "workout". Both of these titles probably refer to the music listened to while exercising. In a nutshell, syntactic similarity alone is not enough; an effective APC should also consider semantic similarity of titles.

[8] https://en.wikipedia.org/wiki/Michael_Jackson

iv. In some rare cases where the title of a new playlist cannot be associated with any existing data, title-based approaches are unfortunately not applicable. A robust system should also consider such cases. Thus, a popularity-based approach such as GTF can be employed as a fallback strategy to keep recommendation process going.

v. It is experimentally shown that some titles (e.g., “my music”, “jams”, and “good stuff”) do not carry any valuable information although they are frequently preferred by music listener. Therefore, there is no accuracy gain in these situations. With prior knowledge of such titles, selective strategies can be employed to improve the overall recommendation accuracy. CBF approaches that utilize user demographics to reflect cultural (Hong et al., 2019) or geographic (Schedl & Schnitzer, 2013) preferences might be a part of this selective strategy.

Based on the key factors described above, an ideal title-aware recommender system for new playlists suffering from the cold-start problem can be hypothetically depicted as given in Figure 3.8. The primary motivation is to maximize the inference attained from playlist titles and tolerate cases where these titles are not useful.

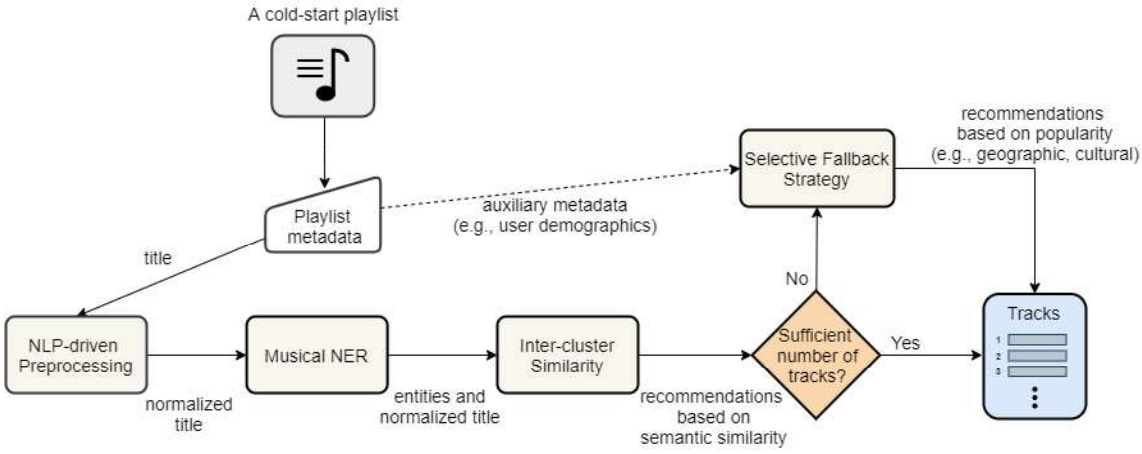

Figure 3.8. *A hypothetical title-aware recommender system for playlists suffering from the cold-start problem*

4. A MULTI-STAGE RECOMMENDATION APPROACH BASED ON LATENT SEMANTIC INDEXING FOR COLD-START PLAYLISTS

The analyses and findings presented in Chapter 3 can be regarded as a proof of concept for the effectiveness of user-generated titles to alleviate the cold-start problem in APC. In the absence of predictive data, playlist titles can be utilized as an auxiliary source of information to build recommendation models (Yürekli et al., 2021). It is also evident that elegant approaches deriving meaning from raw text are likely to produce better recommendations. For example, a model that can infer language and genre from title texts might successfully recommend Spanish rock music for new playlists with titles such as "my favorite spanish rock songs" and "canciones de rock en español".

In order to come up with an MRS having such capabilities, this dissertation approaches to the problem in a novel retrieval perspective. When playlists and tracks are considered as documents and terms, respectively, the cold-start problem in APC becomes an IR task that requires effective retrieval of similar documents for a given user query, which is actually the title of a new playlist. Eventually, the significant terms (i.e., tracks that are more likely to appear in playlists) in the retrieved documents are going to form a recommendation list for the target playlist. Based on the given idea, this dissertation proposes a multi-stage recommendation approach using LSI to discover hidden relations between titles and tracks. As illustrated in Figure 4.1, the proposed approach consists of four main stages, which are playlist clustering, LSI modeling, neighboring clusters retrieval, and track weighting.

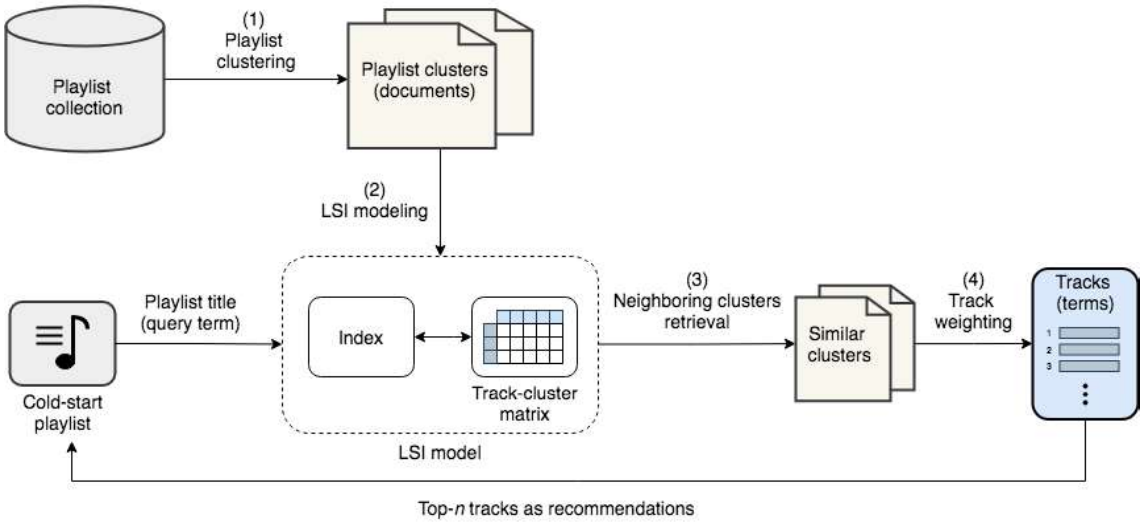

Figure 4.1. *An architectural overview of the proposed recommendation approach*

Playlist clustering and LSI modeling (i.e., the first two stages in Figure 4.1) form the basis of representation learning (Liu et al., 2011), which aims to construct latent space representations of music items. On the other hand, neighboring clusters retrieval and track weighting (i.e., the last two stages in Figure 4.1) serve as a deployment environment in which the actual recommendations are produced for new playlists.

In the rest of this chapter, the four stages of the cascaded recommendation architecture are described in more detail. Afterward, the experiments conducted to evaluate recommendation accuracy in cold-start APC are presented. Finally, a comparison with the state-of-the-art approaches is given.

4.1. Playlist Clustering

User-generated expressions like “summer hits”, “throwbacks”, and “rock classics” can be utilized as indicators of common music sense (Pichl et al., 2015), especially in cold-start situations where predictive data is missing. The first stage of the proposed recommendation approach aims to build a knowledge base using the syntactic similarities of playlist titles.

Firstly, playlists in the MPD (only the instances in the training partition that is presented later in Chapter 4.5) are clustered around normalized titles, which are homogenized forms of title texts obtained by the naïve title normalization technique. This way, it is possible to obtain condensed sets of playlists (up to 17,381 different sets). Then, the resulting clusters are transformed into document representations, which are essential structures for further IR tasks. Figure 2.1 depicts how a cluster is represented as a document.

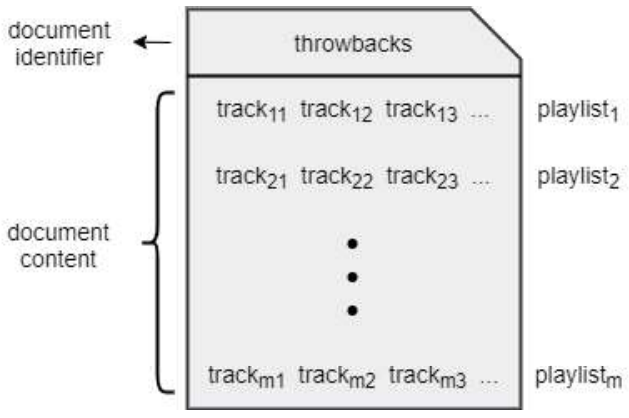

Figure 4.2. *Representing a playlist cluster as a document with an identifier and content*

As illustrated in Figure 2.1, a document consists of two main fields, which are the document identifier and the document content. While normalized title of a cluster is used as the document identifier, concatenation of tracks in playlists forms the actual document content. Naturally, as the number of playlists and tracks contained in a cluster increases, the corresponding document content is expected to have long sequences with repetitive terms.

4.2. LSI Modeling

In the previous stage, a collection of playlists is transformed into a set of documents by clustering the playlists using their normalized titles. The next step in the recommendation process is to create a model capable of estimating pairwise similarities of the documents.

The second stage of the proposed system applies LSI, a powerful dimensionality reduction technique to explore structures in data and extract meaningful feature spaces (Cunningham & Ghahramani, 2015), on the document collection to obtain such a model. Typically, LSI builds a semantic vector space by employing singular value decomposition (SVD) on a term-document matrix. SVD decomposes this large matrix into smaller matrices of reduced dimensions, so that it becomes possible to reveal hidden relations between terms. Remarkably, in the terminology of this dissertation, the concept expressed by a track-cluster matrix is actually a term-document matrix.

In order to perform SVD, firstly, a track-cluster matrix A is created. This is a rectangular $m \times n$ matrix where m is the number of unique tracks and n is the number of clusters. An element $a_{i,j}$ in A indicates the frequency of track t_i in cluster c_j. Once all the frequencies (i.e., the elements of A) are computed, A is factorized into the product of three matrices as shown in the equation below.

$$A = U\Sigma V^T \tag{4.1}$$

where U and V represent track and cluster vectors, and $\sum$ denotes the singular values of A. Figure 4.3, which is derived from Berry et al. (1995), illustrates the described SVD process on the track-cluster matrix A. The number of dimensions to perform SVD is denoted by k in the figure.

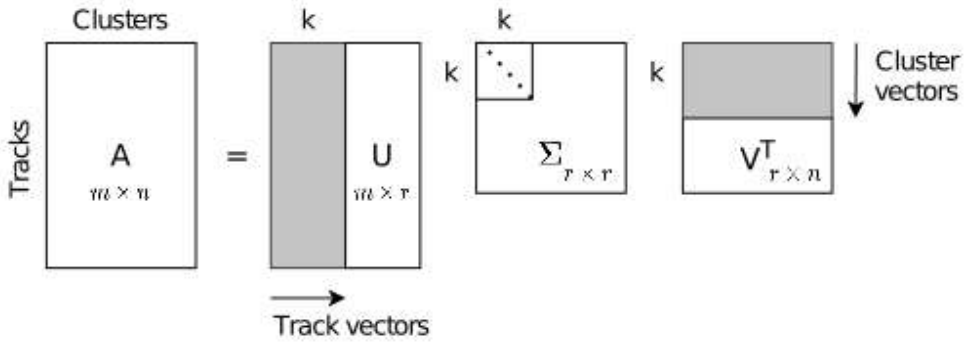

Figure 4.3. *An illustration of the SVD process on a track-cluster matrix*

The decomposition process helps to reveal latent semantic cold-start representations by transforming frequency data into a dense vector space. The resulting structures (i.e., the track-cluster matrix, track and cluster vectors, singular values, and the index itself) form the final LSI model.

Building an LSI model completes the representation learning process that is composed of playlist clustering and LSI modeling stages. The system is now able to assess the similarity of two clusters by calculating the cosine similarity of their corresponding vectors (Kuhn et al., 2005).

4.3. Neighboring Clusters Retrieval

At the deployment stage of the system, a new playlist p suffering from the cold-start problem is accepted as a target playlist to be provided with music recommendations. Initially, the title of p is transformed into a user query by applying the naïve title normalization technique. Then, this query is used to retrieve an initial cluster c_i from the LSI model. The system treats c_i as a seed and computes cosine similarity of the other clusters to the seed cluster.

At the current state, the system is aware of a recommendation request through c_i. The next step is to pick up promising documents (i.e., similar clusters to c_i) that will form the track search space. For this purpose, the following neighborhood selection techniques are introduced.

i. **K-Nearest Neighbors (KNN)** simply classifies the k closest documents as the neighboring clusters of the seed cluster. After ranking all the clusters according to their similarity to c_i, top-k documents are directly retrieved.

ii. **Thresholding Nearest Neighbors (TNN)** determines neighboring clusters of c_i by employing a threshold value t (i.e., a cut-off boundary). In short, documents with larger similarity scores than t are chosen as the neighboring clusters.

iii. **Adaptive Thresholding Nearest Neighbors (ATNN)** is a variant of TNN that adapts the given threshold t dynamically. ATNN discovers a region of interest by computing the mean of similarity scores above the initial cut-off and weighting the mean by t.

The neighborhood selection techniques are designed to perform different behaviors when narrowing down the track search space for final recommendations. Each technique is controlled by an internal hyperparameter (k and t values) that has a direct impact on the recommendation process. The empirical determination of these values is discussed later in Chapter 4.5.

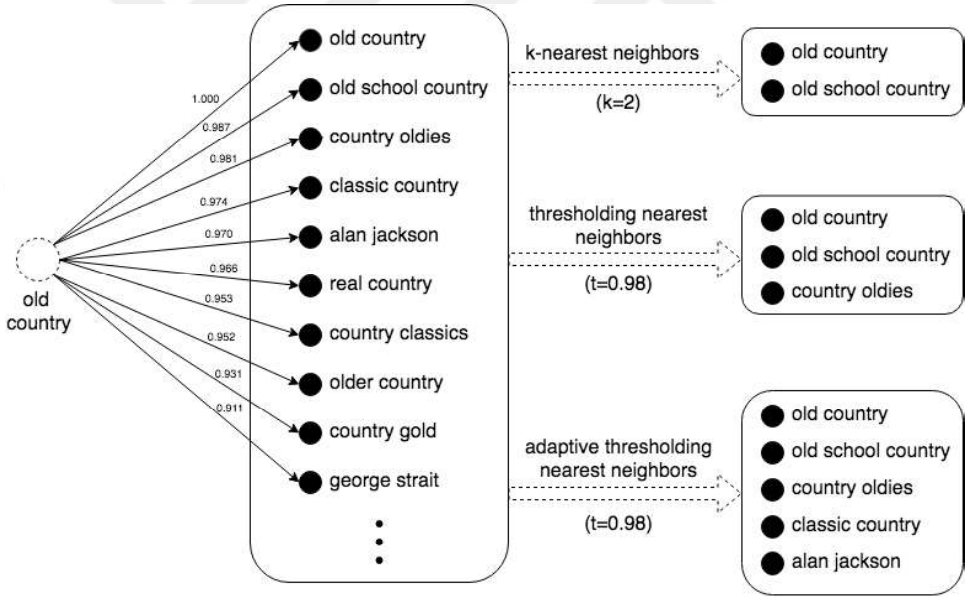

Figure 4.4. *An illustration of neighboring clusters retrieval for a playlist having "old country" as its title*

Figure 4.4 presents a sample case for the neighboring clusters retrieval task to illustrate the characteristics of the employed neighborhood selection techniques better. For a given cold-start playlist having "old country" as its normalized title, the system first retrieves an initial cluster c_i from the LSI model using "old country" as the document identifier. Then, pairwise cluster similarities to c_i are computed and documents are ranked according to these similarity scores. As it is shown in the left side of the figure, the identifiers of the first ten similar documents are some terms (e.g., "old school country",

"real country", and "alan jackson") that are related to old country music. Among these documents, KNN (with k=2) directly chooses the two most similar clusters. Similarly, TNN (with t=0.98) selects three clusters having larger similarity score than t. On the other hand, ATNN (with t=0.98) computes a dynamic threshold from the given t and chooses five clusters as the final retrieved clusters. It is notable that although TNN and ATNN use the same threshold value, they result in different neighboring clusters. Furthermore, c_i always appears in the result sets as the first neighboring cluster.

4.4. Track Weighting

The last step of the recommendation process is to rank tracks in the neighboring playlist clusters and choose an appropriate subset among these items as the final recommendations. This way, the user can be provided with a list of recommended tracks for her new playlist and the primary task of cold-start APC can be accomplished.

At this point, the question of how to rank tracks in the playlist clusters arises. A possible factor that might indicate the significance of a track is the number of occurrences of that track within the clusters. Furthermore, similarity scores of clusters might also be utilized in the ranking process. Inspired from term frequency (Aizawa, 2003), which is a well-known term weighting scheme in the IR domain, this dissertation proposes four track weighting schemes as follows.

i. **Track Frequency (TF)** is the sum of all occurrences of a track in the neighboring playlist clusters. As presented in Eq. (4.2), TF of a track t is calculated by simply summing all occurrences of t in the set of all retrieved playlist clusters C.

$$weight_{TF}(t) = \sum_{c \in C} f_{t,c} \tag{4.2}$$

ii. **Weighted Track Frequency (WTF)** is the scheme that incorporates cluster similarities into the weighting process. When calculating WTF, the frequency of a track is multiplied by the similarity score of the corresponding playlist cluster. WTF is computed as shown in Eq. (4.3).

$$weight_{WTF}(t) = \sum_{c \in C} sim_c * f_{t,c} \tag{4.3}$$

iii. **Maximized Weighted Track Frequency (MWTF)** is a variant of WTF that only employs the maximum cluster similarity score observed for a

track. Although tracks might appear in multiple clusters with different similarity scores, MWTF focuses on the closest cluster to determine the coefficient. MWTF is calculated as given in Eq. (4.4).

$$weight_{MWTF}(t) = argmax_{sim} * \sum_{c \in C} f_{t,c} \quad (4.4)$$

where $argmax_{sim}$ represents the maximum cluster similarity score observed for track t.

iv. **Track Frequency*Inverse Cluster Frequency (TF*ICF)** mimics the behavior of inverse document frequency (Robertson, 2004), which is an effective weighting factor in IR systems. TF*ICF is calculated as given in Eq. (4.5).

$$weight_{TF*ICF}(t) = \sum_{c \in C} f_{t,c} * \log \frac{N}{n_t} \quad (4.5)$$

where N is the total number of clusters and n_t is the number of clusters where track t appears.

All the track weighting schemes described above operate in the same manner. Each unique track is associated with a final weighting score that is calculated by a track weighting scheme (TF, WTF, MWTF, or TF*ICF). Then, the tracks are ranked by their weighting scores. In this way, it becomes possible to create a list of recommended tracks that might continue cold-start playlists.

4.5. Experimental Evaluation

In order to evaluate the effectiveness of the proposed system, a series of extensive experiments are performed on the MPD. This section presents the methodologies followed to realize the cold-start problem in APC, optimize the number of dimensions for LSI, and tune the hyperparameters of the proposed architecture. Afterward, the comparison of the final model with the state-of-the-art approaches is given.

4.5.1. Cold-start scenario realization

The MPD is a complete dataset with warm-start playlists; none of the instances actually suffer the cold-start problem. However, each playlist in the dataset can be easily treated as a new playlist with only title information by simply clearing all the tracks of the playlist. This way, it is possible to realize the cold-start scenario and obtain a test set of any desired size.

Using the strategy described above, the MPD is partitioned into train and test sets as illustrated in Figure 4.5. While 99% of the dataset is allocated for training, the rest is reserved for testing purposes. All samples in the resulting 10,000 test set are randomly selected. The only criterion applied during random sampling is that each track in the test set must also be included in the train set. Otherwise, it would not be possible to recommend such a track regardless of any employed algorithm or approach.

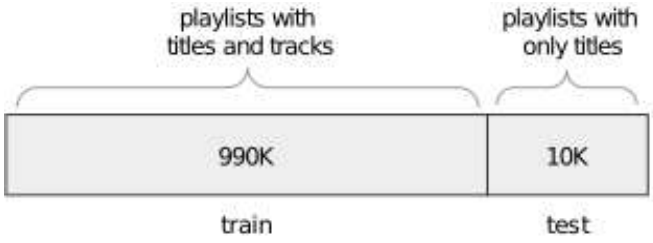

Figure 4.5. *Splitting the MPD into train and test sets*

After splitting the MPD into two separate sets, playlists in the training partition are clustered using the naïve title normalization method introduced in Chapter 2.3. For further IR tasks such as document similarity estimation, these playlist clusters are transformed into document representations. As a result, a training collection of 17,359 documents with more than 66 million terms is obtained. Some basic properties of this collection are presented in Table 4.1.

Table 4.1. *Basic properties of the train set after document transformation*

Property	Value
Number of documents	17,359
Number of terms	66,065,212
Minimum terms in a document	5
Maximum terms in a document	922,262
Average terms per document	3,806

4.5.2. Dimensionality optimization

A vital aspect to apply LSI on a document collection is to determine the optimal number of dimensions spanning that collection (Landauer et al., 2013). This parameter is highly critical for dimensionality reduction; it has a direct impact on the SVD process. Thus, dimensionality optimization is an essential to step to improve the overall recommendation performance of the proposed system.

One way to determine optimal number of dimensions for LSI is to measure and compare topic coherence of different values. Denoting the set of frequent terms in the document collection as *T*, coherence is formulated as shown in Eq. (4.6). In the context of this dissertation, topic coherence corresponds to the semantic similarity between frequent tracks.

$$\text{Coherence} = \sum_{(t_i,t_j)\in T} score(t_i, t_j) \tag{4.6}$$

In Eq. (4.6), coherence is expressed as the sum of similarity scores for a set of frequent tracks. The assessment of similarity can be either intrinsic or extrinsic. This dissertation employs the UMass score (Mimno et al., 2011), which is an intrinsic measure of distributional similarity. For a pair of tracks t_i and t_j, the UMass coherence score is computed as given in Eq. (4.7).

$$score_{UMass}(t_i, t_j) = \log \frac{D(t_i, t_j) + 1}{D(t_i)} \tag{4.7}$$

where $D(t_i)$ is the number of clusters containing track t_i and $D(t_i,t_j)$ is the number of clusters containing track t_i and track t_j together.

In order to find the optimal number of dimensions for the LSI model to be employed in the proposed recommendation architecture, several latent models are trained using the values in the range of 50 to 1000 (in increments of 50). Then, the UMass coherence score of each model is calculated. Figure 4.6 presents these scores along with their dimensions.

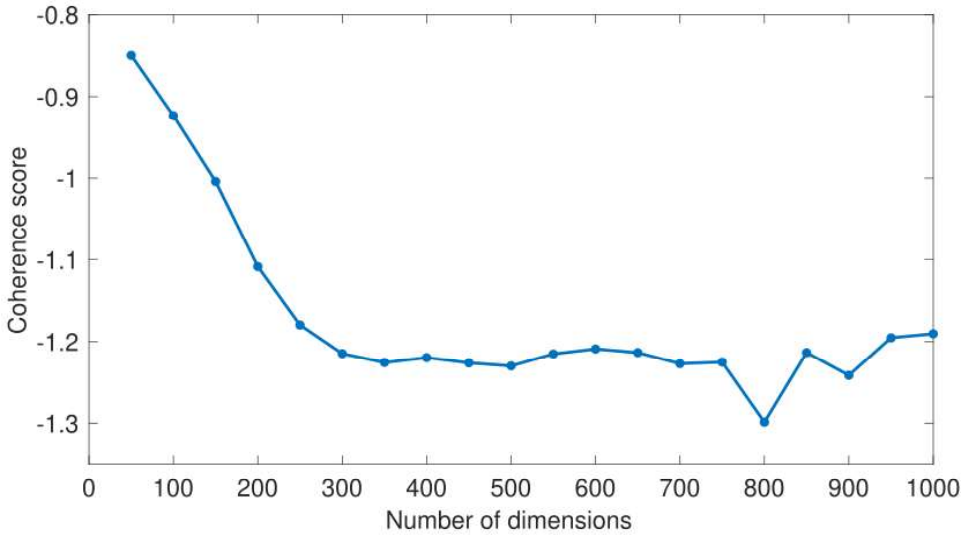

Figure 4.6. *UMass coherence scores of LSI models with varying dimensions*

As illustrated in Figure 4.6, the UMass coherence scores of LSI models with varying dimensions decrease along the x-axis. This trend implies that as the number of dimensions increases, documents in the collection begin to separate better. The coherence has a sharp decrease until the dimensionality reaches 300. However, the minimum coherence is achieved at 800, which appears to be the optimal number of dimensions for the collection of playlist clusters. Therefore, the final LSI model to be employed in the proposed system uses 800 as the number of dimensions.

4.5.3. Hyperparameter tuning

In the proposed multi-stage recommendation architecture, the data flow between the stages causes each stage to affect the succeeding one. Therefore, the overall recommendation accuracy of the system depends on cascaded interactions of several factors. Figure 4.7 presents these factors and their hyperparameters that should be determined empirically for a better accuracy.

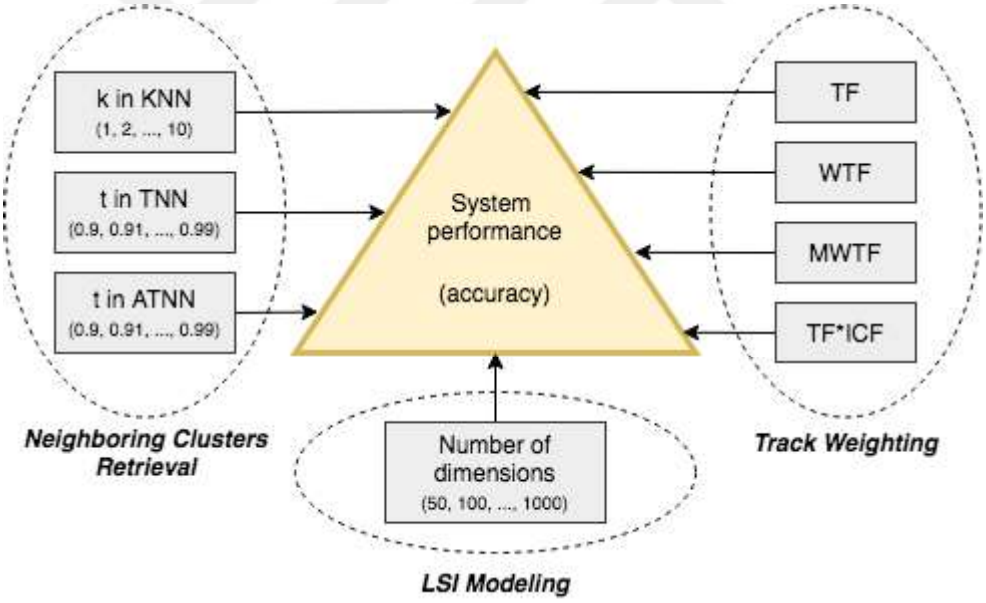

Figure 4.7. *Factors and hyperparameters affecting recommendation accuracy of the proposed system*

First of all, cold-start recommendation performance depends on accurate estimation of inter-cluster similarity, which is directly related to the number of dimensions chosen for dimensionality reduction during LSI modeling. Secondly, the selection of promising clusters at the neighboring clusters retrieval stage also affects the accuracy of recommendations. Finally, the last factor affecting system performance is the track weighting scheme employed to weight and rank tracks when producing a final list of recommendations.

The optimal number of dimensions to perform SVD is previously determined as 800 (i.e., the point of minimum coherence as shown in Figure 4.6). Since the first factor is already resolved, it is required to focus on the next two factors. The three neighborhood selection techniques (KNN, ANN, and ATNN) can be used in combination with the four track weighting schemes (TF, WFT, MWTF, and TF*ICF). In addition, each of the neighborhood selection techniques must be associated with an internal control parameter (k and t values introduced in Chapter 4.3). For KNN, k can be any natural number larger than 0, while for TNN and ATNN, t must be a real number between 0 and 1. This dissertation uses ten empiric values as control parameters (k = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 and t = 0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99). Accordingly, the proposed system can be operated with a total number of 120 (3 neighborhood selection techniques * 4 track weighting schemes * 10 control values) different configurations.

In order to maximize recommendation accuracy by tuning the hyperparameters of the system, a full-factorial experimental design (Antony, 2014) is followed. For each of the possible configurations (120 combinations in total), an experiment is conducted on the same test set of 10,000 cold-start instances. It is notable that the recommendations are limited with 500 tracks per each target playlist.

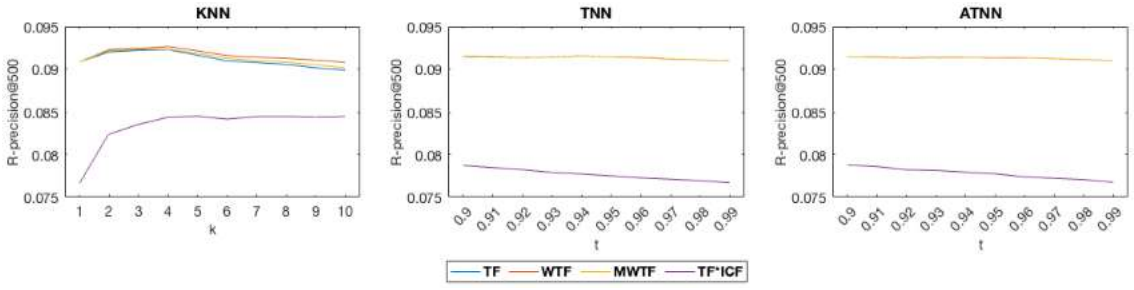

Figure 4.8. *R-precision interaction plots for the factors affecting recommendation accuracy*

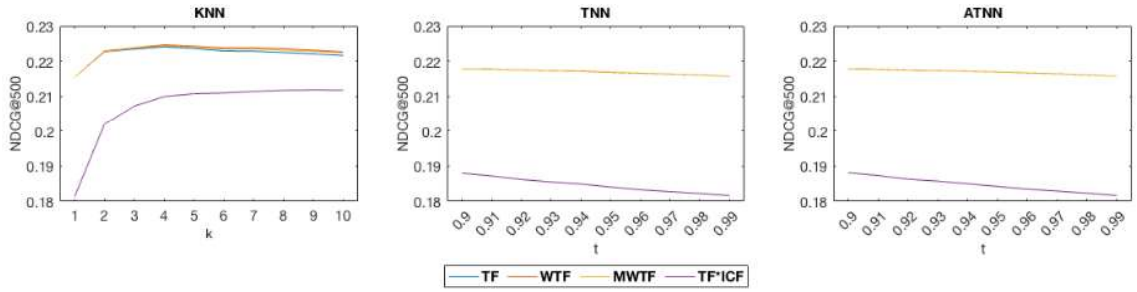

Figure 4.9. *NDCG interaction plots for the factors affecting recommendation accuracy*

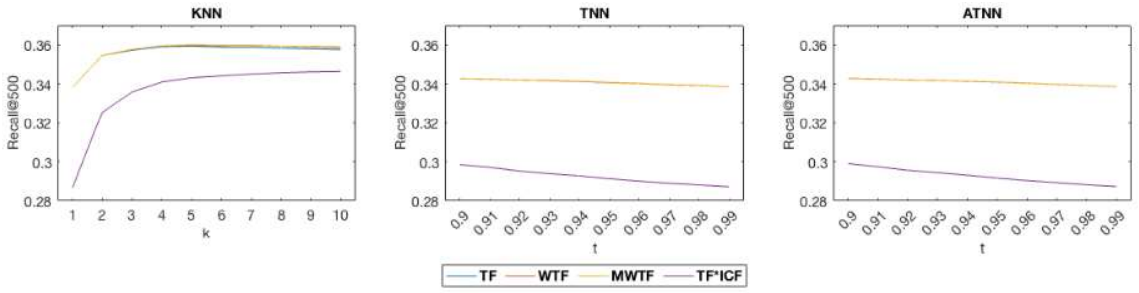

Figure 4.10. *Recall interaction plots for the factors affecting recommendation accuracy*

Figure 4.8, Figure 4.9, and Figure 4.10 present the interactions of neighborhood selection techniques and track weighting schemes in terms of R-precision, NDCG, and Recall, respectively. For all the three evaluation metrics, similar inferences can be made as follows:

i. As a neighborhood selection technique, KNN provides more accurate and discriminative results than TNN and ATNN.
ii. Although TNN and ATNN represent different design choices, they show a similar neighboring tendency on a large scale evaluation.
iii. TF, WTF, and MWTF perform very close to each other. Any of these schemes can be utilized effectively in the ranking process.
iv. TF*ICF performs much worse than other track weighting schemes. It is clear that including ICF in the ranking process does not fit the proposed cold-start APC approach.

In addition to the findings above, Table 4.2 presents the model performs best in each metric. Maximum R-precision and NDCG scores, which are computed as 0.0926 and 0.2246, are acquired by the model combining KNN (k=4) with WTF. On the other hand, maximum Recall is measured as 0.3601 by employing KNN (k=5) with MWTF. Accordingly, building a track search space with four nearest neighbors and weighting track frequencies by cluster similarity are the two ultimate factors for a better accuracy.

Table 4.2. *Best-performing configurations of the proposed approach in all evaluation metrics*

Metric	Max score	Employed neighborhood selection technique	Employed track weighting scheme
R-precision@500	0.0926	KNN (k=4)	WTF
NDCG@500	0.2246	KNN (k=4)	WTF
Recall@500	0.3601	KNN (k=5)	MWTF

4.5.4. Comparison with the state-of-the-art

The experimental work on hyperparameters tuning reveals a final configuration of the proposed system (i.e., KNN (k=4) as the neighborhood selection technique and WTF as the track weighting scheme) achieving the best cold-start recommendation accuracy. The next step is to compare this final model with the state-of-the-art approaches in cold-start APC. For this purpose, the following approaches are adapted to the experimental settings of this dissertation and included in the comparison with the-state-of-the-art:

- CBF with stemming on titles (Faggioli et al., 2018)
- CBF with naïve title normalization (Kaya & Bridge, 2018)
- Similarity search on title metadata (Kelen et al., 2018)
- Weighted hybridization using matrix factorization (Ludewig et al., 2018)

Table 4.3 presents the comparison of the proposed recommendation approach with four state-of-the-art approaches. As shown in the table, the proposed approach outperforms other models in terms of all evaluation metrics. The best configuration of the proposed system (i.e., KNN (k=4) with WTF) achieves 0.0926 as R-precision, 0.2246 as NDCG, and 0.3594 as Recall. These scores are the highest achievements within the corresponding metrics.

Table 4.3. *A comparison of the proposed approach with the state-of-the-art APC techniques*

Approach	R-precision@500	NDCG@500	Recall@500
Faggioli et al. (2018)	0.092	0.2206	0.3479
Kaya & Bridge (2018)	0.0905	0.2125	0.3310
Kelen et al. (2018)	0.0911	0.217	0.3411
Ludewig et al. (2018)	0.0906	0.2229	0.3557
Proposed approach	**0.0926**	**0.2246**	**0.3594**

Table 4.4 presents the results of statistical significance tests performed to show whether the improvements are statistically significant or not. For simplicity, the table only reports the comparisons between the proposed approach and the second-best performing approach for each evaluation metric. According to the p-values (0.0579 and 0.0001) shown in the table, the accuracy improvements are statistically significant with confidence levels of 90% for R-precision, and 99% for NDCG and Recall.

Table 4.4. *Statistical significance of the accuracy improvements provided by the proposed approach*

Comparison	Metric	p-Value	Confidence level (%)
Proposed approach vs. Faggioli et al. (2018)	R-precision@500	0.0579	90
Proposed approach vs. Ludewig et al. (2018)	NDCG@500	0.0001	99
Proposed approach vs. Ludewig et al. (2018)	Recall@500	0.0001	99

The comparisons made so far are directly related to accuracy. Both R-precision, NDCG, and Recall are metrics borrowed from the IR domain to measure the accuracy of recommendations. As previously stated in Chapter 1.2, beyond-accuracy measures are also necessary to evaluate MRS (Kaminskas & Bridge, 2016). Such an aspect can be realized by examining the capabilities of cold-start models on contextual information level. In this direction, the titles of the 10,000 test playlists are classified into ten major groups (artist, context, decade, freshness, genre, holiday, month, season, soundtrack, and year) using the semantic tags provided in Yürekli et al. (2021). However, not every title can be associated with a semantic tag; some user-generated expressions (e.g., “my playlist”, “my favourite music”, and “jams”) do not point out any common sense. Nevertheless, sufficient number of meaningful annotations (3684 test instances) are made for further analysis.

For each playlist associated with a specific theme, the cold-start APC approach achieving the best score is identified. This procedure is repeated for all the three evaluation metrics. Figure 4.12 presents the model capabilities of best NDCG achievements. As illustrated in the figure, the proposed approach performs better than the state-of-the-art approaches in almost all themes. Especially in playlists pointing out a specific context (e.g., “gym”, “studying”, and “work out”), a music genre (e.g., “rock”, “korean pop”, and “hip hop”), or a time-reference (e.g., “throwbacks”, “oldies but goldies”, “summer 2017”, and “new stuff”), the proposed recommendation approach becomes prominent. Similar observations also hold for R-precision and Recall evaluation metrics, which are illustrated in Figure 4.11 and Figure 4.13, respectively.

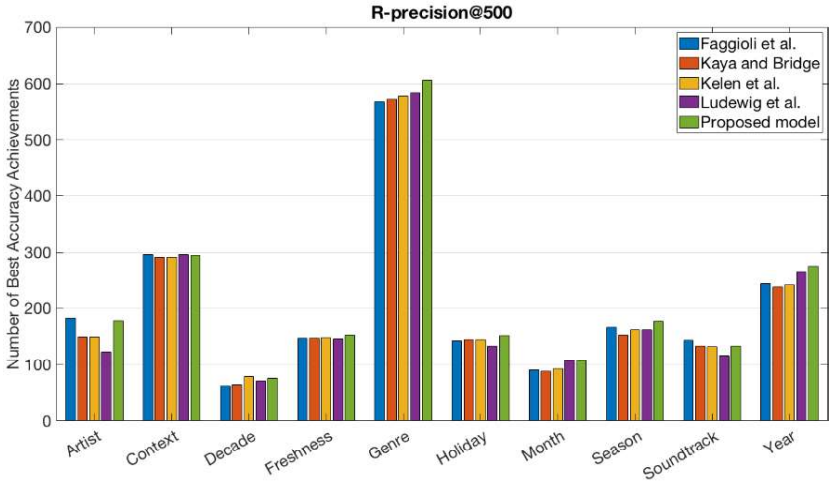

Figure 4.11. *Thematic distribution of best accuracy achievements in terms of R-precision*

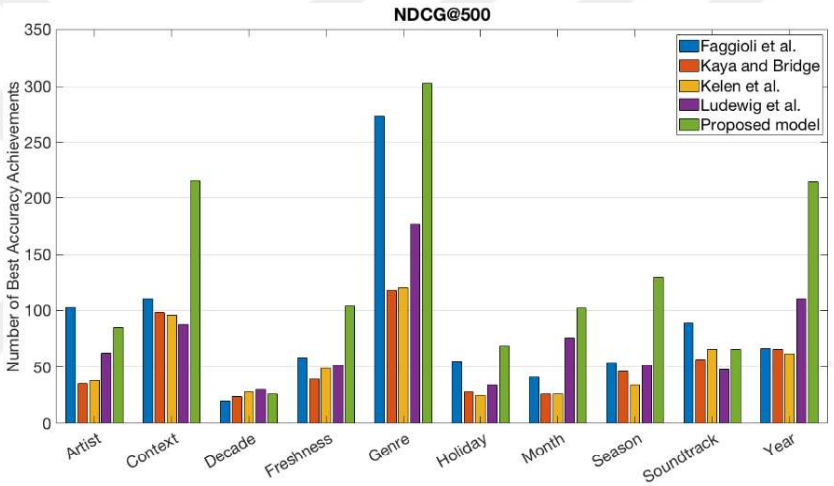

Figure 4.12. *Thematic distribution of best accuracy achievements in terms of NDCG*

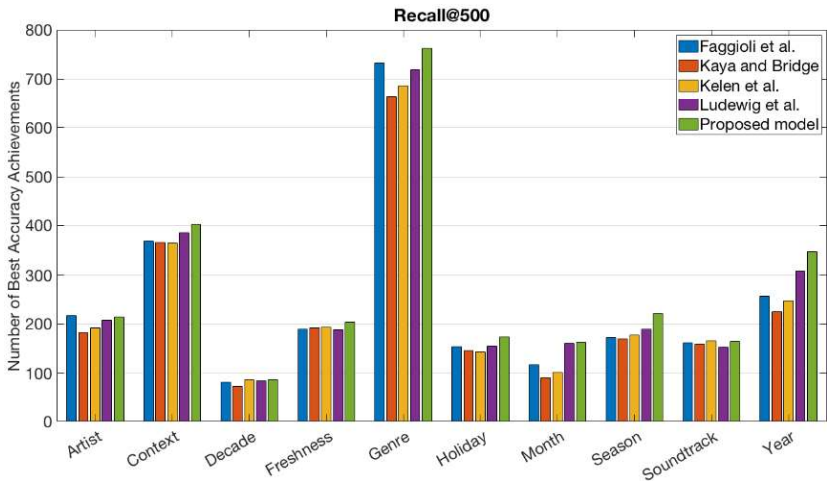

Figure 4.13. *Thematic distribution of best accuracy achievements in terms of Recall*

4.6. Demonstrations of Accurate Cold-Start Recommendations

In order to demonstrate the effectiveness of the proposed recommendation approach, three highly accurate cold-start recommendation cases are given in Table 4.5. In the first example (with playlist id=986478), the normalized title refers to the famous movie called La La Land[9]. In the second example (with playlist id=80875), the normalized title refers to a well-known artist Calvin Harris[10]. Similarly, the third example (with playlist id=84364) includes a signal pointing out Spanish music. In all these three cases, the information available in title texts are quite strong.

Accordingly, the top-5 recommendations substantially match the first-5 tracks of the corresponding playlist. The NDCG scores measured for these samples are 1.0, 1.0, and 0.886, respectively. In a nutshell, the proposed cold-start APC approach is capable of capturing semantic concepts from raw playlist titles. This ability, in a sense, corresponds to the entity recognition mechanism of the ideal title-aware recommender system hypothetically depicted in Figure 3.8.

Table 4.5. *Some examples of highly accurate cold-start recommendations*

Playlist id	First-5 tracks	Top-5 recommendations
986478 "la la land"	1. La La Land Cast – Another Day Of Sun 2. Emma Stone – Someone In The Crowd 3. Justin Hurwitz – Mia & Sebastien's Theme 4. Ryan Gosling – A Lovely Night 5. Justin Hurwitz – Herman's Habit	1. Justin Hurwitz – Mia & Sebastien's Theme 2. Justin Hurwitz – Epilogue 3. Ryan Gosling – A Lovely Night 4. Emma Stone – Someone In The Crowd 5. La La Land Cast – Another Day Of Sun
80875 "calvin harris"	1. Calvin Harris – Green Valley 2. Calvin Harris – Bounce 3. Calvin Harris – Feel So Close 4. Rihanna – We Found Love 5. Example – We'll Be Coming Back	1. Calvin Harris – Feel So Close 2. Calvin Harris – Sweet Nothing 3. Rihanna – We Found Love 4. Calvin Harris – I Need Your Love 5. Calvin Harris – Thinking About You
84364 "spanish"	1. Nicky Jam – El Perdón 2. Enrique Iglesias – Bailando 3. Carlos Vives – La Bicicleta 4. Shakira – Hips Don't Lie 5. Chino & Nacho – Andas En Mi Cabeza	1. Nicky Jam – El Perdón 2. Enrique Iglesias – Bailando 3. Marc Anthony – Vivir Mi Vida 4. Nicky Jam – Hasta el Amanecer 5. Don Omar – Danza Kuduro

[9] https://www.imdb.com/title/tt3783958
[10] https://en.wikipedia.org/wiki/Calvin_Harris

4.7. Conclusions

The cold-start problem makes APC, which is already a complex task in MRS, more challenging. However, with the availability of large datasets, it might be possible to make use of semantically limited (or weak) information such as playlist titles and build effective recommendation models.

In line with the purpose mentioned above, a novel recommendation approach is proposed for new playlists suffering from the cold-start problem. By representing music items as abstractions of documents and terms, cold-start APC is transformed into an IR task. The cascaded architecture consisting of layers, each of which provides data flow to the successor, provides a smooth and effective transition from syntactic similarity of titles to semantic similarity of music preferences. This perspective allows to develop effective clustering, similarity estimation, and ranking methods.

The experimental results performed on a large dataset are highly promising. The proposed latent semantics model, which employs KNN (k=4) and WTF as neighborhood selection technique and track weighting scheme, respectively, outperforms the state-of-the-art approaches in terms of all the employed evaluation metrics. The accuracy improvements are proven to be statistically significant (with confidence levels of 99% for NDCG and Recall, 90% for R-precision). In addition to the accuracy measures, latent space representations are also more effective in discovering certain themes such as artists and soundtracks from title information.

5. CONCLUDING REMARKS

This dissertation focuses on cold-start APC, which is an extreme recommendation case that involves two of the grand challenges of music recommendation. In this final chapter, results attained in the work are summarized, and then some future research directions are highlighted.

5.1. Conclusions

The dissertation proposes a novel, effective, and complete music recommendation approach to alleviate the cold-start problem occurred during automatic continuation of new playlists. The whole work can be evaluated in main two stages: *(i)* exploring the hidden content in playlist titles and *(ii)* building an elegant cold-start recommendation approach.

The first stage of the dissertation can be considered as a preliminary study to understand the dynamics of user-generated titles as an auxiliary data source. Firstly, a testing mechanism is established where each instance of a large music collection containing one million playlists can be analyzed. By using this mechanism, extensive experiments are carried out with naïve recommendation models that make use of title information at different levels. In line with the experimental results, it is shown that titles can be utilized at a certain degree in cold-start APC solutions. In addition, it is revealed that titles referring to a specific theme, such as artists, albums, soundtracks, and genres, provide highly accurate recommendations. Furthermore, the relationship between frequent preference of a title by music listeners and the value of that title in terms of MRS in investigated. Finally, for fellow researchers interested in the subject, the features that should be in an ideal title-aware cold-start APC approach are described.

At the second stage of the dissertation, the cold-start problem is transformed into a retrieval task that regards playlists as documents and tracks as terms, respectively. Based on this analogy, a multi-stage retrieval system is proposed. Hyperparameters are determined to find the optimal configuration of the proposed system, where each stage provides data flow to its successor. A full-factorial experimental design is prepared based on the hyperparameters. Then, the optimal number of LSI dimensions, the best-performing neighborhood selection technique, and the most effective track weighting scheme that maximize overall recommendation accuracy are determined empirically. Afterward, the final architecture and the state-of-the-art approaches are compared on a

test set of 10,000 cold-start instances. It is shown that the proposed system outperforms the state-of-the-art approaches. The accuracy improvements are also shown to be statistically significant.

5.2. Future Research Directions

In this dissertation, the cold-start problem encountered in APC is addressed by utilizing user-generated titles as an auxiliary data source to compensate the lack of predictive data. Although titles are partially successful in understanding playlist characteristics, they are not strong indicators as seed tracks. Alternative data sources such as audio features might be incorporated into music recommendation process to obtain better quality of recommendations. This aspect remains a future work.

Although neighborhood selection techniques and track weighting schemes introduced in Chapter 4 are able to produce better final results than the state-of-the-art approaches, it might be possible to develop more elegant methods to choose and rank candidate tracks. Therefore, the selection of the most probable tracks for a new playlist remains an open issue.

REFERENCES

Aggarwal, C. C. (2016). *Recommender systems* (Cilt 1). Cham: Springer International Publishing.

Aizawa, A. (2003). An information theoretic perspective of tf-idf measures. *Information Processing & Management, 39*(1), 45-65.

Antenucci, S., Boglio, S., Chioso, E., Dervishaj, E., Kang, S., Scarlatti, T., & Dacrema, M. F. (2018). Artist-driven layering and user's behaviour impact on recommendations in a playlist continuation scenario. *Proceeding of the ACM Recommender Systems Challenge 2018*, (s. 1-6).

Antony, J. (2014). *Design of experiments for engineers and scientists.* Elsevier.

Baltrunas, L., Kaminskas, M., Ludwig, B., Moling, O., Ricci, F., Aydin, A., . . . Schwaiger, R. (2011). Incarmusic: Context-aware music recommendations in a car. *International Conference on Electronic Commerce and Web Technologies*, (s. 89-100).

Batmaz, Z., Yürekli, A., Bilge, A., & Kaleli, C. (2019). A review on deep learning for recommender systems: challenges and remedies. *Artificial Intelligence Review, 52*(1), 1-37.

Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM review, 37*(4), 573-595.

Bobadilla, J., Ortega, F., Hernando, A., & Bernal, J. (2012). A collaborative filtering approach to mitigate the new user cold-start problem. *Knowledge-based systems, 26*, 225-238.

Bogdanov, D., & Boyer, H. (2012). Taking advantage of editorial metadata to recommend music. *Proceedings of the 9th International Symposium on Computer Music Modeling and Retrieval*, (s. 618-632).

Bollen, D., Knijnenburg, B. P., Willemsen, M. C., & Graus, M. (2010). Understanding choice overload in recommender systems. *Proceedings of the fourth ACM conference on Recommender systems*, (s. 63-70).

Bonnin, G., & Jannach, D. (2015). Automated generation of music playlists: Survey and experiments. *ACM Computing Surveys (CSUR), 47*(2), 1-35.

Burke, R. (2002). Hybrid recommender systems: survey and experiments. *User modeling and user-adapted interaction, 12*(4), 331-370.

Chen, C.-W., Lamere, P., Schedl, M., & Zamani, H. (2018). Recsys challenge 2018: automatic playlist continuation. *Procedings of the 12th ACM Conference on Recommender Systems*, (s. 527-528).

Chung, C.-H., Chen, Y., & Chen, H. H. (2017). Exploiting playlists for representation of songs and words for text-based music retrieval. *Proceedings of the 18th International Society for Music Information Retrieval Conference*, (s. 478-485).

Clark, E., & Araki, K. (2011). Text normalization in social media: progress, problems and applications for a pre-processing system of casual English. *Procedia-Social and Behavioral Sciences, 27*, 2-11.

Cunningham, J. P., & Ghahramani, Z. (2015). Linear dimensionality reduction: Survey, insights, and generalizations. *The Journal of Machine Learning Research, 16*(1), 2859-2900.

Dror, G., Koenigstein, N., Koren, Y., & Weimer, M. (2012). The Yahoo! Music Dataset and KDD-Cup'11. *Proceedings of the KDD Cup 2011*, (s. 3-18).

Faggioli, G., Polato, M., & Aiolli, F. (2018). Efficient similarity based methods for the playlist continuation task. *Proceedings of the ACM Recommender Systems Challenge 2018*, (s. 1-6).

Ferraro, A., Bogdanov, D., Yoon, J., Kim, K., & Serra, X. (2018). Automatic playlist continuation usign a hybrid recommender system combining features from text and audio. *Proceedings of the ACM Recommender Systems Challenge 2018*, (s. 1-5).

Hong, M., An, S., Akerkar, R., Camacho, D., & Jung, J. J. (2019). Cross-cultural contextualisation for recommender systems. *Journal of Ambient Intelligence and Humanized Computing*, 1-12.

Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS), 20*(4), 422-446.

Kallumadi, S., Mitra, B., & Iofciu, T. (2018). A line in the sand: recommendation or ad-hoc retrieval? *Proceedings of the ACM Recommender Systems Challenge 2018*, (s. 1-6).

Kaminskas, M., & Bridge, D. (2016). Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS), 7*(1), 1-42.

Kaya, M., & Bridge, D. (2018). Automatic playlist continuation using subprofile-aware diversification. *Proceedings of the ACM Recommender Systems Challenge 2018*, (s. 1-6).

Kelen, D. M., Berecz, D., Béres, F., & Benczúr, A. A. (2018). Efficient K-NN for playlist continuation. *Proceedings of the ACM Recommender Systems Challenge 2018*, (s. 1-4).

Kim, J., Won, M., Liem, C. C., & Hanjalic, A. (2018). Towards seed-free music playlist generation: Enhancing collaborative filtering with playlist title information. *Proceedings of the ACM Recommender Systems Challenge 2018*, (s. 1-6).

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer, 42*(8), 30-37.

Kuhn, A., Ducasse, S., & Girba, T. (2005). Enriching reverse engineering with semantic clustering. *12th Working Conference on Reverse Engineering (WCRE'05).*

Landauer, T. K., McNamara, D. S., Dennis, S., & Kintsch, W. (2013). *Handbook of latent semantic analysis.* Psychology Press.

Liu, N. N., Meng, X., Liu, C., & Yang, Q. (2011). Wisdom of the better few: cold start recommendation via representative based rating elicitation. *Proceedings of the Fifth ACM Conference on Recommender Systems*, (s. 37-44).

Logan, B. (2002). Content-based playlist generation: exploratory experiments. *Proceedings of the Third International Conference on Music Information Retrieval*, (s. 295-296).

Ludewig, M., Kamehkhosh, I., Landia, N., & Jannach, D. (2018). Effective nearest-neighbor music recommendations. *Proceedings of the ACM Recommender Systems Challenge 2018*, (s. 1-6).

Magnusson, M., Vehtari, A., Jonasson, J., & Andersen, M. (2020). Leave-one-out cross-validation for Bayesian model comparison in large data. *International Conference on Artificial Intelligence and Statistics* (s. 341-351). PMLR.

McFee, B., Lanckriet, G., & Jebara, T. (2011). Learning Multi-modal Similarity. *Journal of machine learning research, 12*(2).

Mimno, D., Wallach, H., Talley, E., Leenders, M., & McCallum, A. (2011). Optimizing semantic coherence in topic models. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, (s. 262-272).

Park, Y.-J., & Tuzhilin, A. (2008). The long tail of recommender systems and how to leverage it. *Proceedings of the 2008 ACM Conference on Recommender Systems*, (s. 11-18).

Pichl, M., Zangerle, E., & Specht, G. (2015). Towards a context-aware music recommendation approach: What is hidden in the playlist name? *2015 IEEE International Conference on Data Mining Workshop (ICDMW)* (s. 1360-1365). IEEE.

Pichl, M., Zangerle, E., & Specht, G. (2017). Improving context-aware music recommender systems: Beyond the pre-filtering approach. *Proceedings of the 2017 ACM Conference on Multimedia Retrieval*, (s. 2018-208).

Rashid, A. M., Karypis, G., & Riedl, J. (2008). Learning preferences of new users in recommender systems: an information theoretic approach. *Acm Sigkdd Explorations Newsletter, 10*(2), 90-100.

Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation, 60*(5), 503-520.

Rubtsov, V., Kamenshchikov, M., Valyaev, I., Leksin, V., & Ignatov, D. I. (2018). A hybrid two-stage recommender system for automatic playlist continuation. *Proceeedings of the ACM Recommender Systems Challenge 2018*, (s. 1-4).

Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. *Data mining and knowledge discovery, 5*(1), 115-153.

Schedl, M., & Schnitzer, D. (2013). Hybrid retrieval approaches to geospatial music recommendation. *Proceedings of the 36th ACM SIGIR Conference on Research and Development in Information Retrieval*, (s. 793-796).

Schedl, M., Knees, P., McFee, B., Bogdanov, D., & Kaminskas, M. (2015). Music recommender systems. *Recommender systems handbook* (s. 453-492). içinde Boston: Springer.

Schedl, M., Zamani, H., Chen, C.-W., Deldjoo, Y., & Elahi, M. (2018). Current challenges and visions in music recommender systems research. *International Journal of Multimedia Information Retrieval, 7*(2), 95-116.

Slaney, M., & White, W. (2007). Similaritry Based on Rating Data. *Proceedings of the Eight International Conference on Music Information Retrieval*, (s. 479-484).

Van Den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. *Neural Information Processing Systems Conference (NIPS 2013)*, *26.*

Vehtari, A., & Ojanen, J. (2012). A survey of Bayesian predictive methods for model assessment, selection and comparison. *Statistics Survey, 6*, 142-228.

Volkovs, M., Rai, H., Cheng, Z., Wu, G., Lu, Y., & Sanner, S. (2018). Two-stage model for automatic playlist continuation at scale. *Proceedings of the ACM Recommender Systems Challenge 2018*, (s. 1-6).

Wang, X., Rosenblum, D., & Wang, Y. (2012). Context-aware mobile music recommendation for daily activities. *Proceedings of the 20th ACM Internation Conference on Multimedia*, (s. 99-108).

Yang, H., Jeong, Y., Choi, M., & Lee, J. (2018). Mmcf: multimodal collaborative filtering for automatic playlist continuation. *Proceedings of the ACM Recommender Systems Challenge 2018*, (s. 1-6).

Yürekli, A., Bilge, A., & Kaleli, C. (2021). Exploring playlist titles for cold-start music recommendation: an effectiveness analysis. *Journal of Ambient Intelligence and Humanized Computing*, 1-20.

Zamani, H., Schedl, M., Lamere, P., & Chen, C.-W. (2019). An analysis of approaches taken in the acm recsys challenge 2018 for automatic music playlist continuation. *ACM Transactions on Intelligent Systems and Technology (TIST), 10*(5), 1-21.

Zhao, X., Song, Q., Caverlee, J., & Hu, X. (2018). Trailmix: an ensemble recommender system for playlist curation and continuation. *Proceedings of the ACM Recommender Systems Challenge 2018*, (s. 1-6).

Zhu, L., He, B., Ji, M., Ju, C., & Chen, Y. (2018). Automatic playlist continuation via neighbor-based collaborative filtering and discriminative reweighting/reranking. *Proceedings of the ACM Recommender Systems Challenge 2018*, (s. 1-6).

CURRICULUM VITAE

Full Name	:	Ali Yürekli
Foreign Languages	:	English, German

Educational Status

- Eskişehir Technical University, Institute of Graduate Programs, Doctor of Philosophy in Computer Engineering. — October 2021
- Anadolu University, Graduate School of Science, Master of Science in Computer Engineering. — June 2013
- Middle East Technical University, Faculty of Engineering, Bachelor of Science in Computer Engineering. — June 2010

Professional Background

- Research Assistant, Eskişehir Technical University, Faculty of Engineering, Computer Engineering Department, Eskişehir, Turkey. — 2019 – ∞
- Research Assistant, Anadolu University, Faculty of Engineering, Computer Engineering Department, Eskişehir, Turkey. — 2010 – 2019
- Software Engineer, Anadolu University, Computer Research and Application Centre, Eskişehir, Turkey. — 2010 – 2019

Articles Published in International Peer-Reviewed Journals

- Yürekli, A., Kaleli, C., & Bilge, A. (2021). Alleviating the cold-start playlist continuation in music recommendation using latent semantic indexing. *International Journal of Multimedia Information Retrieval*, 1-14.
- Mouheb, K., Yürekli, A., & Yılmazel, B. (2021). TRODO: A public vehicle odometers dataset for computer vision. *Data in Brief*, 107321.

- Yürekli, A., Bilge, A., & Kaleli, C. (2021). Exploring playlist titles for cold-start music recommendation: an effectiveness analysis. *Journal of Ambient Intelligence and Humanized Computing*, 1-20.
- Batmaz, Z., Yürekli, A., Bilge, A., & Kaleli, C. (2019). A review on deep learning for recommender systems: challenges and remedies. *Artificial Intelligence Review*, 52(1), 1-37.