ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ÉTUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre : 4012

THÈSE

en vue de l'obtention du : **DOCTORAT**

Structure de Recherche: Intelligent Processing and Security of Systems (IPSS)
Discipline : Informatique
Spécialité : Informatique et Intelligence artificielle

Présentée et Soutenue le : *16 / 11 / 2024*

par:

Rabie MADANI

AI-Powered Approaches to Advanced Contexts Collection and Ratings Prediction in Recommender Systems

Devant le JURY:

Mohamed BENKHALIFA	PES, Université Mohammed V, Faculté des Sciences, Rabat	Président
Mostafa EL MALLAHI	MCH, Université Sidi Mohammed Ben Abdellah, Ecole normale supérieure, Fès	Rapporteur/Examinateur
Abderrahim AIT WAKRIME	MCH, Université Mohammed V, Faculté des Sciences, Rabat	Rapporteur/Examinateur
Ahmed DRISSI EL MALIANI	MCH, Université Mohammed V, Faculté des Sciences, Rabat	Rapporteur/Examinateur
Ahmed EL-YAHYAOUI	MCH, Université Mohammed V, Faculté des Sciences, Rabat	Examinateur
M'hamed RAHMOUNI	MCH, Université Mohammed V, Ecole normale supérieure, Rabat	Examinateur
Amine ABOUAOMAR	PA, Université Al Akhawayn, Ifrane	Invité
Abderrahmane EZ-ZAHOUT	MCH, Université Mohammed V, Faculté des Sciences, Rabat	Directeur de thèse

Année Universitaire : 2023 - 24

Dedications

I dedicate this work

To my parents, may God protect and keep you.

To my dear wife, Jihane, for her support.

To my elder brother, Nabil, for his support and advice throughout my academic journey.

To my brothers, Achraf and Soufiane.

To my nieces and nephews.

To my sisters-in-law.

To all my friends and relatives.

Acknowledgements

This thesis was conducted in the Department of Computer Science at the Faculty of Sciences in Rabat, under the guidance of the Intelligent Processing Systems & Security (IPSS) team.
I extend my deepest gratitude to Professor **Fouzia OMARY**, Director of IPSS, for her warm welcome into her team and her steadfast support.
First and foremost, I am immensely grateful to my thesis advisor, Professor **Abderrahmane EZ-ZAHOUT**, for his expert guidance and supervision of this work. His rigorous scientific approach, invaluable advice, support, and trust enabled the development and expansion of my ideas, providing me with a new perspective on research.
I would like to extend my sincere thanks to Professor **Mohamed BENKHALIFA** for agreeing to serve as the president of my thesis committee and for generously dedicating his valuable time.
I would also like to express my gratitude to Professor **Mostafa EL MALLAHI** for kindly accepting the role of reviewer and for dedicating his valuable time to reviewing this manuscript.
I extend my heartfelt gratitude to Professor **Abderrahim AIT WAKRIME** for accepting the role of reviewer and for devoting his valuable time to reviewing this manuscript.
I am deeply grateful to Professor **Ahmed DRISSI EL MALIANI** for agreeing to serve as a reviewer and for devoting their valuable time and expertise to reviewing this manuscript.
Additionally, I would like to thank Professor **Ahmed EL-YAHYAOUI** and Professor for his agreement to examine my thesis and for the valuable time he dedicated to this important task.
I would also like to extend my heartfelt thanks to Professor **M'hamed RAHMOUNI** for kindly agreeing to examine my thesis and for dedicating his valuable time to this significant endeavor.

Abstract

Recommender systems (RS) are a subset of filtering systems primarily designed to deliver personalized recommendations by analyzing user-specific preferences. Traditional RS primarily utilize user and item data to generate recommendations but often neglect critical factors such contextual factors that could enhance the quality of these recommendations. Context-aware recommender systems (CARS) address this issue by integrating contextual information, which enables a deeper understanding of user preferences in various scenarios. However, implementing CARS poses several challenges, including the difficult task of obtaining contextual data due to users' privacy concerns. Additionally, incorporating contextual information into RS can lead to increased sparsity and dimensionality, posing further complications.

In this thesis, we introduce a range of models that leverage advanced techniques like Factorization Machines and deep learning to surmount the inherent challenges of CARS and develop robust, cross-domain recommender systems. These models are designed around two principal functions: the collection of contextual information function which aims to address the scarcity of such data, and the ratings prediction function that incorporates this data in order to provide more precise and accurate recommendations. The results from our experiments demonstrate that our proposed models significantly outperform the baseline models, thereby confirming their effectiveness.

Keywords: Recommender Systems, Context-aware recommender systems, Deep learning Factorization machines, Named Entity Recognition.

Résumé

Les systèmes de recommandation (SR) sont une sous-catégorie des systèmes de filtrage, principalement conçus pour fournir des recommandations personnalisées en analysant les préférences spécifiques des utilisateurs. Les SR traditionnels utilisent principalement des données sur les utilisateurs et les articles pour générer des recommandations, mais ils négligent souvent des facteurs critiques tels que les facteurs contextuels qui pourraient améliorer la qualité de ces recommandations. Les systèmes de recommandation sensibles au contexte (SRSC) résolvent cette problématique en intégrant des informations contextuelles, ce qui permet une compréhension plus approfondie des préférences des utilisateurs dans divers scénarios. Cependant, la mise en œuvre des SRSC présente plusieurs défis, y compris le problème de l'obtention de ces données en raison des préoccupations des utilisateurs concernant la confidentialité. De plus, l'incorporation d'informations contextuelles dans les SR peut compliquer le problème de la rareté des données et augmente considérablement le nombre des dimensions. Dans cette thèse, nous proposons une gamme de modèles qui exploitent des techniques avancées telles que les machines de factorisation et l'apprentissage profond pour surmonter les défis inhérents aux SRSC et développer des systèmes de recommandation multi-domaines robustes. Ces modèles sont conçus autour de deux fonctions principales : la fonction de collecte d'informations contextuelles qui vise à répondre à la rareté de ces données, et la fonction de prédiction des scores qui intègre ces données afin de fournir des recommandations plus précises et plus exactes. Les résultats de nos expériences démontrent que nos modèles proposés surpassent considérablement les modèles de base, confirmant ainsi leur efficacité.

Mots clés: Recommender Systems, Context-aware recommender systems, Deep learning
Factorization machines, Named Entity Recognition.

Résumé détaillé

Les systèmes de recommandation (SR) sont une branche essentielle des systèmes de filtrage, visant à fournir des recommandations personnalisées en analysant les préférences individuelles des utilisateurs. Traditionnellement, ces systèmes s'appuient principalement sur des données relatives aux utilisateurs (par exemple, leurs historiques d'interaction) et aux articles (caractéristiques des produits ou contenus) pour générer leurs recommandations. Cependant, cette approche classique présente des limites importantes, notamment l'absence de prise en compte des facteurs contextuels qui influencent de manière significative les préférences et les comportements des utilisateurs. Ces facteurs contextuels peuvent inclure, par exemple, le moment de la journée, la localisation géographique, l'humeur oules conditions météorologiques. Ignorer ces éléments peut limiter la pertinence et la précision des recommandations générées.

Les systèmes de recommandation sensibles au contexte (SRSC) s'efforcent de répondre à cette problématique en intégrant explicitement les informations contextuelles dans leurs modèles. En ajoutant une dimension contextuelle aux SR, ils permettent une compréhension plus fine des préférences des utilisateurs dans des situations spécifiques, augmentant ainsi la qualité et la personnalisation des recommandations. Cependant, la mise en œuvre de ces systèmes est loin d'être triviale et s'accompagne de défis considérables.

L'un des principaux défis des SRSC réside dans la collecte et l'exploitation des données contextuelles. Souvent, ces données sont difficiles à obtenir, car leur acquisition peut susciter des préoccupations importantes en matière de confidentialité de la part des utilisateurs. De plus, la collecte de ces données peut nécessiter des capteurs ou des dispositifs supplémentaires, augmentant ainsi la complexité technique et les coûts opérationnels. Un autre problème majeur est la rareté des données contextuelles, qui découle du fait que chaque situation ou contexte peut être unique, rendant difficile la généralisation des modèles. Cette rareté des données est exacerbée par le problème de "l’expansion des dimensions", puisque l'intégration des informations contextuelles augmente considérablement le nombre de

dimensions dans les modèles, compliquant davantage leur apprentissage et leur optimisation. Dans le cadre de cette thèse, nous proposons une approche novatrice visant à résoudre ces défis et à développer des systèmes de recommandation sensibles au contexte robustes, capables de fonctionner efficacement dans des environnements multi-domaines. Notre méthodologie repose sur deux axes principaux étroitement liés et complémentaires. Le premier axe concerne la collecte et l'extraction des données contextuelles, un défi majeur pour les systèmes de recommandation sensibles au contexte en raison de la rareté et de la complexité de ces données. Pour y remédier, nous proposons des techniques avancées basées sur l'intelligence artificielle, permettant de collecter passivement les données contextuelles pour limiter les efforts requis de la part des utilisateurs. Dans ce cadre, des algorithmes d'apprentissage profond sont mis en œuvre pour optimiser le processus de collecte en réduisant le nombre d'interactions nécessaires tout en maximisant l'information extraite. Le second axe se concentre sur la prédiction des scores intégrant le contexte, en s'appuyant sur des modèles hybrides combinant les dernières avancées en matière de machines de factorisation et d'apprentissage profond. Ces modèles sont conçus pour intégrer les informations contextuelles de manière transparente, améliorant ainsi la précision et la pertinence des recommandations.

Les résultats expérimentaux obtenus à l'aide de nos modèles montrent des améliorations significatives par rapport aux approches traditionnelles et aux méthodes de base dans divers scénarios et domaines d'application. En conclusion, cette thèse contribue à faire avancer l'état de l'art en proposant une nouvelle génération de systèmes de recommandation capables de s'adapter aux besoins des utilisateurs dans des contextes variés, tout en relevant les défis inhérents à la collecte et à l'intégration des données contextuelles. Ces travaux ouvrent également de nouvelles perspectives pour le développement de systèmes de recommandation multi-domaines, capables de répondre aux exigences de personnalisation et de pertinence dans des environnements de plus en plus complexes et dynamiques.

List of figures

List of abbreviations

RS	**R**ecommender **S**ystems
CARS	**C**ontext-**A**ware **R**ecommender **S**ystems
CBRS	**C**ontent-**b**ased **R**ecommender **S**ystem
VSM	Vector Space Model
TF	Term frequency
IDF	inverse document frequency
MDP	Markov Decision Process
RBM	Restricted Boltzmann Machine
SVD	Singular Value Decomposition
MDP	Markov Decision Process
RBMs	Restricted Boltzmann Machines
MF	Matrix Factorization
CFRS	Collaborative Filtering Recommender System
KBRS	Knowledge-based Recommender Systems
CBR	Case-based Reasoning
FC	Filtering Constraints
CC	Compatibility Constraint
PC	Product Constraints
UBRS	Utility-based recommender systems
SAU	Single-attribute utility
MAU	Muli-attribute utility
MAUT	Multi-Attribute Utility Theory
SBRS	Session-based recommender systems
FPM	Frequent Pattern Mining
NCF	Neural Collaborative Filtering
RNNs	Recurrent Neural Networks
CNNs	Convolutional Neural Networks
MTML	Multi-Theoretical Multi-Level
CCR	Content-Collaborative Reciprocal
CB-TARS	Community-Based Trust Aware Recommender System
RMSE	Root Mean Square Error
CBFM	Context-based Factorization Machines

Contents

Introduction

In this introduction, we provide a general overview of the thesis, starting with the background and motivation that underline the key research themes. This is followed by a detailed articulation of the problem statement, pinpointing the specific challenges the thesis aims to address. We then highlight the significant contributions of the research, demonstrating how this work advances knowledge in the field. The chapter concludes by describing the structure of the thesis, outlining how each part contributes to the overarching narrative and analysis.

Background and Motivation

Recommender Systems (RS) have become essential tools across various sectors due to the vast and continually expanding selection of products and services in today's market. Industries such as e-commerce, video streaming, cinema, travel, and particularly gaming, rely heavily on these systems. As a subcategory of data filtering technologies, RS are adept at providing personalized recommendations that align closely with individual user preferences [1].

The advantages of implementing RS are twofold. Firstly, they significantly enhance the user experience by reducing the time and effort needed to locate and select desirable items. This efficiency is not only convenient for users but also encourages continued engagement and satisfaction. Secondly, RS contribute directly to increasing a company's revenue by promoting products or services tailored to consumer preferences, thereby increasing the likelihood of purchase. The concept of Recommender Systems emerged as an independent area of research in the 1990s [1]. Since then, significant progress has been made in both industrial and academic settings, where researchers have developed a variety of innovative approaches to improve recommendation accuracy and relevance [2]. These advancements include the exploration of complex algorithms that analyze user behavior patterns, machine

learning techniques that adapt to user feedback, and hybrid models that combine different recommendation strategies to enhance system performance. The ongoing development of RS reflects their growing sophistication and critical role in facilitating effective consumer interaction in an increasingly digital marketplace. This continuous innovation is crucial for keeping pace with consumer expectations and maintaining competitive advantage.

Traditional Recommender Systems, which primarily operate within the user and item dimensions, aim to provide personalized recommendations based on these limited criteria. The content-based approach [3] expands upon this by integrating both item features and user profiles into the recommendation process. This method analyzes the specific characteristics of items along with user preferences and behaviors, enabling the system to offer recommendations that are finely tuned to individual users based on the content they interact with.

Conversely, Collaborative filtering approaches [4] offer a distinct model by eschewing the need for detailed user or item information. Instead, these systems rely on patterns in user ratings to infer preferences, building a model based on the assumption that similar users like similar items. While effective in many scenarios, this approach has notable limitations. It fails to consider a range of contextual information that can significantly influence user preferences. For instance, a user's location might dictate the kind of music they prefer at a given time, with more upbeat music at the gym and more relaxing tunes at home. Similarly, a user's mood can affect the type of movies they choose; perhaps preferring comedies when happy and dramas when sad. Moreover, the company a user keeps can also sway their choices, such as selecting a multiplayer video game when with friends and a single-player game when alone. The time of day can impact media consumption patterns, like watching news in the morning and movies at night. Additionally, weather conditions can influence activities, leading to choices like indoor activities on cold days and outdoor activities when it's warm.

Enhancing Recommender Systems to incorporate these contextual factors would dramatically improve their accuracy and relevance. By capturing and analyzing this contextual data, systems can adapt recommendations in real-time to match the changing

environments and states of users. For instance, integrating geolocation data could allow a system to suggest nearby restaurants or entertainment options, while mood detection technology could tailor music or video recommendations according to the user's emotional state. This integration of contextual information represents a significant evolution in the development of more adaptive and sensitive Recommender Systems, potentially leading to higher user satisfaction and engagement by aligning suggestions more closely with the user's current context and needs.

Context-aware Recommender Systems (CARS) [5] represent an advancement in the field of recommendation technology, designed to address some limitations inherent in traditional Recommender Systems (RS). Traditional RS primarily utilize user and item data to generate recommendations. In contrast, CARS incorporate additional contextual data to enhance the accuracy and relevance of their predictions. The core of the CARS approach is the integration of "context," which encompasses various external factors that influence customer behavior. This context varies significantly across different applications, making it essential to tailor CARS to the specific needs and dynamics of each application.

For example, consider a restaurant recommender system. Traditional methods might generate suggestions based on user profiles and restaurant characteristics, such as menus, food quality, and pricing. However, without considering contextual factors like the customer's current location and the distance to the restaurant, the recommendations may not be practical or appealing. A customer who receives recommendations for restaurants that are too far away might find these suggestions unhelpful and inconvenient. In essence, CARS extend the functionality of traditional RS by enriching the data model with contextual insights, thus offering a more dynamic and user-centered experience. This leads to smarter, more practical recommendations that are better aligned with the specific circumstances and needs of each user at any given moment.

The primary challenge for CARS is their dependence on acquiring sensitive and personal contextual data to enhance recommendation accuracy. However, the collection and use of such data have raised significant privacy concerns, especially in light of recent high-profile data breaches. A pertinent example is the Facebook-Cambridge Analytica scandal. In this

case, Cambridge Analytica accessed the data of millions of Facebook users without proper authorization. The data was then used to create detailed political profiles and targeted advertising campaigns, which were intended to influence voter behavior. This breach was particularly alarming because it highlighted not only the ease with which data could be misused but also the insufficient protections in place to prevent such incidents [6]. This scandal has had a broad impact, making users increasingly cautious about sharing their personal information. They are now more aware of the potential misuse of their data and the consequences it can have on their privacy and personal lives [7, 8]. This heightened awareness has led to increased public and regulatory scrutiny of how companies collect, store, and utilize personal data.

The reluctance of users to share their data presents a significant impediment to the effectiveness of CARS. The absence of rich contextual data significantly undermines these systems' capacity to generate highly personalized and relevant recommendations. Consequently, there is an urgent need for CARS to identify new data sources and develop creative approaches for gathering contextual information.

In addition to the primary challenge of data privacy, CARS also grapple with issues such as data sparsity and high dimensionality. Data sparsity occurs when there is a limited number of interactions or ratings between users and items. Interestingly, the inclusion of contextual data, while enriching the dataset, can actually intensify the sparsity problem. This happens because adding more dimensions to the data introduces more possibilities for missing data points, which complicates the recommendation process further. Data sparsity refers to significant gaps within the dataset, whereas high dimensionality is caused by the incorporation of new variables that expand the overall complexity of the dataset.

Problem statement

As highlighted in the thesis background, traditional recommender systems have generally neglected contextual information, focusing primarily on user and item data to generate recommendations. Yet, integrating contextual data can significantly improve the quality of these recommendations by aligning them more closely with users' preferences in specific

situations or circumstances, based on the idea that similar contexts predict similar preferences.

The potential benefits of such context-aware recommender systems (CARS) are substantial. By aligning recommendations more closely with the current context of the user, systems can achieve higher user engagement, improved satisfaction, and increased accuracy in meeting user needs. Moreover, the integration of contextual information can substantially enhance the accuracy of recommendations and provide personalized, well-suited suggestions for users in various scenarios.

However, collecting this data poses a major challenge, especially since it often involves personal information. This difficulty is compounded by issues like data sparsity and high dimensionality, which are exacerbated by the introduction of new, context-related data dimensions.

To maximize the benefits of context-aware recommender systems and create high-quality, personalized recommendations, several critical questions need to be addressed:

- Data Collection Approaches: How can contextual data be gathered indirectly or inferred from user interactions rather than directly solicited from users and What are the potential sources from which contextual data can be extracted without compromising user privacy?
- Model Efficiency: How can we develop effective models that collect contextual data and require minimal manual input in terms of feature engineering, particularly those that utilize publicly available data?
- Integration and Optimization: How can contextual information be seamlessly integrated into recommender systems while addressing issues like data sparsity and high dimensionality?
- Contextual data usefulness: Does the contextual data collected enhance the quality of recommendations?

By addressing these questions and challenges, context-aware recommender systems can significantly enhance the personalization and relevance of recommendations, leading to

improved user satisfaction and engagement.

Contributions

The research conducted within this thesis significantly advances the field of Context-aware Recommender Systems (CARS) by enhancing their effectiveness and overcoming some of their inherent limitations. The key contributions of this study are outlined as follows:

- Development of a Comprehensive CARS: This thesis details the creation of a full-fledged CARS that encompasses the entire process from collecting contextual information to predicting ratings and generating tailored recommendations. This system is designed to seamlessly integrate various types of contextual data, ensuring a holistic approach to personalized recommendation. Additionally, the model is cross-domain, meaning it is capable of applying its techniques across different domains, such as e-commerce, movies recommendation, and last but not least restaurant recommendation, further enhancing its utility and adaptability in providing accurate recommendations in diverse environments.
- Innovative Models for Contextual Information Extraction: The research introduces advanced methodologies for extracting contextual data from unstructured sources, such as user reviews. These novel models are specifically tailored to assimilate this data effectively, thereby enhancing the accuracy of predictions. A key focus of these models is to solve the issue of data lack while integrating context into the recommendation process, showcasing a balance between personalization and privacy concerns.
- Creation of a Vast Contextual Information Corpus: By leveraging open knowledge resources, this research has constructed a large-scale corpus of contextual information. This corpus serves as a critical asset for training our models, providing a rich dataset that enhances the system's ability to understand and predict user preferences under various contextual scenarios.
- Development of New Models to Handle Sparsity and High Dimensionality:

Addressing two of the most challenging issues in recommender systems, this thesis presents new models designed to incorporate freshly collected contextual data. These models are crafted to mitigate problems associated with data sparsity and high dimensionality, thereby improving the system's overall predictive performance and recommendation relevance.

Together, these contributions represent a significant step forward in the refinement and application of Context-aware Recommender Systems, providing a robust framework for delivering more accurate, personalized user experiences across diverse application settings.

Thesis structure

The thesis is structured into two main parts. The first part offers a detailed literature survey on recommender systems, with a specific emphasis on Context-aware Recommender Systems. The second part details the various models developed for extracting contextual information and predicting ratings. The contents of this thesis are elaborated further as follows:

Part I. State-of-the-art: Recommender systems and context-aware recommender systems.

- **Chapter 1** offers an overview of the current state of recommender systems, examining traditional, special and advanced approaches, discussing the challenges, and the motivation for CARS.
- **Chapter 2** provides an in-depth review of the state of the art in context-aware recommender systems. This includes a classification of the primary methods used in the literature to exploit contextual information, along with an exploration of various challenges these systems face.

Part II. Context extraction and ratings prediction models

- **Chapter 3** delves into the definition of the context concept and explores how it is applied within the framework of recommender systems. It describes the process of

building a corpus using open knowledge resources, which serves as a foundational element for training and testing the models discussed. Additionally, the chapter introduces advanced techniques and models specifically designed for extracting contextual data from unstructured sources, such as user reviews. This involves discussing the various methodologies and technologies used to analyze and interpret the text data to glean relevant contextual information that can enhance the accuracy and relevance of recommendations.

- **Chapter 4** delves into newly developed models that successfully integrate recently gathered contextual data into the recommendation process. It examines how these models address two significant challenges in recommender systems: sparsity and high dimensionality. Furthermore, this chapter details the specific algorithm selected for this purpose, explaining the rationale behind its choice and highlighting its advantages in managing the complexities of contextual data within recommender systems. Additionally, this chapter outlines the limitations of the chosen algorithm and discusses the proposed solutions to overcome these challenges, enhancing the overall effectiveness of the recommender system.
- **Conclusion and future work:** conclude the thesis by summarizing the main contributions and discussing potential avenues for future research.

Part I

State-of-the-art: Recommender systems and context-aware recommender

Chapter 1

Recommender Systems

This chapter presents a detailed analysis of the current status of recommender systems, encompassing a wide spectrum of approaches ranging from traditional to advanced methodologies. It explores the evolution of these systems, highlighting the progression from classical techniques like collaborative filtering and content-based filtering to more sophisticated methods such as deep learning-based models, and graph-based models.

Throughout this exploration, the chapter delves into the operational intricacies and challenges faced by recommender systems in diverse application domains. It examines how these systems adapt to varying data complexities, user behaviors, and scalability requirements. Furthermore, the chapter sheds light on the efforts to enhance recommendation accuracy, address the cold start problem, and the motivation for incorporating contextual information to personalize recommendations.

1.1 Traditional Recommender Systems

Traditional recommender systems aim to provide personalized recommendations to users based on historical data such as user interactions or item attributes. These systems leverage user preferences and item characteristics to suggest items that align with individual tastes and interests. By analyzing patterns in user behavior and item features, traditional recommender systems generate recommendations that cater to specific user needs and enhance user experience. These systems are foundational in the field of recommendation technology, enabling businesses to deliver targeted suggestions and improve customer engagement and satisfaction. Here are key traditional recommender system approaches:

1.1.1 Content-Based approach

Content-based recommender systems (CBRSs) rely on analyzing item and user descriptions to create representations of items and profiles of users. These systems recommend items that share similar attributes to those liked by a target user in the past. The process involves comparing the attributes of a user's profile (which stores preferences and interests) with the attributes of available items, resulting in a relevance score that predicts the user's potential interest in those items. Item attributes used for description can come from metadata associated with the item or directly from textual features extracted from item descriptions. However, metadata often lacks depth, while using textual features presents challenges due to the nuances and ambiguities of natural language, making it complex to accurately learn user preferences and construct effective user profiles in content-based recommendation systems. This highlights the importance of robust techniques to handle textual data and improve recommendation accuracy.

In recommending news articles to a user, a CBRSs aims to identify specific keywords that are frequently present in articles the user finds interesting. The system then uses these identified keywords to suggest news articles that contain similar or related keywords. This approach focuses on analyzing the content of articles that align with the user's preferences, leveraging keyword matching to recommend relevant content. This process emphasizes the utilization of content attributes, such as keywords extracted from articles, to personalize recommendations based on the user's demonstrated interests and article content similarity [9].

Most content-based recommender systems use relatively simple retrieval models [10], such as keyword matching or VSM (Vector Space Model). Because of the significant and early advancements made by information retrieval, many content-based systems focus on recommending items containing textual information. TF/IDF is considered as among the largest used term weights measure.

Formally, let d_j represent the item description of item j. In CBRSs, item descriptions are constructed by extracting a set of features, typically keywords, that describe the characteristics of the items. The importance or relevance of a keyword k in the item

description d_j is denoted by the weight wk_j. The term frequency (TF) of a keyword k in the description of item j is calculated as the number of occurrences *Nk* of keyword *k* divided by the total number of terms TN in the description:

$$TF = NK/TN \quad (1.1)$$

Here, TN represents the total number of terms (or words) in the item description d_j. This TF value indicates the relative frequency of a specific keyword within the context of the item's description, enabling the system to assess the significance of keywords in describing and characterizing items for recommendation purposes in CBRSs.

The inverse document frequency (IDF) is used in conjunction with term frequency (TF) to assess the significance of words in item descriptions for recommendation purposes. The IDF of a word *i* is defined as:

$$IDF = \frac{log\ N}{DF} \quad (1.2)$$

Where, *N* is the total number of item descriptions (or documents) in the dataset and *DF(k)* is the document frequency of the word k, which represents the number of item descriptions containing the word *k*. The TF-IDF weight *wij* for word k in the item description is defined as follows:

$$w_{kj} = TF \times IDF \quad (1.3)$$

Given that item descriptions can be represented as vectors, the similarity between two descriptions can be assessed using vector-based scoring heuristics like the cosine similarity measure [11]. Consequently, in a CBRS, items most akin to the user's favored ones are suggested. By amalgamating item descriptions previously liked by the user, a user profile is constructed [12]. Hence, the resemblance between items can be computed as follows:

$$Similarity(di, dj) = \frac{\left(\sum_{i=1}^{n} w_{kj}\ .w_{ki}\right)}{\sqrt{\sum_{i=1}^{n} w_{kj}^2}\ \sqrt{\sum_{i=1}^{n} w_{ki}^2}} \quad (1.4)$$

Items that have the highest similarity scores are suggested to the user as recommendations.

The development in semantic technologies has significantly influenced the evolution of CBRSs. As the field matures, researchers are shifting away from traditional keyword-based methods of representing items and user profiles. Instead, they are adopting concept-based approaches [13] that offer a more nuanced understanding of content. This shift enables CBRSs to analyze and interpret the deeper meanings and relationships embedded within the data.

Concept-based representation leverages semantic knowledge graphs and ontologies that link various concepts according to their semantic relationships. This allows systems to recognize and recommend items not just based on superficial keyword matches but on the underlying concepts they represent. For instance, a CBRS can understand that a documentary on "climate change" is conceptually related to articles about "global warming" or "environmental policy," even if the exact terms do not match.

This approach enhances the recommendation process by making it more accurate and relevant to the user's true interests and needs. It also allows for more sophisticated user profiles that accumulate insights based on conceptual preferences rather than just tracking keyword interactions. As a result, recommendations become more personalized and effective, potentially increasing user engagement and satisfaction. Furthermore, the integration of open knowledge sources enriches the semantic base of CBRSs, providing a broader and more dynamic foundation of data from which to draw inferences. This availability of rich, structured semantic data opens new avenues for research and application, expanding the potential of CBRSs to operate across diverse domains and data types.

CBRSs are effective in various applications but face several critical limitations. One significant issue is over-specialization, where systems may create a "filter bubble" by recommending items excessively similar to those previously liked, limiting diversity and potentially leading to user dissatisfaction [14]. Additionally, the performance of CBRS heavily relies on the availability and quality of item metadata; insufficient or poor-quality metadata can lead to inadequate recommendations [16]. Another challenge is the new item problem, where items without user interaction history are less likely to be recommended, thus diminishing the visibility of new content [9]. Scalability also becomes an issue as the

number of items and the dimensionality of feature spaces grow, posing significant computational challenges [11]. Moreover, the subjectivity of content analysis can result in recommendations that do not fully capture user preferences, as many desirable item aspects are subjective and not easily quantified through standard feature extraction methods [10]. Finally, CBRS often lacks serendipity, tending to recommend predictable items, thus missing opportunities to enhance user experience through unexpected discoveries [16]. These limitations underscore the need for continuous innovation in CBRS, including the development of hybrid models, improved feature extraction techniques, and strategies to increase recommendation diversity and serendipity.

1.1.2 Collaborative filtering approach

Filtering is the process of identifying and retrieving a relevant subset of items from a larger pool that aligns with a user's interests [17]. Conversely, early information retrieval systems required user engagement in only a singular activity, which quickly proved inadequate as the deeper, essential aspects of a user's informational needs began to surface during their initial interactions with the system [18].

Collaborative filtering is a technique used in recommendation systems and is aimed at predicting the rating or preference that a user would give to an item. It has become an important and active area of research in the past few years. Collaborative filtering has many forms, but can be generally defined as methods for the automatic generation of predictions about the interests of a user by collecting preference information from many users [19]. In other words, collaborative filtering is based on the idea that users similar to you can be used to predict how much you will like a particular product or service that those users have used to rate. It could be movies, music, books, web pages, or whatever. The user of collaborative filtering is the one who provides item ratings, while preference predictions are for items without ratings.

Generally, collaborative methods in recommendation systems can be categorized into two main types [20]:

- Memory-Based Methods: These approaches utilize heuristics to predict user ratings

based on a complete dataset of previously rated items. Essentially, they estimate the unknown rating for a specific user-item pair by aggregating the ratings from other users who are similar to the target user.

- Model-Based Methods: These techniques involve constructing a probabilistic model using the existing ratings data. Once the model is developed, it is then applied to predict unknown ratings. This process allows the system to make informed guesses about user preferences based on learned patterns.

1.1.2.1 Memory-Based Collaborative approach

Memory-based recommender systems are a type of collaborative filtering algorithm that generate predictions using the entire database of user ratings on items. These systems operate on the principle that all human experiences and decisions are informed by past memories. Essentially, people tend to repeat behaviors that they have previously engaged in. This methodology stands out because it leverages the complete dataset to forecast preferences, making it uncomplicated yet potentially less efficient as the volume of data increases. Typically, these systems are categorized into two principal forms: user-based [21] and item-based [22] collaborative filtering.

- **User-Based Collaborative Filtering**: In user-based collaborative filtering, recommendations are produced by analyzing the similarities among users. The fundamental principle here is that users with similar preferences in the past are likely to have similar tastes in the future. To determine how similar two users are, various methods can be employed, among which the Pearson correlation coefficient [23] and cosine similarity [22] are the most commonly used. These similarity metrics help identify users who share preferences and guide the recommendation of items they might enjoy. To compute the similarity using the Pearson correlation, Let S be the set of items rated by both user u and user v. The similarity between the two users, similarity (u , v), is given by:

$$\text{similarity}(u,v) = \frac{\sum_{i\in S}\left(R_{ui} - \overline{(R_u)}\right)\left(R_{vi} - \overline{(R_v)}\right)}{\sqrt{\left(\sum_{i\in S}\left(R_{ui} - \overline{(R_u)}\right)^2 \sqrt{\left(\sum_{i\in S}\left(R_{vi} - \overline{(R_v)}\right)^2\right)}\right)}} \quad 1.5$$

R_{UI} and R_{VI} are the ratings from user u and v or item i, respectively, and $\overline{Ru}$ and $\overline{Rv}$ are the mean ratings for each user. To calculate the similarity between two users u and v using cosine similarity, the following formula is applied:

$$\text{similarity}(u,v) = \frac{\vec{u}\cdot\vec{v}}{||\vec{u}||||\vec{v}||} \quad 1.6$$

Where, $\vec{u}$ and $\vec{v}$ represent the rating vectors for users u and v respectively. The denominator $||\vec{u}||\ ||\vec{v}||$involves the magnitudes (or norms) of these vectors, calculated as the square root of the sum of the squares of their components.

- **Item-Based Collaborative Filtering:** Item-based collaborative filtering is a powerful technique extensively utilized across various sectors to enhance user experience through personalized recommendations. Its ability to efficiently handle large datasets and provide stable, comprehensible recommendations establishes it as a crucial component of contemporary recommender systems. Unlike methods that analyze user-to-user relationships, item-based collaborative filtering concentrates on the relationships between items themselves. The process of item-based collaborative filtering can be broken down into several key steps: First, Initially, the system determines the similarities between different pairs of items. The similarity can be quantified through several methods, including cosine similarity, Pearson correlation, or adjusted cosine similarity. The selection of a specific similarity metric is influenced by the characteristics of the data and the requirements of the application. Cosine Similarity for Items can be defined as follows:

$$\text{similarity}(i,j) = \frac{\sum_{u\in U} R_{ui} \times R_{uj}}{\sqrt{\sum_{u\in U} R_{ui}^2} \times \sqrt{\sum_{u\in U} R_{uj}^2}} \quad 1.7$$

R_{ui} and R_{uj} are the ratings given by user u to items i and j respectively, and U is the set of all users who have rated both items. After establishing item similarities, the system predicts a user's interest in an item by aggregating the ratings of other items that the user has previously rated, adjusting these ratings based on the computed similarities. This process frequently includes normalizing the ratings to mitigate any bias towards items that have been rated frequently or have higher average ratings. The formula for the predicted rating is articulated as follows:

$$\widehat{R_{ui}} = \frac{\sum_{j \in N(i;u)} \text{similarity}(i,j) \times R_{uj}}{\sum_{j \in N(i;u)} |\text{similarity}(i,j)|} \qquad 1.8$$

Where, $N(i;u)$represents the set of items similar to item i that have been rated by user u. Item-based collaborative filtering offers several distinct advantages that make it particularly effective for recommendation systems. First, it benefits from stability, as item properties generally remain more constant over time compared to user preferences, leading to less volatility and more consistent recommendations. Secondly, scalability is enhanced through the use of a precomputed item-item similarity matrix, which can be periodically updated, allowing the system to efficiently manage a large user base in real-time. Lastly, item-based methods provide a high level of transparency, offering users understandable explanations for recommendations, such as showing items that others with similar tastes have liked, which is a significant advantage over other collaborative filtering techniques.

1.1.2.2 Model-Based Collaborative approach

Model-based methods in collaborative filtering are an advanced set of techniques used in recommender systems to predict and recommend items to users based on their preferences. Unlike memory-based methods, which directly use the entire user-item dataset to find similarities and make recommendations, model-based methods involve building a predictive model from user data to infer user preferences [24]. Among the notable approaches are the Markov Decision Process (MDP), Restricted Boltzmann Machine (RBM), Singular Value Decomposition (SVD), and matrix factorization techniques, each addressing different challenges within the recommendation process.

Markov Decision Process (MDP) [25] is a mathematical framework designed for modeling decision-making where outcomes are partially random and partially under the control of a decision-maker. This model is particularly useful for sequential recommendation scenarios, such as in video streaming or online shopping, where each recommendation can influence subsequent user choices. By considering each action as leading to a new state, MDPs are adept at optimizing long-term user satisfaction, not just immediate rewards. This sequential decision-making framework is essential for enhancing user engagement over time, which is crucial for services where sustained interaction is beneficial. However, implementing MDPs comes with its challenges, primarily in scalability and data requirements; MDPs require extensive computation, particularly with large state and action spaces, and they depend on detailed data on user interactions which may not always be readily available. These factors make MDPs resource-intensive, both in terms of computational power and data management.

Restricted Boltzmann Machines (RBMs) [26] offer a different set of advantages, particularly in dealing with the high-dimensional data typical of user-item interaction matrices. RBMs are a type of neural network that excels in feature learning, capable of discovering intricate patterns in user preferences that are not readily apparent. This capability makes RBMs ideal for generating latent factors in collaborative filtering, thereby addressing data sparsity—one of the most pervasive challenges in recommender systems. They can effectively predict missing entries in user-item matrices, facilitating more accurate recommendations. Despite their strengths, RBMs also present significant challenges, particularly in training and interpretability. They require careful parameter tuning and substantial computational resources to train, especially with large datasets. Moreover, the latent features RBMs learn are often difficult to interpret, which can be a disadvantage in applications where understanding the basis for recommendations is important. Singular Value Decomposition (SVD) [27] and Matrix Factorization [28] are other prominent techniques in model-based recommendation systems. SVD is a matrix factorization method that decomposes a user-item matrix into three lower-dimensional matrices, revealing the latent structures within the data. This approach is highly effective at predicting ratings and is widely used in scenarios with large, sparse datasets. General matrix factorization techniques extend this idea by incorporating additional factors into the decomposition process, such as user or item biases,

which can further refine recommendation accuracy. Both SVD and matrix factorization share common advantages, such as the ability to handle very large datasets efficiently and improve prediction accuracy by capturing deeper patterns in the data. However, like RBMs, these methods also involve complex model training processes and are susceptible to overfitting, especially when the model parameters are not adequately regularized.

Despite all the advantages of CFRSs, they encounter significant challenges including scalability, cold start problems, data sparsity, and popularity bias. As the user and item base expands, scalability issues become pronounced, increasing the computational demands of these systems; Sarwar et al. [27] delve into this issue, suggesting dimensionality reduction as a potential solution. The cold start problem, accentuated by Schein et al. (2002), complicates effective recommendation for new users and items lacking substantial interaction data. Additionally, data sparsity issues, discussed by Adomavicius and Tuzhilin (2005), hinder the identification of similar users or items due to the prevalence of unknown values in the user-item interaction matrix. Finally, popularity bias, which leads to a disproportionate recommendation of popular items and potentially overlooks niche but relevant options for certain users, is a challenge addressed by Celma and Herrera (2008) who recommend new metrics for evaluating novel recommendations. Addressing these challenges is crucial for the continuous improvement and effectiveness of CFRS in delivering accurate, relevant, and satisfying user experiences.

1.2 Special Recommender systems

1.2.1 Knowledge-Based approach

Traditional recommendation methods excel at suggesting products driven by personal preferences and quality, like books, movies, or news. However, these approaches often struggle with more complex items such as cars, computers, real estate, or financial services. For instance, financial services are not typically contracted frequently, which poses a challenge for collaborative filtering algorithms that rely on accumulating ratings for specific items to function effectively.

Knowledge-based recommender systems (KBRS) [29] solve this problem by utilizing explicit knowledge about the relationship between users and items, rather than solely relying on past interaction data like collaborative and content-based filtering systems do. These systems are particularly useful in scenarios where there is limited user interaction data or when items do not have enough historical data to build a model. For instance, in a travel recommendation scenario, a knowledge-based system might employ an ontological description of destinations. This would include hierarchical concepts and relationships, such as recognizing Paris as a part of France, known for attractions like the Eiffel Tower and the Louvre Museum. If a user shows interest in France, unlike a collaborative filtering system that might indiscriminately recommend popular tourist sites to all users, a knowledge-based system leverages its structured knowledge to suggest specifically relevant Parisian attractions to that user, ensuring the recommendations are both accurate and actionable.

Furthermore, knowledge-based systems can extend their utility beyond mere location-based information to logistical aspects of travel. They can provide invaluable assistance to both local and international tourists by recommending optimal transportation routes and vehicle rentals tailored to their specific itinerary. Many foreign tourists face challenges understanding local public transportation systems; a knowledge-based system can alleviate this by mapping out the most efficient routes and modes of transport. Additionally, for those preferring not to rely on public transport, the system can suggest car rental options, enhancing travel experience and convenience. This tailored approach not only saves time and reduces frustration but also opens up opportunities for local tourists to explore car rental as a viable option for their travels, demonstrating the versatility and user-centric nature of knowledge-based recommender systems. There are several primary types of KBRS: case-based Reasoning (CBR) approach [29, 30] and constraint-based approach [31, 32]. In terms of their functionality and the knowledge they utilize, these systems share many similarities: they gather user requirements, generate recommendations based on how well items match those requirements, suggest modifications for any inconsistent requirements when no suitable options are found, and provide explanations for their recommendations. However, the primary distinction lies in the methodology used to compute these solutions. Case-based systems generate recommendations based on similarity metrics, identifying items that closely

resemble user specifications. On the other hand, constraint-based systems rely heavily on predefined knowledge bases, which include explicit rules that correlate customer requirements with specific item attributes.

1.2.1.1 Case-Based Reasoning

Case-Based Reasoning (CBR) is an approach used in various fields such as law, medicine, and customer support, which solves new problems based on the solutions of similar past problems. This paradigm operates on the premise that similar problems have similar solutions, leveraging previous cases as a basis for current decision-making. Each case in a CBR system encapsulates real-world scenarios, typically including both a problem and its successful solution.

The functionality of CBR is structured around four sequential phases known as the "Four Rs": retrieve, reuse, revise, and retain. Initially, the system retrieves from its database a case

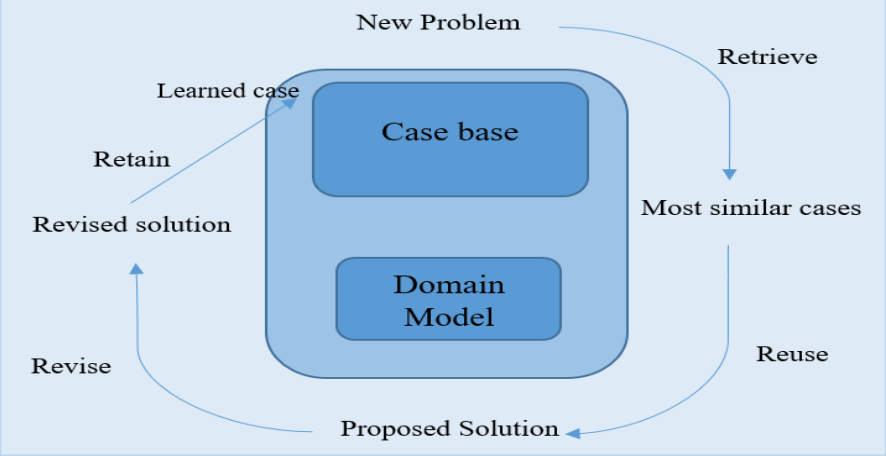

Figure 1.1: Case-based reasoning cycle.

similar to the new problem at hand, using sophisticated similarity assessment techniques.

This phase is crucial as the accuracy of the similarity measurement significantly influences the outcome. Following retrieval, the solution from the past case is reused and adapted if necessary to fit the new problem's specifics, which is the second phase. In the revise phase, this solution is tested and refined to ensure it effectively resolves the new issue, adjusting based on real-world feedback or simulated outcomes. Finally, the retain phase involves

storing the new problem and its solution as a part of the system's growing case database, thereby enriching future problem-solving. The four steps cyclical resolution issues method is shown in figure 1.1.

The iterative learning process of CBR offers several advantages, including the system's ability to evolve and improve over time, gaining efficiency and accuracy with each new case. This adaptability makes CBR particularly valuable in dynamic environments where problem parameters frequently change. Additionally, CBR systems reduce the knowledge engineering burden typical of more static AI systems, as they do not require extensive rule definition and domain modeling. Instead, they operate on practical examples from previous cases, which also enhances their ability to explain their recommendations or decisions clearly to users.

However, the effectiveness of CBR systems hinges on the quality and breadth of the case database. A rich collection of diverse and well-documented cases enhances the system’s ability to match and adapt solutions effectively, whereas a sparse or poor-quality case base can significantly impair performance. Managing the case database as it grows can also become challenging, necessitating efficient indexing and possibly the pruning of outdated or irrelevant cases to maintain system performance. Additionally, measuring case similarity can be complex, requiring sophisticated algorithms, especially when cases involve multiple dimensions or heterogeneous data.

1.2.1.2 Constraint-Based approaches

Constraint-based recommender systems utilize user-specified constraints to sift through and exclude items that do not meet those specific requirements. These constraints might range from straightforward criteria like price limits to more complex ones involving multiple attributes that must fulfill certain logical condition, for example, specific combinations of color, size, and brand in electronic products. The success of these systems is rooted in their capability to interpret and process the inputs from users, which clearly delineate the desired attributes of the items. This approach differs from collaborative filtering, which bases recommendations on user interaction data, and from content-based systems, which rely on

the characteristics of the items themselves. Instead, constraint-based systems exclusively rely on the constraints set by the user during the query process.

Constraint-based approaches are characterized by several distinct features. Firstly, they require users to provide explicit constraints, detailing precisely what they are seeking. This can include specifying a range of acceptable values for certain item attributes or selecting specific characteristics desired in the item. The systems utilize advanced filtering techniques to sift through the database, employing complex query mechanisms capable of handling multiple, often interdependent, constraints. Additionally, many of these systems offer a dynamic user interface, enabling users to continuously refine their preferences based on the feedback they receive from the system, thus allowing for the dynamic adjustment of constraints. Lastly, these systems are highly specialized and domain-specific, integrating extensive domain knowledge to effectively process and interpret the constraints specified by users, ensuring that recommendations are both relevant and precise.

In a constraint-based, the foundational structure can technically be described through the lens of a constraint satisfaction problem [33]. This framework consists of two sets of variables VC (variables related to customer preferences) and VPROD (variables related to product attributes) and three distinct sets of constraints CR (requirements constraints), CF (filtering constraints), and CPROD (product constraints). These elements are integral to the system's ability to generate recommendations.

A constraint satisfaction problem, which forms the theoretical basis of these systems, requires finding solutions that involve assigning values to these variables in such a way that all imposed constraints are met. This means that every solution to the problem corresponds to a combination of product attributes and customer preferences that adhere perfectly to the specified requirements, filtering criteria, and product-related constraints. This methodology ensures that recommendations are not only accurate but also highly personalized, adhering strictly to user-specified needs and conditions.

The following example provided illustrates how these elements interact to filter and recommend properties that meet specific user requirements.

- User Variables (U):
  - u1: Marital status (Married, Single)
  - u2: Family Size (integer)
  - u3: Minimum Bedrooms (integer)
  - u4: Maximum Bedrooms (integer)
  - u5: Maximum Price (integer)
- Product Variables (P):
  - p1: ID (integer)
  - p2: Bedrooms (integer)
  - p3: Price (integer)
- Compatibility Constraint (CC): This ensures that user properties are compatible. For example, a Family Size = 6 is deemed incompatible with Max Bedrooms = 2, meaning that such a scenario would automatically eliminate any listings with only two bedrooms from the recommendations for a family of six.
- Product Constraints (PC): These are predefined characteristics of available properties:
  - PC1: Property with ID = 1, Bedrooms = 3, Price = 20,000
  - PC2: Property with ID = 2, Bedrooms = 5, Price = 40,000
  - PC3: Property with ID = 3, Bedrooms = 2, Price = 15,000
- Filtering Constraints (FC): These create links between users' requirements and properties:
  - FC1: If Family Size (u2) = 6, then Bedrooms (p2) > 4
  - FC2: If Minimum Bedrooms (u3) = 3 and Maximum Bedrooms (u4) = 5, then the property price (p3) should be between 2,000 and 4,000

- Filtering Constraints (FC): These create links between users' requirements and properties:
  - FC1: If Family Size (u2) = 6, then Bedrooms (p2) > 4
  - FC2: If Minimum Bedrooms (u3) = 3 and Maximum Bedrooms (u4) = 5, then the property price (p3) should be between 2,000 and 4,000

Let's consider a user, who is looking to buy a home and has specified the following requirements:

- Marital Status: Married
- Family Size: 6
- Minimum Bedrooms: 4
- Maximum Bedrooms: 5
- Maximum Price: 50,000

Given user's specifications, the system would process her input as follows:

- User's Family Size = 6 would require a property with more than 4 bedrooms due to the compatibility constraint that a large family size is incompatible with a small number of bedrooms.
- Among the predefined product constraints, only PC2 (ID = 2, Bedrooms = 5, Price = 40,000) meets the bedroom requirement for user's family size.
- FC1 would be activated since user's family size is 6, thus requiring properties with more than 4 bedrooms, which aligns with PC2.
- However, FC2 does not activate because user’s price range does not match the specified range (2,000 to 4,000).

As a result, the only property that would be recommended to the user based on his constraints and the system's knowledge base is the property with ID = 2 (5 bedrooms, priced at 40,000). This property meets all his specified needs in terms of bedrooms and price, providing a

targeted and efficient recommendation.

Knowledge-based recommender systems offer significant advantages, especially, handle complex, multi-attribute user queries that might challenge simpler collaborative or content-based systems. Users can specify various attributes and requirements, and these systems can efficiently sift through extensive data to identify items that fulfill all specified criteria. The transparency and explainability of the recommendations provided by knowledge-based systems are another significant advantage. Since the recommendations are derived from explicit rules or a clear understanding of item attributes and their relationships, these systems can offer users understandable and justifiable reasons for their recommendations, which helps in building trust and enhancing user satisfaction.

However, the deployment of knowledge-based systems comes with considerable challenges, primarily the substantial effort required to create, structure, and maintain the knowledge base. The effectiveness of these systems is directly tied to the completeness, accuracy, and currency of this knowledge, which demands meticulous crafting and regular updates by domain experts and data scientists. Additionally, as the domain of application evolves, so too must the knowledge base, which can be resource-intensive. Maintenance overheads and scalability are further challenges. As the complexity of the knowledge base increases, the system may become harder to scale, dealing with a larger or more complex set of items and user requirements can lead to increased processing times and higher computational demands. Moreover, while the structured foundation of predefined rules in knowledge-based systems is beneficial for transparency, it can also make the system less flexible. This rigidity might hinder the system's ability to adapt to unforeseen user needs or preferences that fall outside the predefined rules.

1.2.2 Utility-Based Recommender Systems

Utility-based recommender systems (UBRS) [34] are designed to predict and recommend items based on a calculated utility value that reflects the predicted satisfaction or benefit that a user would derive from each item. Unlike traditional recommender systems that primarily focus on predicting items based on similarity or past interactions (such as collaborative or

content-based filtering), utility-based systems assess the utility of each item for a user. This approach stems from the principles of utility theory in economics, which suggests that individuals make decisions to maximize their satisfaction, often expressed in terms of "utility."

The principle that the likelihood of a product being purchased increases with the utility gap between it and the next best alternative is well-supported by economic and psychological theories. For instance, Luce [35] suggests that the probability of choosing an item is a function of its relative advantage in terms of utility over other available items. Similarly, in the realm of behavioral economics, Kahneman and Tversky's [36] Prospect Theory posits that decisions are made based on the potential value of losses and gains rather than the final outcome, and that people evaluate these potential outcomes using a certain heuristic that includes utility.

Utility-based recommenders are particularly effective because they integrate these decision-making models to assist consumers in efficiently and effectively locating the most suitable products. By predicting the utility values of items based on consumer preferences and presenting those with the highest values, these systems enhance the shopping experience, reducing the cognitive load and choice overload often experienced by consumers in rich information environments.

The concept of utility in recommender systems is borrowed from microeconomic theory, where the utility function is used to represent a consumer's preference ordering over a set of alternatives. In the context of a recommender system, a utility function U can be defined for a user u and an item i as $U(u,i)$. This function typically depends on various factors that contribute to an item's overall utility to the user, such as cost, quality, relevance, and personal preference indicators. The simplest form of a utility function might be linear, represented as:

$U(u,i) = w1 \cdot f1(i) + w_2 \cdot f_2(i) + \cdots + w_n \cdot f_n(i)$, where, $f_1(i), f_2(i), \ldots, f_n(i)$ are features of the item i, and $w1, w2, \ldots, w_n$ are weights that represent the importance of these features to user u. This linear approach assumes that the different attributes contribute independently to the utility, which simplifies computation but may not capture interactions between attributes. Utility-based systems can be categorized based on the nature of the utility function and the type of

data they use.

1.2.2.1 Single-attribute utility (SAU)

SAU systems are a specific type of utility-based recommender system that focus primarily on one key attribute of an item to determine its utility to a user. This approach simplifies the utility calculation by assuming that one predominant feature overwhelmingly influences the user's decision-making process. In SAU systems, the entire recommendation process revolves around optimizing or maximizing one particular attribute that is considered most critical from the user's perspective. For example, in a travel booking recommender system, the single attribute could be the price of tickets. The system would then recommend the options that offer the lowest prices, assuming that cost is the user's primary concern.

The utility U(p) of a product p is typically calculated as the weighted sum of individual attribute utility functions, according to the formula:

$$U(p) = \sum_{j=1}^{n} v_j f_j(y_{jp}) \qquad 1.9$$

Here, n represents the total number of attributes considered for each product. The coefficient (v_j) is the weight assigned to the jth attribute, reflecting its importance to the consumer's overall satisfaction, with the condition that the sum of all weights equals one, i.e., $(\sum_{j=1}^{n} v_j = 1)$. The term y_{jp} denotes the level or value of the jth attribute for product p, and $f_j(y_{jp})$ is the utility function for this attribute, calculating how much utility the specific attribute level contributes to the overall utility of the product. This model allows for a detailed assessment of product suitability by incorporating the diverse preferences and priorities of the consumer.

1.2.2.2 Muli-attribute utility (MAU)

MAU systems are a sophisticated class of utility-based recommender systems that evaluate items based on multiple attributes to calculate an overall utility score for each item. These systems are designed to consider a range of factors or characteristics that contribute to an item’s desirability or usefulness from the perspective of the user. Multi-attribute utility systems use a utility function that aggregates the utilities derived from multiple attributes

into a single score. This approach is based on Multi-Attribute Utility Theory (MAUT) [37], which is widely used in decision-making processes where decisions depend on multiple criteria.

The utility of a product p in a multi-attribute system is often expressed as:

$$U(p) = \sum_{i=1}^{m} w_i \cdot u_i(x_{ip}) \qquad 1.10$$

Where, m is the number of attributes consider, w_i represents the weight assigned to the i^{th} attribute, reflecting its importance relative to other attributes. These weights are normalized so that $\sum_{i=1}^{m} w_i = 1$. x_{ip} is the level of the i^{th} attribute for product p, $u_i(x_{ip})$ is the utility function for the i^{th} attribute, which calculates how much the specific level of this attribute contributes to the overall utility of the product. Utility functions in multi-attribute systems can be Linear, where the contribution of each attribute is directly proportional to its level. Or non-linear, including more complex relationships where the utility of one attribute may depend on the levels of others, or where diminishing returns and thresholds apply.

while utility-based recommender systems offer a highly personalized approach to making recommendations, they face several challenges. The complexity of designing an effective utility function cannot be underestimated, as it requires a deep understanding of user preferences and the impact of various attributes on perceived utility. These systems also demand extensive data on both user preferences and item attributes, which can be challenging to collect and maintain accurately. Moreover, user preferences can change over time, requiring the utility functions to adapt accordingly to remain relevant. This dynamic adaptation, along with the need to scale computations for large numbers of users and items, presents significant technical hurdles. There is also the risk of introducing bias into the system, either through the data used to model preferences or the methods by which attributes are weighted and evaluated. Balancing the model to avoid overfitting and underfitting is crucial—overfitting makes the system overly sensitive to minor details, while underfitting might result in ignoring important nuances in user preferences.

1.2.3 Session-based recommender systems (SBRS)

Session-based recommender systems are designed to predict the next action of a user within a specific session, particularly useful in situations where long-term user history may not be available or relevant [38]. These systems are adept at adjusting recommendations based on the user's short-term intentions observed during an active session. They primarily utilize two main types of data: first, the sequence of interactions recorded with an individual user since the beginning of the session, and second, information about past sessions from other users. This approach is especially valuable in domains like e-commerce, where a consumer's shopping goals might vary significantly from one session to another, often involving different product categories. Moreover, session-based systems are crucial when dealing with new or anonymous users, who may have no past recorded preferences or interests. Under these circumstances, the system must effectively generate useful recommendations with minimal information about the user. This method exemplifies the challenge of making relevant recommendations in session-based systems where user data is sparse.

Session-based recommendation techniques were initially proposed in the early 2000s, as indicated by references [39, 40]. However, despite the significant practical relevance of this area, research remained relatively limited, especially compared to the extensive body of work focusing on recommendations based on long-term user preferences. Until 2015, contributions to session-based recommendation were primarily published in popular fields such as e-commerce, music, and news, with citations like [41, 42, 43, 44], and occasionally for specialized uses like adaptive modeling support for machine learning workflows [45]. The landscape began to change in 2015 when the ACM Conference on Recommender Systems highlighted session-based recommendation challenges, introducing a large dataset of anonymous click logs. This release, combined with the burgeoning interest in applying deep learning to recommender systems, sparked a surge in research and development in session-based recommendation approaches. This shift has led to a growing interest in exploring and enhancing these techniques, reflecting their increasing importance in the field of recommender systems. Session-based recommendation problems can be broadly classified into two primary categories: Model-free approaches and Model-based approaches.

1.2.3.1 Model-free approaches

Model-free approaches utilize data-mining techniques to analyze session data without relying on complex user or session models. These methods primarily concentrate on identifying frequent patterns, both ordered and unordered, from past data [46, 47]. They also employ nearest neighbor algorithms to draw inferences from the similarities between current and historical user sessions. Appreciated for their simplicity, ease of implementation, and generally lower resource requirements, model-free approaches offer a practical alternative to model-based systems.

Instead of complex mathematical modeling, these methods use well-optimized data mining procedures to detect and utilize patterns within session data, making them less demanding on resources and simpler to analyze and debug. Recommendations generated by techniques like frequent-pattern and rule mining are easily interpretable based on the extracted patterns, increasing both transparency and user understanding. The inherent simplicity of these methods also minimizes the risk of overfitting, especially valuable in scenarios with limited training data. Remarkably, despite their straightforward nature, model-free strategies, particularly those using nearest neighbor techniques, can achieve performance that is competitive with, or even superior to, more complex model-based techniques. This blend of simplicity, efficiency, and effectiveness renders model-free approaches an attractive option for various applications within recommendation systems.

We can categorize model-free approaches into two distinct methods:

- Frequent Pattern Mining (FPM): FPM aims to discover items that appear together frequently in a dataset. The patterns identified can be used to predict behavior or preferences based on historical data [48]. The basic concept is to generate candidate item sets (either single items or combinations of items) and then test these item sets in the database to determine if they meet a predefined minimum support threshold. These techniques extract patterns that indicate the likelihood of subsequent actions a user might take during a session, based on their previous activities. This capability makes FPM a valuable tool in SBRS, where the goal is to enhance the user's current interaction by predicting and recommending items based on their ongoing behavior.

- In the simplest form of SBRS, FPM is used to determine pairwise item co-occurrence frequencies within past sessions. At this, recommendations are generated by identifying items that have historically been associated with those currently being considered by the user in their session. This method of leveraging Association Rules [49], specifically those that are size two, simplifies the mining process but can be very effective in increasing the relevance of recommendations. Association rules are a key aspect of FPM, designed to detect items or actions that commonly occur together within the same session, regardless of their order. While basic FPM-based algorithms don't specifically target session-based analysis, they lay the groundwork for more advanced recommendation techniques. For a more nuanced approach, Sequential Pattern Mining can be applied, which not only considers the co-occurrence of items but also takes into account the order in which they appear. This approach can be further refined through Contiguous Sequential Patterns, which demand that the related items appear consecutively in the sequence of actions within a session.

- Nearest Neighbor: The Nearest Neighbor approach in session-based recommender systems strategically identifies and leverages similarities between current and past user sessions to generate recommendations [50]. This method is highly effective in environments where user interactions during a session such as product views, cart additions, or content engagement can indicate user intent. There are two main variants of nearest-neighbor methods in session-based recommendation systems: item-based and session-based. Item-based methods, as outlined in [50], determine item similarity based on their frequency of co-occurrence in training session data. On the other hand, session-based methods expand on this by evaluating the entire user session against historical sessions from the training data, not just the last item viewed. A thorough discussion of these model-free approaches is available in [51]. Initially, each session is represented as a set or sequence of actions taken by the user. To determine the similarity between the current session and historical sessions, various metrics can be employed such as, Jaccard or cosine similarity. After identifying which past sessions are most similar to the current one using these

metrics, the Nearest Neighbor method aggregates items from these similar sessions. The aggregation focuses on items that frequently appear across similar sessions but are not present in the current session.

1.2.3.2 Model-based approaches

Model-based approaches in recommender systems utilize sophisticated machine learning algorithms to build predictive models from user and item data, enabling personalized recommendations. In contrast to model-free approaches, which assess interactions or patterns directly without employing statistical models, model-based methods involve developing algorithms that capture user preferences and behavior patterns. These algorithms are subsequently employed to forecast user ratings or preferences for items with which they have not yet interacted. In the realm of session-based recommender systems (SBRS), recent developments in model-based strategies primarily involve sequence learning techniques. These techniques focus on constructing models from historical sequences of user actions to predict subsequent behaviors, as noted in [52]. Advanced sequence learning techniques are particularly adept at modeling the intricate relationships between user actions within and across sessions, providing a deeper understanding of user interaction dynamics.

- Markov Models play a crucial role in the realm of model-based approaches within recommender systems, especially effective for tasks involving sequence prediction such as session-based recommendations [51]. These models are based on the Markov property, which posits that the likelihood of moving to a subsequent state (or action) is solely dependent on the present state, rather than the entire history of previous states. This feature renders them highly adept at modeling user interactions within a session, determining the likely next product a user might explore after the current one. Functionally, Markov Models employ a transition matrix that captures the probabilities of transitioning from one item to the next, with these probabilities derived from historical user interaction data. Although this matrix can become quite extensive, it serves as the core of the model, enabling the prediction of a user's subsequent actions based on their present actions. For instance, if a user is currently watching a comedy movie, the model might predict the next view to also be a

comedy, based on patterns in the data that indicate users frequently watch similar genres in succession. While standard Markov Models focus only on the immediate preceding state, more advanced, higher-order versions consider multiple prior states to better capture complex user behaviors and patterns. However, increasing the model's order also amplifies its complexity and computational requirements. Nevertheless, the straightforwardness and operational efficiency of Markov Models make them particularly appealing; they can swiftly generate predictions, ideal for settings that necessitate immediate recommendation capabilities.

- Deep learning models are fundamentally transforming model-based approaches in session-based recommender systems (SBRS), employing advanced machine learning techniques to analyze and predict user behavior with high precision. These innovative models provide a deep insight into the intricate dynamics of user actions within individual sessions and across multiple sessions, substantially improving the personalization of recommendations. For instance, Neural Collaborative Filtering (NCF) revitalizes traditional collaborative filtering techniques by integrating neural networks. This approach excels at modeling complex, non-linear interactions between users and items—a critical capability in SBRS where nuanced user behaviors significantly impact the effectiveness of recommendations. NCF's ability to capture these detailed interactions allows for the dynamic tailoring of recommendations to each session's specific context, thereby boosting user engagement and satisfaction [53]. Recurrent Neural Networks (RNNs) are invaluable in scenarios where the sequence of interactions is crucial. Excelling in processing data sequences, these networks are perfect for SBRS that need to adapt recommendations as user preferences evolve throughout a session. RNNs and LSTMs continuously update the recommendation strategy in real-time, maintaining relevance from the beginning to the end of the session [54]. Convolutional Neural Networks (CNNs) play a significant role in SBRS by analyzing visual content related to items, like images and videos. They recommend items based on visual similarities or content features, adding a unique layer of personalization beyond mere transactional data. This visual processing ensures that recommendations align

more closely with users' current visual preferences, enhancing the overall user experience [55]. Transformers and Self-Attention Mechanisms, adapted from their breakthroughs in natural language processing, effectively manage sequences of user interactions in SBRS. Utilizing self-attention mechanisms, these models evaluate the importance of each item in a sequence, determining the most relevant recommendations based on the context of prior interactions. This method ensures that each recommendation is contextually appropriate for the ongoing session, optimizing user satisfaction [56].

Session-based Recommender Systems (SBRS) face several challenges and limitations despite their utility in providing immediate, context-specific recommendations. The cold start problem is significant for new items lacking interaction history, hindering the system's ability to offer diverse and novel recommendations. Maintaining session continuity can also be challenging, especially when users return after a break, which may disrupt the relevance of recommendations. The effectiveness of SBRS heavily relies on the quality of interaction within each session; sparse interactions lead to poor recommendation quality. Furthermore, the implementation of complex models like deep learning (e.g., RNNs, LSTMs) necessary to process intricate session data is computationally intensive and requires substantial resources, posing a barrier for resource-limited environments. Additionally, sessions may include noise or outlier actions that do not accurately reflect user preferences, complicating the accuracy of the system's predictions. Scalability also becomes an issue as user and item numbers grow, necessitating advanced scaling strategies to maintain performance without compromising recommendation quality. Privacy concerns are paramount too, especially given the reliance on anonymized session data, requiring stringent compliance with data protection laws like GDPR. Lastly, the dynamic nature of user preferences, which can change significantly within and across sessions, requires SBRS to continuously adapt in real-time, a demanding task without long-term historical data. Addressing these challenges involves enhancing data processing capabilities, refining algorithms, and implementing sophisticated data handling and privacy measures, which are crucial for advancing the effectiveness and user-friendliness of SBRS.

1.2.4 People-to-people reciprocal recommenders

People-to-people reciprocal recommenders are systems designed to facilitate interactions between users where both parties benefit from the connection [57]. This kind of recommender system is particularly useful in contexts where the success of the interaction depends on the mutual interest or compatibility of both users. This process involves identifying potential friends, professional contacts, and communities on social networks; matching individuals on dating sites; connecting job seekers with potential employers; and pairing mentors with mentees or students in educational online platforms. Networks like Facebook and LinkedIn support multi-directional connections, whereas platforms such as dating sites typically facilitate one-on-one relationships. These recommendations often require reciprocal connections, where both parties share their preferences and both must find the match satisfactory. For instance, in job recruitment, both the employer and the candidate must mutually assess suitability. In online dating, both parties must show a mutual interest to form a successful relationship. Similarly, matching students in educational settings often requires a reciprocal approach to enhance learning benefits. The critical role of reciprocity in person-to-person recommendation systems has only recently been acknowledged. In this discussion, we delve into the distinct aspects of reciprocal recommenders, review existing studies, and offer a case study on an online dating scenario.

1.2.4.1 Social Networking and Dating Apps

Social Networking and Dating Apps: On platforms like dating apps and social networks, reciprocal recommendation systems play a crucial role in connecting users. These systems match individuals based on shared interests, geographical proximity, demographic data, and other compatibility factors. The objective in dating apps is to recommend potential partners who are likely to find mutual interest, thereby enhancing the likelihood of successful engagements. In social networks, the aim is also to foster connections that encourage interaction and community-building among users with similar interests or backgrounds, further enriching the user experience.

For instance, Chen et al. [58] explored the efficiency of various people recommendation algorithms through a detailed study of 3,000 users on a social networking site called Beehive.

Their research aimed to enhance users' abilities to reconnect with known offline contacts and forge new friendships online. They evaluated four distinct algorithms and found that each effectively broadened users' social circles. The study revealed that algorithms drawing on social network data were better received and more successful in identifying known contacts, while those relying on content similarity were more adept at discovering new friends. Additionally, they gathered qualitative feedback, which offered valuable insights for refining future recommender systems. The research categorized the algorithms into two groups: those that utilize social relationship information (FoF and SONAR) and those that focus on content similarity (Content and CplusL). It was observed that relationship-based algorithms outperformed those based on content similarity in terms of user engagement, possibly due to the more comprehensive relationship data available within specific environments such as IBM, which might not be as accessible elsewhere. The findings suggest a hybrid approach to optimize the effectiveness of recommender systems on social platforms. Initially employing relationship-based algorithms can swiftly build a user's network by linking them with known contacts, thereby boosting trust in the system. As the network expands, incorporating content similarity algorithms can sustain the network's relevance by continually introducing new and intriguing contacts. This strategy not only capitalizes on the strengths of both algorithm types but also aligns with the user trust dynamics highlighted in the study, suggesting a balanced and dynamic approach to social networking recommendations.

Fazel-Zarandi et al. [59] introduced a novel approach to expert recommendation within organizations and scientific communities, focusing on personalizing expert selection. The system leverages the Multi-Theoretical Multi-Level (MTML) framework to incorporate diverse social science theories, effectively mapping user motivations to specific domain data and recommendation strategies. This approach considers multiple complex networks, such as collaboration and citation networks, to tailor recommendations to individual user needs. The prototype system developed combines social network analysis and semantic web technologies to handle data from heterogeneous sources effectively. A preliminary evaluation using offline methods showed promising results in predicting scientific collaborations, demonstrating the system’s potential for substantial personalization and flexibility in recommendation processes.

Brozovsky and Petricek [60] presented a study on the application of recommender systems in online dating, highlighting the challenge of information overload on these platforms, especially with multimedia profiles. They developed and tested a system using two collaborative filtering (CF) algorithms and two global algorithms. Their findings demonstrate that CF recommenders outperform the global algorithms commonly used by dating sites, both in terms of user satisfaction and prediction accuracy. Notably, a blind test with actual users confirmed a preference for CF-based recommendations over those based on global popularity. The research underscores the potential of recommender systems to enhance the value and monetization of online dating services. They also noted that while CF algorithms such as User-User and Item-Item significantly outperformed global popularity benchmarks

Pizzato et al. [61] introduced an innovative recommendation system for online dating, which not only aligns potential partners based on mutual interests but also on reciprocal attractiveness, using a dataset from a leading online dating platform in China. This system employs novel similarity metrics that analyze user interactions within the dating network, specifically looking at interest similarity when users message the same individuals, and attractiveness similarity when they receive messages from the same users. A reciprocal score is then generated to determine the compatibility between a user and potential matches. Their analysis indicates that the collaborative filtering-based algorithms employed by this system significantly outshine earlier approaches and content-based algorithms in both precision and recall metrics. The research further uncovers distinct behavioral differences in the way male and female users approach the search for potential partners, with males typically prioritizing their interests and females focusing more on their perceived attractiveness to others.

Akehurst et al. [62] introduced the Content-Collaborative Reciprocal (CCR) recommender system for an online dating website, using a substantial dataset from a well-known platform. Initial analysis revealed that individuals with similar profile attributes often share preferences and receive similar responses from others. This observation established the foundation for CCR, a hybrid model that merges content-based and collaborative filtering techniques. The content-based component of the system identifies users with analogous profile traits, while the collaborative part utilizes interaction data, such as likes and dislikes, to generate

reciprocal recommendations. CCR is particularly beneficial for new users because it solves the cold start problem by immediately providing recommendations based on user profiles.

1.2.4.2 Mentoring Platforms

These type recommender systems designed to facilitate optimal pairings between mentees and mentors. The goal is to ensure that both parties benefit significantly from the relationship. For example, a mentee who is actively seeking expertise in a specific area, such as digital marketing, software development, or even academic research can be matched with a mentor who not only has established expertise in that field but is also keen to pass on their knowledge. This mentor is typically someone who values the opportunity to guide enthusiastic learners, helping to foster their development in the subject area. Conversely, mentors can benefit from the fresh perspectives and potential collaboration opportunities that mentees bring to the table, creating a dynamic and mutually rewarding exchange.

For example, in their research, Bull et al. [63] highlight the crucial role of user modeling in improving the I-Help communication system designed for learners. The system tailors its services by intricately modeling various user characteristics such as interests, goals, and cognitive styles. This modeling is executed from the perspectives of individual users, their peers, and system agents, leading to a highly dynamic yet segmented approach that complements the distributed, agent-based structure of the system. Such a framework facilitates real-time, targeted user interactions for tasks like selecting a helper, negotiating help sessions, or retrieving information. Looking ahead, I-Help plans to enhance user engagement by proactively anticipating user needs and expanding its implementation in both educational and corporate environments, thus improving learning effectiveness and overall system functionality.

Li [64] introduced a pioneering approach to matching mentors and mentees on the Codementor platform, focusing on optimizing mentorship pairings in live mentoring environments. This study introduces two key predictive tasks: Mentor Willingness Prediction (MWP) and Mentee Acceptance Prediction (MAP). MWP assesses a mentor's readiness to address specific mentee requests, while MAP evaluates the likelihood of a mentee accepting a given mentor. The proposed method recommends mentors in a ranked order, aimed at

maximizing both the mentor's willingness to participate and the mentee's likelihood of acceptance. The model uses four critical features—availability, capability, activity, and proximity—to predict successful pairings effectively. These features were validated using various supervised learning techniques, significantly enhancing the quality of mentorship.

1.2.4.3 Job recommendations

In the context of job recommendations within people-to-people recommender systems, the goal is to create beneficial matches between job seekers and employers. This type of system extends beyond just suggesting jobs to potential candidates; it also involves recommending potential candidates to employers, ensuring a reciprocal benefit. This not only improves the hiring process but also enhances job satisfaction and retention rates, making it a critical tool in the human resources technology landscape.

For instance, Malinowski et al. [65] introduced an innovative approach by applying recommendation systems to the recruitment process, which has traditionally been overlooked by such technologies. They developed a methodology that leverages dual recommendation systems to better align the preferences of both job seekers and recruiters, emphasizing the importance of a mutual selection process in job placements. Through initial tests conducted with data from a student experiment, their results demonstrated the system's capability to improve the accuracy of matching candidates with job vacancies. This approach not only highlights the potential of personalized recommendation systems in human resource settings but also paves the way for future applications in enhancing team dynamics and interpersonal relations.

Hong et al. [66] conducted a detailed study and implementation of a cutting-edge online recruitment system named iHR, tailored for the Xiamen Talent Service Center. The system incorporates the latest advancements in data mining and recommendation technologies, designed specifically to synchronize the interests of job seekers and businesses effectively. Their research involved a thorough analysis of existing online recruiting platforms, from which they extracted key functionalities that contribute to a mutually beneficial recruitment environment. The iHR system was meticulously developed and has been successfully operational, attracting a substantial user base of over 1.77 million, including 1.7 million job

seekers and 70,000 enterprises, demonstrating its profound influence on Xiamen's job market and broader economic growth. The architecture of iHR integrates a complex algorithm for constructing user profiles by synthesizing data from diverse sources, alongside a dynamic recommendation engine that finely balances the demands and preferences of both job seekers and employers. Despite the high operational and maintenance costs, and the challenges in assessing each component's impact in a live setting, the iHR system stands out as a pioneering and influential entity within the realm of online recruiting in China.

Despite their potential, people-to-people reciprocal recommender systems face numerous challenges that complicate their implementation and efficacy. Accurately matching individuals based on mutual interests involves a delicate balance of diverse and nuanced preferences, adding significant complexity. These systems also deal with major concerns around data privacy and security, as they require sensitive personal information to produce accurate recommendations, thus increasing vulnerability to breaches. Like all AI-driven platforms, they are susceptible to perpetuating biases present in the training data, which can lead to unfair matchmaking that favors certain user groups. Scalability is another critical issue; the computational complexity of finding suitable matches escalates as the user base expands, posing a daunting technical challenge. The success of these systems largely depends on user honesty, yet users might manipulate their profiles to appear more attractive, compromising the quality of matches. Furthermore, dynamic changes in user preferences can make the system less accurate unless it quickly adapts. Evaluating the effectiveness of these systems is complex since success must be assessed based on the mutual satisfaction of both parties, not just individual engagement or satisfaction. New platforms often encounter a 'cold start' problem, where insufficient initial users hinder effective recommendations. Lastly, the interdependency of recommendations, where the outcome for one user impacts another, creates a dynamic that is difficult to manage and optimize, emphasizing the continuous need for research and development to refine these systems.

1.2.5 Community-Based recommender systems

Community-Based recommender systems harness the collective preferences and activities of a community to deliver tailored recommendations to its members. This strategy proves highly

effective in settings where users actively engage with one another, exchange information, or participate in social networking. Research indicates that individuals tend to trust recommendations from friends more than those from anonymous yet similar users. By analyzing the preferences and behaviors of a person's social circle or community, these systems can suggest items or content that have received approval from like-minded members within that group. This method not only taps into the collective intelligence of the community but also increases the relevance of the recommendations, ensuring they are more customized and meaningful for each user through their social ties and common interests within the community. Numerous studies have been conducted on the topic of Community-Based recommender systems.

For example, Chen et al. [67] presented an in-depth analysis of the business models utilized by community-based recommender systems, emphasizing their profitability potential through the use of user-generated content. The study develops a framework based on existing theories to assess the business value of these systems from the viewpoint of system operators. This framework delineates five distinct roles for system operators, each associated with specific business models, and outlines the primary sources of revenue and operational efficiencies. They also provide a classification of these business models to assist future practitioners like investors and managers in evaluating the value propositions of community-based recommender systems. Although the study is exploratory and not exhaustive, it serves as a preliminary foundation for further investigation into the business models of recommender system providers. The authors acknowledge limitations, such as the lack of empirical testing of the framework and full validation of all used constructs, noting that further empirical research could enhance the robustness of their conclusions.

Vatsavayi [68] introduced a community-based content recommender system aimed at boosting the accuracy and speed of recommendation services. This method integrates a static community detection technique into the recommender system framework. The process begins by constructing a feature matrix, where each column represents a movie genre and binary values indicate the genre's presence in each movie. User profiles are then formed through a dot product of the feature matrix with a rating matrix. Additionally, network interactions from

genre co-occurrences are examined using a community detection algorithm to identify tight-knit substructures that share similar features. Recommendations are subsequently made based on the semantic genre features of communities among active users. This approach has shown considerable improvements in accuracy over conventional recommendation methods and suggests potential for future exploration into integrating low-level visual features that extend beyond standard expert annotation methods. Deebak and Al-Turjman [69] introduced a novel Community-Based Trust Aware Recommender System (CB-TARS) tailored for IoT-enabled big-data applications in smart sustainable cities. This system refines traditional collaborative filtering by incorporating trusted neighbors to mitigate the risks associated with deceptive ratings. CB-TARS calculates user preferences using a weighted average from trusted neighbors' ratings within the community, enhancing the accuracy of recommendations. The system assesses the reliability of ratings based on the confidence drawn from the number of ratings and the proportion of positive to negative feedback. Furthermore, they developed a new similarity metric that integrates this confidence measure into the standard Pearson correlation used in big-data cloud services. The efficacy of CB-TARS was tested using three real-world datasets and evaluated based on metrics such as Mean Absolute Error (MAE), Recommendation Coverage (RC), Root Mean Square Error (RMSE), and F-Measure. The findings confirmed that CB-TARS significantly surpasses conventional methods in terms of accuracy, coverage, and overall performance.

Community-Based recommender systems, which leverage collective user behaviors and preferences for personalized recommendations, face several significant challenges. Key among these is scalability, as managing large volumes of data from growing user communities can lead to performance bottlenecks. Additionally, these systems often struggle with the cold start problem, where new users or items lack sufficient interaction data to generate reliable recommendations. Privacy concerns are also paramount, given the need to access detailed user interaction and social connection data, raising issues about user consent and data protection. Integrating diverse data sources adds complexity, while bias and fairness issues may arise from reinforced popularity biases and echo chambers, potentially limiting exposure to diverse viewpoints. Ensuring user engagement is crucial, requiring strategies to maintain active participation, which is critical for the system's effectiveness. Furthermore,

algorithmic transparency is necessary to build trust, necessitating clear communication about how recommendations are generated and allowing user control over their data. Finally, developing appropriate evaluation metrics that reflect community influence and user satisfaction is essential for accurately assessing the performance of these systems.

1.3 Advanced recommender systems

Advanced recommender systems are at the forefront of recommendation technology, leveraging cutting-edge machine learning methods and intricate data models to enhance the accuracy, relevancy, and personalization of their suggestions. These systems are specifically engineered to address the complexities arising from enormous data volumes, varied user preferences, and intricate item features within contemporary digital contexts. Among these advanced systems, deep learning-based and graph-based approaches are particularly noteworthy for their capabilities in handling complex and large-scale data.

1.3.1 Deep learning-based approaches

Deep learning employs a hierarchical structure of neural networks for machine learning tasks. These networks are comprised of units called neurons, which are modeled after biological neurons. A typical deep neural network features several layers, each containing multiple neurons. Connections exist between neurons in successive layers. Signals are accumulated from one layer to the next, and the transmission of information across these layers is regulated by activation functions [70]. These functions, including ReLU, Sigmoid, and Tanh, allow the network to handle data in a nonlinear manner, distinguishing deep learning from traditional linear approaches.

1.3.1.1 Multi-Layer Perceptrons

A Multilayer Perceptron (MLP) is a type of artificial neural network structured with an input layer, several hidden layers, and an output layer. The input layer captures and forwards signals to the hidden layers, where each neuron applies a nonlinear activation function to progressively refine inputs into more sophisticated representations. This architecture enables MLPs to serve as a universal approximation algorithm, theoretically capable of emulating

any continuous function accurately, provided it is equipped with enough neurons. MLPs utilize backpropagation for training, a supervised learning technique that involves forwarding inputs through the network to produce an output, evaluating the error by comparing this output against the expected results, and then reversing the direction of data flow to adjust the weights and reduce errors. Activation functions such as sigmoid, tanh, and ReLU (Rectified Linear Unit) are integral to this process, introducing the necessary non-linearity that allows the network to capture and model complex patterns.

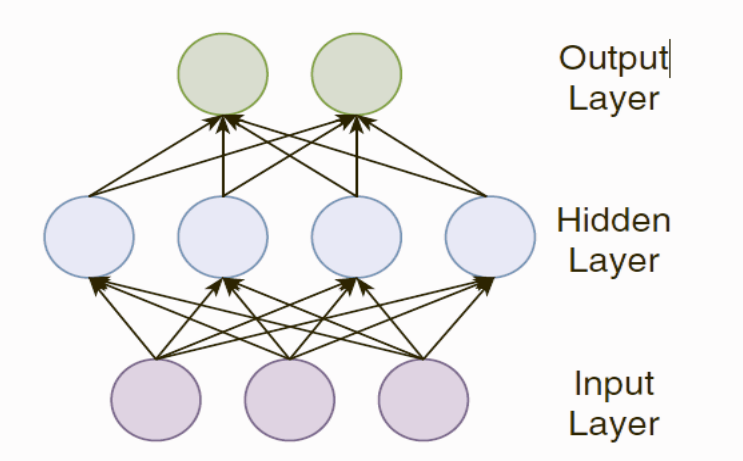

Figure 1.2: A Multilayer Perceptron [71].

Figure 1.2 illustrates an example of a Multilayer Perceptron (MLP). The input layer receives the data and passes it on to the subsequent hidden layers. Each layer consists of neurons that apply nonlinear activation functions to transform the data progressively. These transformations allow the MLP to extract and learn increasingly abstract features of the data. After processing through one or more hidden layers, the data reaches the output layer. Depending on the specific task, this final layer outputs a single value for regression problems or multiple values for classification tasks. This hierarchical structure and processing capability enable the MLP to perform complex computational tasks effectively. Last year, numerous studies employed Multilayer Perceptrons (MLPs) to develop sophisticated recommender systems, tackling various challenges such as the cold-start problem for new users and issues concerning existing users, as noted in reference [72]. Additionally, MLPs were utilized to derive latent factors for items and users from the user-item interaction matrix,

as detailed in reference [73]. This application of MLPs underscores their versatility and effectiveness in enhancing the capabilities of recommender systems.

1.3.1.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are increasingly being utilized in recommendation systems beyond their traditional domain of image processing, especially in areas where visual content or structured data plays a significant role [74, 75]. In e-commerce and multimedia services, CNNs are adept at extracting features from product images to recommend items with similar visual attributes or to identify stylistic trends, enhancing the system's ability to match user preferences with visual characteristics [76]. Additionally, CNNs can process sequences of user interactions using 1D convolutional layers, capturing local dependencies within time-series data or user activity sequences, which is valuable for understanding user behavior over time. They can also treat structured non-image data—like user demographics or interaction histories—as input channels, drawing meaningful patterns similar to spatial hierarchies in images. Often, CNNs are integrated with other neural network architectures, such as MLPs or RNNs, in hybrid models that combine spatial feature extraction with sequential or abstract pattern recognition [78]. This approach not only enhances traditional collaborative filtering techniques by providing deep, nuanced feature representations of items and users but also uncovers subtle interactions that might be missed by standard matrix factorization methods. For example, in a movie recommendation system, CNNs could analyze the visual styles of movie posters while collaborative filtering algorithms assess user rating patterns, together delivering more personalized and precise recommendations by leveraging both visual and behavioral data insights.

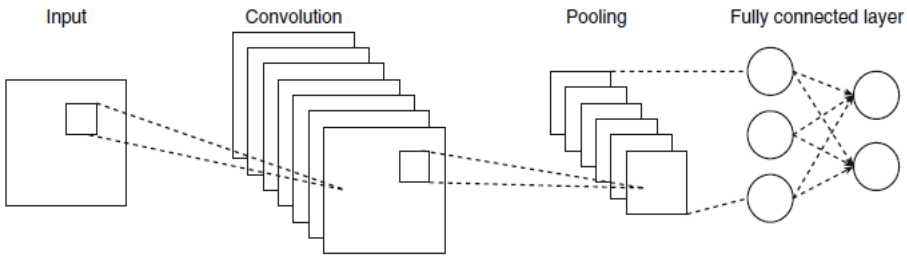

Figure 1.3: CNNs illustration [79].

As depicted in Figure 1.3, the Convolutional Neural Network (CNN) processes input data

through several key stages. Initially, the data enters the convolutional layer where it is convolved with various filters, generating a set of feature maps that highlight essential attributes like edges and textures. These feature maps then pass through a pooling layer, which reduces their spatial dimensions through operations such as max pooling, effectively summarizing the most significant features while minimizing data size and computational demands. Following pooling, the reduced feature maps are flattened into a single vector, preparing them for the fully connected layer. In this final stage, the flattened vector undergoes processing by neurons that are interconnected, facilitating the classification of the data into categories based on the learned features.

1.3.1.3 Autoencoders

Encoder-decoder architectures have become foundational in advanced recommender systems, adeptly processing a wide range of data through a dual-stage approach: first, encoding the input data into a compact, contextual representation, and then decoding this information to output specific recommendations or predictions about user behavior. Recent studies highlight the effectiveness of CNNs for analyzing spatial data and Transformers for handling sequences, which are crucial for addressing long-range dependencies and enhancing model scalability [80, 81]. These architectures are widely applied in scenarios such as personalized content recommendations, where platforms like Netflix predict what users might watch next based on their viewing history, or in e-commerce, where predictive models suggest products based on complex patterns of user-item interactions [82, 83]. The versatility and scalability of encoder-decoder models make them indispensable in modern recommender systems, facilitating more accurate and contextually relevant suggestions across various sectors.

An autoencoder is a type of unsupervised learning model that comprises three main components: an encoder, a decoder, and a bottleneck layer, detailed in Figure 1.4. The encoder part of the autoencoder compresses the input data into a smaller, dense representation in the bottleneck layer, which captures the essential information about the data. This compressed representation is then used by the decoder to reconstruct the original input data as accurately as possible. The aim is to force the autoencoder to prioritize the most salient

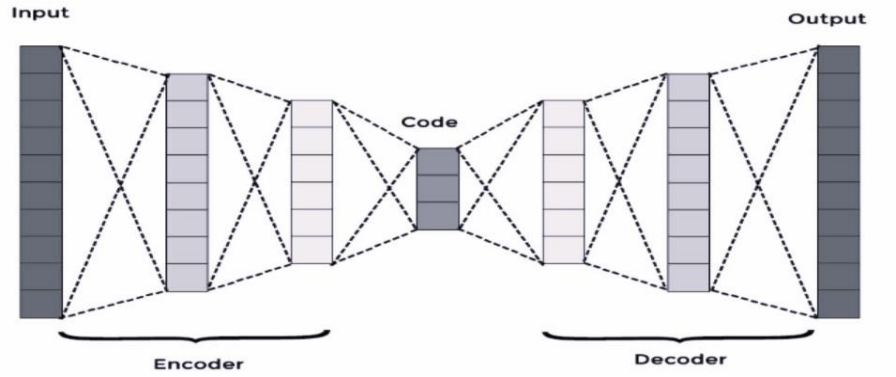

Figure 1.4: Autoencoder architecture [84].

attributes of the input data, thus learning efficient and often insightful representations. This mechanism not only helps in dimensionality reduction but also aids in noise reduction, feature extraction, and even anomaly detection in complex datasets. The process enhances the model's ability to capture and represent the underlying structure of the input data, making it a powerful tool for various applications in data analysis and machine learning.

Deep learning-based recommender systems offer significant capabilities but also face multiple limitations that can impact their practical deployment and effectiveness. These systems are resource-intensive, requiring significant computational power and thereby increasing costs and limiting accessibility, especially for smaller entities or those with constrained resources. Additionally, they depend heavily on large datasets to function optimally and tend to underperform in scenarios with sparse or poor-quality data, such as with new users or during the cold start phase. The inherent complexity of these models can lead to challenges in fine-tuning and a tendency to overfit, meaning they might excel with training data but fail to generalize to new, unseen data. The lack of transparency in how these models generate recommendations poses issues in industries that require clear audit trails and explanations of decision-making processes. Furthermore, these systems can unknowingly perpetuate biases present in the training data, resulting in recommendations that may be unfair or unbalanced. They are also less flexible in adapting to changes in user behavior or updates in the item catalog without extensive retraining or regular updates, potentially leading to stale recommendations. Vulnerability to adversarial attacks, where minor tweaks to inputs can mislead the system into making incorrect recommendations, presents serious security risks. These challenges highlight the crucial need for continuous advancements in

improving the robustness, interpretability, fairness, and security of deep learning-based recommender systems.

1.3.2 Graph-based recommender systems

Graph-based recommender systems (GBRS) utilize graph theory to significantly enhance recommendation processes, presenting a robust alternative to the traditional matrix-based collaborative filtering methods [85]. These systems construct a graph where nodes symbolize entities like users and items, and edges represent the interactions or relationships among them, such as purchases, ratings, or social connections. Alongside these basic elements, the graph can also incorporate additional attributes for nodes and edges, including timestamps, or textual reviews to enrich the context. Within this framework, the system employs various graph theory algorithms to meticulously analyze the network's structure. This analysis involves tasks such as computing the shortest paths between nodes, detecting communities to identify groups of similar preferences, or determining centrality measures that highlight the most influential or central items and users in the graph. These sophisticated techniques allow the system to uncover complex, nuanced patterns and relationships within the data, providing a deeper understanding of user behavior and preferences which, in turn, enhances the quality and relevance of the recommendations provided.

1.3.2.1 Path-based similarity

Path-based similarity evaluates the degree of similarity between two nodes within a graph by analyzing the paths that connect them. This metric is especially useful in networks where the relationships between entities are not fully described by direct connections alone. Path-based similarity assesses the "closeness" of two nodes by examining the multiple paths that bridge them. The fundamental principle is that nodes are considered more similar if there are numerous, shorter, and more direct paths connecting them, suggesting a stronger relationship through various routes within the network.

Path-based similarity can be calculated using various methods that analyze the connections between two nodes within a graph. One common approach is the Shortest Path method, where the similarity between two nodes is inversely related to the length of the shortest path

connecting them; nodes linked by shorter paths are deemed more similar. Another method is the Katz Measure [86], which considers all paths between two nodes but assigns exponentially decreasing weights to longer paths, thereby prioritizing shorter connections. This measure is mathematically expressed as:

$$\text{Katz}(i,j) = \sum_{l=1}^{\infty} \beta^l \cdot \left|\text{paths}_{ij}^{l}\right| \qquad 1.11$$

Where, β is a damping factor that reduces the influence of longer paths, and $\left|\text{paths}_{ij}^{l}\right|$ represents the number of paths of length l between nodes i and j. This comprehensive approach allows for a more nuanced assessment of similarity by incorporating both the quantity and quality of the paths linking nodes, highlighting the depth of their connectivity within the network.

1.3.2.2 Random Walk Similarity

Random Walk Similarity measures the probability of moving from one node to another through a stochastic process, often across various steps or time periods. This method is particularly effective in large and complex networks with detailed and varied connectivity patterns, as it evaluates transitions between nodes to understand their underlying similarities. The basic idea behind Random Walk Similarity is to model how a random walker would move through the network starting from one node and the probability of reaching another node. The similarity between two nodes is determined by the probability distribution of these random walks, indicating how easily one node can be reached from another. Essentially, the more probable it is for a random walk to transition between two nodes, the more similar those nodes are considered to be.

Random Walk Similarity utilizes several calculation methods to assess node relationships in a graph. The Basic Random Walk method [86] initiates from a selected node and moves to adjacent nodes randomly, using uniform probabilities or those weighted by edge strengths, with similarity assessed by how frequently a node is visited from another over a set number of steps. Personalized PageRank [87] modifies this by introducing a restart probability at the source node, allowing the random walker to periodically return to the starting point, thereby

personalizing the similarity score for specific nodes. Hitting Time and Commute Time metrics calculate the expected number of steps to reach one node from another and back, respectively, with shorter times indicating greater similarity. Finally, Random Walk with Restart (RWR) [88] combines random steps with regular returns to the origin, capturing both direct connections and broader network structure in the similarity measure.

Graph-based recommender systems, while offering sophisticated solutions, like other type of recommenders encounter several challenges that impact their effectiveness. Scalability issues arise as the size of the graph increases, requiring extensive computational resources to manage large user and item sets efficiently. Sparsity of interactions within these large graphs can weaken prediction accuracy, compounded by the complexity involved in constructing and maintaining optimal graph structures. Like other systems, they face the cold start problem, where new users or items with minimal interaction data are difficult to integrate effectively. The dynamic nature of user and item data demands continuous updates to the graph structure, a process that can be computationally expensive and complex. Implementing and tuning advanced graph-based algorithms, particularly those involving graph neural networks, introduces algorithm complexity and potential overfitting, while the interpretability of the recommendations can be limited, making it hard to understand why specific recommendations were made. Additionally, graphs are sensitive to noisy or corrupt data, which can skew recommendations, and integrating diverse data sources to build these graphs raises significant privacy concerns. Addressing these challenges is crucial for advancing the capabilities and applicability of graph-based recommender systems in real-world scenarios.

1.3.3 Hybrid recommender systems

Hybrid recommender systems utilize a blend of various recommendation methods to enhance the quality of recommendations and address the shortcomings of any single technique. By merging different strategies, these systems capitalize on the strengths of each approach to offer more precise, varied, and resilient recommendations. As previously mentioned, all recommender systems have their limitations, and one effective way to mitigate these issues is by harnessing the strengths of multiple approaches through combination.

Recent research in hybrid recommender systems has led to significant advancements, enhancing both their efficacy and their ability to address inherent challenges. A notable trend is the integration of deep learning with traditional models such as collaborative filtering, which has improved the accuracy of recommendations by extracting complex features from extensive datasets [89]. There has also been a growing interest in context-aware recommendations, which consider factors like time and location to provide more relevant content [90]. To combat the cold start problem, some researchers have suggested incorporating user demographic information or item metadata into traditional algorithms, thus enhancing recommendations for new users or items with scant historical data [91]. Moreover, cross-domain recommendation strategies are being explored, aiming to use preferences from one domain to enrich recommendations in another, thus expanding user experiences [92]. Additionally, the development of hybrid systems that address fairness and bias is underway, with new models that balance recommendation accuracy with ethical considerations to mitigate bias and promote equity across diverse user groups [93]. With recommender systems growing more sophisticated, research is also focusing on their robustness against adversarial attacks by incorporating sturdy machine learning techniques into hybrid models [94]. These developments underscore a shift towards more adaptable, precise, and secure hybrid recommender systems across various fields.

1.4 Motivation for Context-Aware Systems

The majority of recommender systems operate primarily by analyzing patterns in user and item data, such as past purchases or ratings. While effective to a degree, this approach can be limited because it largely ignores other influential factors that can significantly shape a user's preferences and decisions. These factors include contextual elements like time (e.g., season, time of day), location (e.g., at home or traveling), social settings (e.g., alone or with friends), and even current emotional states or specific tasks at hand.

Incorporating these additional dimensions into the recommendation process can dramatically enhance the system's ability to offer timely and appropriate suggestions that are more aligned with the user's current needs and circumstances. For example, a music streaming service that

considers the user's current activity—such as working out or relaxing—could suggest playlists that enhance those particular experiences, rather than just relying on general listening habits. Consequently, the rating function evolves into a multidimensional function.

By broadening the scope beyond static user and item profiles to include dynamic contextual information, recommender systems can achieve a higher degree of personalization and responsiveness. This leads to improved user satisfaction as recommendations become more relevant and useful in real-time scenarios, potentially increasing engagement and loyalty to the service.

1.5 Conclusion

In this chapter, we offer a comprehensive review of various recommender systems, highlighting recent advancements, inherent advantages, and notable limitations. These systems are categorized into three distinct types: traditional recommender systems, which primarily rely on historical user-item interactions; specialized recommender systems, which are tailored for specific applications or industries; and advanced recommender systems, which incorporate cutting-edge technologies and methodologies. Additionally, we discuss the rationale for our focus on context-aware recommender systems (CARS), emphasizing their relevance and potential in enhancing recommendation accuracy.

In the next chapter, we will delve deeper into CARS, reviewing existing literature, and identifying the key challenges faced by researchers and developers in this field. This exploration will include a discussion of the methodologies employed, the contexts considered, and the impacts of these systems on user experience

Chapter 2

Context-Aware Recommender Systems

This chapter offers a comprehensive review of context-aware recommender systems (CARS), providing a holistic understanding of the current state and potential advancements in context-aware recommendation systems, highlighting various paradigms for incorporating context and exploring different contextual modeling approaches. It then delves into the challenges and limitations of CARS, which serve as the primary motivation for our research.

2.1 Context in recommender systems

Originally introduced by Adomavicius et al. [13], context-aware recommender systems (CARS) have significantly enhanced the personalization and relevance of recommendations by integrating contextual information like time, location, and user mood [95]. These systems go beyond traditional recommender models that rely only on user-item interactions by incorporating additional contextual factors, thus providing more accurate and contextually appropriate suggestions. For instance, a music streaming service might recommend upbeat playlists during morning workout hours and soothing tracks in the evening. An e-commerce platform could suggest buying raincoats during rainy seasons or light attire in hot weather. Travel apps might offer destination recommendations that align with local weather and seasonality, such as promoting skiing trips during winter or beach vacations in summer. Similarly, dining apps could recommend family-friendly restaurants on weekends or romantic spots on special occasions like Valentine's Day. These observations align with findings in behavioral research on consumer decision-making in marketing, which

demonstrate that decision-making is highly context-dependent rather than consistent across different situations. This research indicates that consumers' choices are influenced by various contextual factors, such as time, location, social environment, and psychological state. Therefore, accurately predicting consumer preferences hinges on the recommender system's ability to effectively incorporate these contextual elements into its recommendation algorithms. By integrating relevant contextual information—such as weather conditions for clothing recommendations, time of day for music suggestions, or location for restaurant choices—recommender systems can provide more precise and relevant recommendations that better reflect the dynamic nature of consumer preferences. This foundational shift has set a new standard for integrating context into the recommendation process.

In recent years, numerous context-aware recommender systems (CARS) have been successfully implemented across various domains, demonstrating their versatility and effectiveness. In e-commerce, CARS enhance product recommendations by considering factors such as weather conditions, user location, and recent browsing history, ensuring that suggestions are highly relevant to the user's immediate context [96]. The entertainment industry, particularly streaming platforms, benefits from CARS by offering content recommendations that align with the time of day, the user’s viewing habits, and even their mood, thus enhancing the overall user experience [5]. In the travel sector, CARS provide personalized destination and activity suggestions by factoring in seasonal trends, the user’s current location, and travel history [97, 98]. This ensures that recommendations are not only timely but also tailored to the user's specific preferences and circumstances. Restaurant recommendation systems are another key application, using CARS to provide suggestions based on contextual factors such as the user’s location, dining preferences, time of day, and special events, ensuring that users find the perfect dining options for any occasion [99, 100]. In the music industry, CARS enhance the listening experience by recommending playlists and tracks that fit the user's current activities, such as workout sessions, relaxation periods, or social gatherings [101].

Mobile apps, in general, integrate CARS to deliver personalized experiences by using real-time contextual data such as the user’s location, time of day, and even current activity

detected through sensors [102, 103]. This allows mobile apps to offer more relevant content, notifications, and services, significantly improving user engagement and satisfaction. These applications across various sectors illustrate the broad utility and impact of CARS in delivering highly relevant and personalized recommendations, ultimately enhancing user satisfaction and engagement by aligning suggestions with the user's unique context.

2.2 Paradigms for Context Incorporation

Incorporating context into recommender systems can be achieved through two approaches:

- **Context-driven querying and search**: This approach specifies context in queries to search for the most appropriate items. Queries extract relevant rating data based on contextual information. For example, Google Maps allows users to search for nearby locations, filtering out irrelevant locations using the user's current position.
- **Contextual preference elicitation and estimation**: This approach uses context to model and learn user preferences, predicting unknown ratings for items. For example, a music streaming service can predict which songs a user might like based on the time of day, the user's current activity, and their past listening habits, tailoring recommendations to fit the user's contextual needs.

These two approaches for incorporating context into recommendation systems lead to three paradigms for enhancing the recommendation process as shown in Figure 5. The first one is contextual pre-filtering method, which uses context to filter data before generating recommendations, ensuring only relevant information is considered. The second one is contextual post-filtering method which applies context after generating recommendations, refining or adjusting results to better match user needs. The third one is contextual modeling which integrates context directly into the recommendation algorithm, influencing how user preferences and item ratings are modeled from the outset. These methods improve the recommendation process by leveraging contextual information at different stages.

2.2.1 Contextual Pre-filtering

As depicted in figure 2.1a, Contextual pre-filtering is an approach that leverages specific contextual data to filter appropriate information for making recommendation. This approach treats the context as a query that helps in filtering relevant ratings or data points from a larger dataset. Essentially, it simplifies multidimensional contextual recommendations to a more manageable two-dimensional user-item recommendation model. A significant benefit of the contextual pre-filtering approach is that it enables the use of a wide range of two-dimensional recommendation methods that have been previously introduced [13].

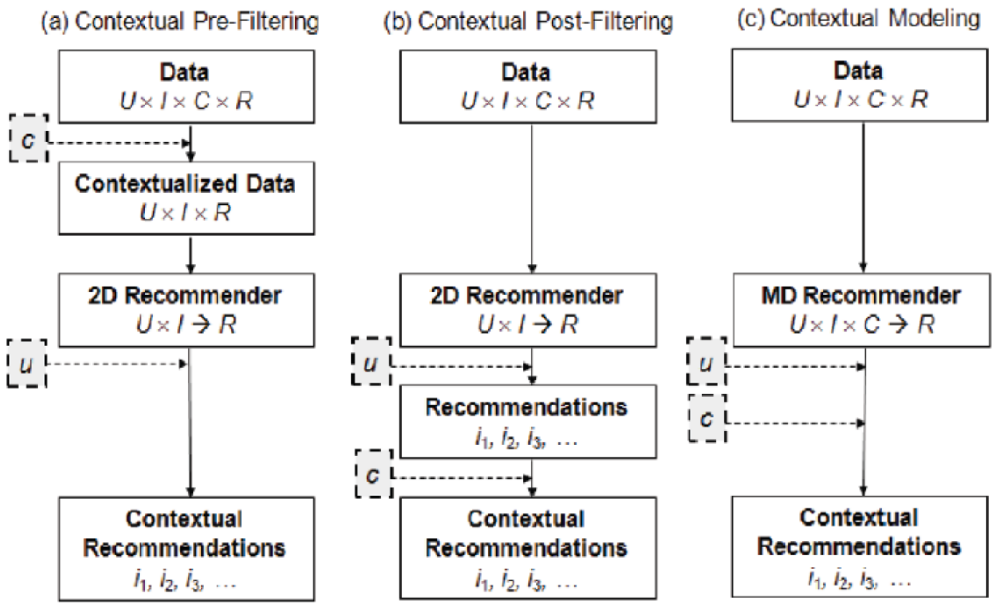

Figure 2.1: Contexts incorporation into recommender systems [86].

One common example provided by Adomavicius et al. [5] illustrates this approach using a movie recommender system. In this scenario, if a user wants to watch a movie on a Sunday, the system only considers the ratings data from Sundays to make movie recommendations to that user. This is an instance of an exact pre-filter where the filtering criteria precisely match the specified context. However, using such exact contextual information can sometimes be too restrictive and may not always be ideal due to a couple of reasons: The first is Generalizability, which means a specific context like watching movies with friends in cinema on Saturdays might not significantly differ from other similar contexts, like Sundays. For

example, the user's preferences might be similar during the entire weekend but differ during weekdays. Thus, a broader context, such as 'Weekend', might be more beneficial and relevant than just 'Saturday'. The second issue is data sparsity, implying that highly specific contexts may not have enough data available. If very few data points meet the exact criteria of friend, cinema, and Sunday, the system may struggle to make accurate recommendations due to insufficient data.

Besides the use of a single pre-filtering model as aforementioned, many works have been focused on combining multiple solutions to enhance performances over single approaches [34, 104, 105]. In fact, combining various contextual pre-filters into a single model can be beneficial. This approach is supported by the observation that a specific context can often be generalized in many relevant ways. Baltrunas & Amatriain [106] presented a contextual pre-filtering technique that leverages implicit user feedback through a method called user micro-profiling. This approach involves dividing the user's profile into several sub-profiles, each representing the user in a different context. Predictions are then made using these micro-profiles rather than a single, unified user model.

Furthermore, the concept of contextual pre-filtering is akin to the creation of local models [107]. Instead of developing a comprehensive global model that encompasses all available ratings, this method constructs a localized model focused solely on ratings that meet specific user-defined criteria, such as time of day. The strength of this local approach lies in its use of data that is more pertinent to the specified context, although it does have the limitation of relying on a smaller dataset. Whether the benefits of relevance outweigh the drawbacks of limited data typically varies based on the application domain and the characteristics of the data at hand.

Other works, such as, Lombardi et al. [108] evaluated the impact of incorporating context into the prediction of relevant items for users, utilizing a pre-filtering approach. The study's findings are threefold. First, contextual models consistently outperform non-contextual ones, showing higher accuracy and true positive rates, particularly when item attributes are included. The context notably enhances the percentage of correctly classified instances. Second, using unstructured text descriptions as independent variables, at least one contextual model demonstrated superior performance, significantly affecting the true positive rate.

Baltrunas et al. [97] proposed a study evaluating a contextual pre-filtering technique for collaborative filtering (CF), termed "item splitting". This technique is predicated on the idea that certain items may receive different evaluations depending on the context. The item splitting method segments item ratings based on contextual features, aiming to create more homogeneous rating groups. This method was compared with a classical context-aware pre-filtering approach, which involves extensive searching to identify contextual segments that enhance baseline predictions. The findings indicate that item splitting proves advantageous in scenarios where a contextual feature significantly impacts the ratings, leading to improved homogeneity within rating groups. Moreover, Codina et al. [109] introduced a cutting-edge contextual pre-filtering strategy that capitalizes on the semantic similarities between contextual situations to enhance recommendation systems. This approach determines context similarity based on user ratings, identifying two contexts as similar if they influence user rating behaviors in comparable ways. The paper addresses a significant limitation of traditional contextual pre-filtering, which typically relies only on user data from the target context when generating recommendations for that context. Instead, their method utilizes ratings from contexts that are semantically similar to the target, employing a new notion of similarity based on the influence of different situations on users' ratings. This method is commonly used as a benchmark for evaluating context-aware recommender systems.

2.2.2 Contextual Post-filtering

Contextual post-filtering (illustrated in figure 2.1b) enhances the adaptability and relevance of recommendations by incorporating context after the initial generation of suggestions. The contextual post-filtering method does not consider contextual information during the initial creation of the ranking for the list of all candidate items from which the top-N recommendations are derived. Subsequently, it modifies the generated recommendations for each user by incorporating contextual data. Adjustments to the recommendation list can be achieved through removing suggestions that do not align with the current context, or adjusting the order of the recommendations to better fit a specified context.

For instance, a movie recommendation system might initially generate a list of movies based on a user's past preferences and ratings. Once this list is created, the system can then

implement a post-filter that takes into account the current context, such as the user's mood (e.g., seeking comedies for a relaxed Sunday evening), the weather conditions (e.g., suggesting indoor movies on a rainy day), or the social setting (e.g., choosing family-friendly films for a family movie night). This method offers versatility and proves especially beneficial in situations where the context can frequently change or is only pertinent at the moment of making the recommendation, ensuring a responsive and personalized user experience.

Contextual post-filtering approaches are divided into two main techniques: heuristic and model-based.

- **Heuristic post-filtering** involves identifying common item attributes relevant to a user in a specific context, and then using these attributes to modify recommendations. For instance, an attribute might be a favored actor or actress that a user prefers to watch in a certain setting.
- **Model-based post-filtering**, on the other hand, involves creating predictive models that estimate the likelihood of a user selecting a particular type of item in a given context, and then adjusting recommendations based on this likelihood. For example, the model might calculate the probability that men will choose to watch a romantic movie in a specific context. Specifically, Panniello & Gorgoglione [114] introduced two methods of adjustment: weightPoF, which reorders recommendations by combining the predicted rating with the probability, and filterPoF, which excludes recommendations that have a low probability of relevance.

Similarly to contextual pre-filtering, a key benefit of the contextual post-filtering approach is its compatibility with a wide range of standard two-dimensional recommendation techniques. Furthermore, these approaches offer rapid responses to users. Traditional recommendation tasks can be completed ahead of time. When context-aware recommendations are needed, the system can swiftly apply contextual adjustments (either filter or modify) to the pre-existing recommendation results.

2.2.3 Contextual modeling

Contextual modeling approaches (illustrated in figure 2.1c) embed contextual data directly inside the recommendation process, treating it as an explicit factor in predicting a user's rating for an item. While contextual pre-filtering and post-filtering methods utilize standard two-dimensional recommendation algorithms, the contextual modeling approach enables the development of truly multidimensional recommendation systems. These systems are essentially predictive models constructed using various techniques such as regressions, probabilistic models, deep learning, or heuristic calculations that incorporate contextual information alongside user and item data. Over the past years, a substantial array of new contextual modeling approaches has emerged, employing a mix of traditional and sophisticated predictive methods. The following section will provide a more detailed exploration of these techniques.

2.3 Contextual Modeling Approaches

The contextual modeling paradigm is commonly adopted for incorporating contextual data into recommendation systems, owing to its effectiveness and performance. This approach enhances recommendations by integrating relevant contextual information to deliver more personalized and accurate suggestions. In this section, we discuss various studies within the field and explore how these approaches contribute to overcoming challenges in recommendation systems, such as sparsity, scalability, and context integration.

2.3.1 Classical techniques

Numerous studies have been conducted using traditional techniques, including heuristic and model-based approaches, which expand the simple two-dimensional models into more complex multi-dimensional models. For instance, Adomavicius et al. [5] a heuristic method based on a two-dimensional model, that consider different possible contexts for each rating. They then aggregate the recommendations derived from each contextual pre-filter. Panniello et al. [111] proposed another heuristic model for Context-Aware Recommender Systems (CARS) by comparing pre-filtering, post-filtering, and contextual modeling methods across

various experimental conditions to assess their predictive performance and diversity. The study underscores the complexity of optimizing CARS and suggests further exploration into hybrid models and the effects of long-term and short-term preferences on recommendations.

In addition to heuristic models, various model-based approaches have also been developed. Oku et al. [99] introduced a method for incorporating context into recommendation systems through the development of Context-Aware Support Vector Machines (C-SVM) and its integration with Collaborative Filtering Systems (C-SVM-CF). This innovation extends traditional SVMs, which are generally used for binary classification, by adding contextual dimensions to the feature space. By applying these models to a restaurant recommendation system, the researchers were able to assess their effectiveness. The results demonstrated that C-SVM can accurately classify user preferences based on varying contexts. Additionally, the C-SVM-CF proved effective, particularly when used among users familiar with the target contexts, showcasing the potential of context-aware methodologies in enhancing the precision and relevance of recommendations. Yu et al. [112] presented a CARS platform called CoMeR, designed for smartphones, utilizing a generic and flexible NtimesMdimensional (N2M) recommendation model combined with a hybrid processing approach. This platform integrates content-based, Bayesian-classifier, and rule-based methods to enhance media recommendation, adaptation, and delivery. The N2M model considers a wide range of context information, including user preferences, situational factors, and the capabilities of both device and network as inputs for recommending both content and presentation formats. This comprehensive approach ensures that media recommendations are both relevant and optimally presented for smartphone users. Abbar et al. [113] proposed a context-aware recommender system that enhances the relevance of recommendations by incorporating both the user's profile and the context of their interactions. Building on previous research in data personalization, they developed a Personalized Access Model that offers a suite of personalization services. These services are specifically designed to advance the functionality of recommender systems through context-aware capabilities. The paper outlines four key services: Context Discovery and Contextualization, which are focused on design, and Binding and Matching, which are directly involved in the recommendation process. Each service is defined by specific operational semantics and supported by distinct

algorithms. Hariri et al. [114] introduced a CARS that transcends traditional personalized recommendation models by incorporating situational awareness. This system delivers recommendations that are not only aligned with a user's preference profile but are also tailored to specific situational contexts. The model leverages a subset of an item's feature space, which represents the short-term interests or needs of a user in a given situation. Contextual information can be either explicitly provided by the user or derived implicitly. The authors propose a unified probabilistic model that integrates user profiles, item representations, and contextual information, using the extended Latent Dirichlet Allocation (LDA) model for joint modeling of users, items, and context-related metadata. This model computes the conditional probability of each item based on the user profile and context, using these probabilities as scores for item ranking.

2.3.2 Tensor-based techniques

Tensor-based techniques offer a variety of sophisticated tools that leverage contextual data to personalize and improve the effectiveness of recommendation systems (RS) [115]. Tensor Factorization (TF), in particular, extends the traditional 2D matrix factorization model into a multi-dimensional representation, allowing the integration of additional data sources. This expansion enables a richer, more nuanced understanding of user preferences and item characteristics by incorporating diverse contextual factors.

For instance, Karatzoglou et al. [101] proposed a Collaborative Filtering approach based on TF, termed Multiverse Recommendation. This method extends traditional Matrix Factorization by transforming the usual 2D User-Item matrix into a User-Item-Context N-dimensional tensor. By treating various types of contextual information as additional dimensions, the data is represented more comprehensively. The factorization of this tensor yields a compact model, enabling the provision of context-aware recommendations that are tailored to specific user circumstances and preferences. Rettinger et al. [116] presented a study on network modeling, specifically focusing on predicting relationships between different classes of objects such as friendships, user preferences, and gene-disease influences. They discussed the effectiveness of factorizing approaches in modeling such relationships. While matrix factorization is suitable for binary relationships, tensor factorization is

employed for ternary relations and is notably applied to analyze the temporal development of these relationships. The proposed method, termed Context-Aware Recommendation Tensor Decomposition (CARTD), features an efficient optimization criterion and a novel decomposition technique, specifically designed to handle the complexities of higher-dimensional data.

One of the most significant challenges is the escalating computational complexity as the number of contextual dimensions grows, which exacerbates the issue of data sparsity. To address these complexities, several researchers have proposed models, such as the case of Chen & Li [117] which proposed the AFT method, that combines TF with adversarial learning to enhance the robustness of context-aware recommendation systems. Traditional TF techniques, while achieves good results in integrating users, items, and contextual factors, often falter in terms of robustness due to the sparsity of the data and the complex, multi-linear nature of the factorization process. The ATF model addresses these challenges by integrating adversarial learning, which not only improves the robustness but also the overall performance of the recommender system. Wu et al. [118] introduced a CARS model named Bias TF, which addresses limitations in traditional tensor-based models by incorporating additional factors like overall mean, user bias, item bias, and context bias. They argue that ratings cannot be fully explained by user-item-context interactions alone, necessitating a broader approach. Furthermore, they highlight three primary drawbacks of existing CARS using TF: the exponential increase in model complexity with more context features, the limitation to categorical context features, and the inability to effectively select relevant features from the available data. To overcome these challenges, they developed an auto-encoding algorithm based on regression trees capable of handling numerical features and selecting effective context features. This algorithm is integrated with Bias TF.

2.3.3 Deep-learning approaches (DL)

In context-aware recommender systems, DL methods are commonly employed for two main objectives: firstly, for representing and modeling context, and secondly, for generating predictions that capture user preferences across both items and contextual settings. Batmaz et al. [119] suggest that the primary reason (DL) enhances recommendation accuracy is its

ability to extract hidden features and synergistically combine information from various sources. Consequently, DL techniques are well-suited for addressing multi-dimensional problems and constraints that are inherently complex and dynamic. Numerous deep learning-based contextual recommender models have been introduced to enhance the traditional architecture of recommender systems, transitioning to an adaptive contextual representation. For example, Xin et al. [120] proposed the Convolutional Factorization Machine (CFM) to address the limitations of traditional Factorization Machines (FM) in capturing high-order and nonlinear interaction signals in context-aware recommender systems (CARS). The CFM enhances the modeling of second-order interactions by using an outer product instead of the inner product, creating "images" that depict correlations between embedding dimensions. These images are then stacked to form an interaction cube, over which 3D convolution is applied to explicitly learn high-order interaction signals. Additionally, CFM incorporates a self-attention mechanism for feature pooling to reduce time complexity. Unger & Tuzhilin [121] introduced a DL recommendation framework designed to integrate contextual data into neural collaborative filtering approaches. Recognizing the dynamic and high-dimensional nature of context in various applications, they developed three deep context-aware recommendation models that utilize explicit, unstructured, and structured latent representations of contextual data from different dimensions such as time, location, and user activity. These models are tailored for distinct purposes including rating prediction, generating top-k recommendations, and classifying user feedback.

To better understand the interactions between contexts and enhance the interpretability of recommendations, numerous attention-based models have been developed in recent years. For instance, Mei et al. [122] proposed a neural model called the Attentive Interaction Network (AIN) to improve CARS by adaptively capturing the interactions between contexts and user/item pairwise. The AIN model features three key components: an Interaction-Centric Module that captures the effects of contexts on user/item interactions; a User-Centric Module and an Item-Centric Module that respectively model the impact of these interaction effects on user and item representations. These representations are then combined to predict recommendation scores. Additionally, AIN employs an effect-level attention mechanism to aggregate multiple interaction effects. Extensive testing on two rating datasets and one

ranking dataset demonstrated that AIN surpasses current leading CARS methods. Ding et al. [123] proposed an approach called Mixture Attentional (MACDAE) aimed at enhancing real-time Online-to-Offline (O2O) recommender systems. Unlike traditional recommendation systems where user preferences are static, O2O systems need to dynamically adapt to changes in users' intentions. MACDAE addresses this by first using interactions among user, item, and explicit context features to deduce users' implicit context features. These inferred contexts are then integrated into an end-to-end model to generate suggestions.

Sequential recommendation is another area where DL is applied in context-aware systems. Numerous studies have been conducted to manage users' activities over time. For instance, Beutel et al. [124] presented a study focused on the effective treatment of contextual information within neural recommenders. Their research began with an empirical analysis of the traditional method of integrating context as features in feed-forward recommenders, which they found to be inefficient in handling common features. Leveraging this insight, they designed art RNN recommender system, exemplified by its implementation at YouTube. They introduced "Latent Cross," a straightforward technique to integrate contextual information into the RNN. This method involves first embedding the context feature, followed by an element multiplication of this context embedding with the model's hidden states, enhancing the system's ability to leverage contextual information effectively. Liu et al. [125] introduced a CARS Sequential Model (CASM) for multi-behavioral recommendations, an approach that enhances the capabilities of sequential models by integrating multiple types of user behaviors. CASM addresses this gap by using context-aware multi-head self-attention layers to analyze the dependencies among different behaviors in historical interactions. Additionally, it employs a weighted binary cross-entropy loss to differentiate the impact of various behaviors during the learning process, allowing for more tailored recommendations based on specific scenarios. This approach not only leverages the strengths of sequential models but also adapts to a range of user behaviors, enhancing the model's overall effectiveness in diverse recommendation environments.

2.4 Challenges

The topic of context-awareness has seen substantial growth in recent years, as evidenced by the surge in academic publications and the adoption of Context-Aware Recommender Systems (CARS) by leading companies in their platforms. Despite this progress, the field continues to evolve rapidly, highlighting many interesting and critical research problems that remain unexplored. This ongoing development presents various challenges and open issues that need to be addressed, serving as the driving motivation behind our work in this area. These challenges can be summarized as follows:

- **Scalability:** The diversity and relevance of context in context-aware recommender systems present a significant challenge, as context can vary widely from tangible factors like location and weather to more abstract elements such as a user's mood or social setting. Determining which context variables are relevant and understanding their impact on recommendations requires complex decision-making. Furthermore, the integration and interpretation of this broad spectrum of contextual data, alongside user and item information, demand sophisticated models and algorithms capable of efficiently managing and analyzing multi-dimensional data. This complexity underscores the need for advanced approaches in the design and implementation of these systems.
- **Sparsity and Cold start:** Data sparsity in CARS presents a major hurdle, as it occurs when there is insufficient user-item interaction data available within specific contexts, exacerbating the challenge when additional context variables are considered. This lack of comprehensive data leads to poor prediction accuracy and difficulties in effectively tailoring recommendations to suit individual user preferences in varying contexts. Additionally, like traditional recommender systems, context-aware systems struggle when encountering new users or new contextual conditions. The lack of historical data in these scenarios makes it difficult to generate reliable recommendations, further complicating the deployment and effectiveness of these systems.

- **Dynamic Context Handling:** The challenges of temporal dynamics and evolving preferences in context-aware recommender systems underscore the complexity of designing adaptive systems. Contexts can shift rapidly, making what was relevant one moment potentially irrelevant the next, which necessitates systems capable of adjusting in real-time or near-real-time, a process that is both technically demanding and resource-intensive. Simultaneously, users' preferences can evolve as contexts change, and without a robust historical dataset to inform these shifts, it becomes increasingly difficult to maintain the accuracy and relevance of recommendations.
- **Privacy:** Handling sensitive data and addressing bias and fairness are crucial challenges in context-aware recommender systems. Contextual data, which can include sensitive information about a user's location, activities, or emotional states, requires meticulous management to ensure compliance with privacy laws and ethical standards. Additionally, there is a risk that these systems might unintentionally learn and perpetuate existing biases in the data or context, resulting in recommendations that could be unfairly skewed against certain user groups. Both aspects emphasize the need for careful system design to protect user privacy and promote equity in recommendations.
- **Accuracy vs. Personalization Trade-off:** The risks of overfitting to specific contexts and the challenge of maintaining a balanced approach are significant concerns in context-aware recommender systems. Overfitting occurs when a system becomes too finely tuned to particular contextual details, leading to overly niche recommendations that may not resonate with users in different situations. Simultaneously, there is a need to strike an optimal balance between utilizing contextual information for personalized recommendations and ensuring the recommendations remain broadly appealing across various contexts. This balancing act is crucial for the effectiveness and user satisfaction with the recommender system.

Despite the various challenges shared with traditional recommender systems, a predominant issue in CARS is the difficulty in obtaining contextual information from users, due to various

constraints. Although much of the existing research has concentrated on leveraging contextual data to enhance algorithmic accuracy and model efficiency, relatively little attention has been given to innovating methods for the collection of context data. Historically, efforts in developing new approaches for this initial stage of data collection have been limited, as seen in [126, 127]. This stage is as crucial as context integration and rating prediction, offering a fertile ground for future research. There lies a significant opportunity for researchers to develop models that effectively address the challenges associated with gathering contextual information.

2.5 Conclusion

In this chapter, we have examined current research in the field of context-aware recommender systems, highlighting established contextual modeling paradigms including contextual pre-filtering, post-filtering, and modeling. We also pinpointed key challenges that require further attention from researchers, such as the difficulties associated with collecting contextual data, along with issues of data sparsity and high dimensionality. These challenges align with the research questions that we outlined in Chapter 1.

These limitations serve as motivation for us to enhance the robustness of Context-Aware Recommender Systems (CARS), spanning from data extraction to rating prediction. In the following chapter, we introduce innovative methods for context collection utilizing advanced model.

Part II

Context extraction and ratings prediction models

Chapter 3

Context extraction

Chapter 2 reviews context-aware recommender systems and the challenges they encounter. The subsequent chapter introduces a solution to address the shortage and unavailability of contextual information. We propose a novel model that extracts contextual information, which relies on a custom named entity recognition system (CustomNER) to automatically gather contextual data from user reviews. The experimental results indicate that this model performs well. However, like most models, it has limitations. One significant limitation is the reliance on a public corpus to train the CustomNER, which is not specifically designed for context identification and lacks the necessary specific contexts.

Therefore, we propose a new approach that initially constructs a contextual corpus from open data sources. This method facilitates more tailored named entity recognition, specifically designed for extracting contexts pertinent to recommendations. This enhanced approach surpasses our original model in performance, delivering more effective results.

3.1 Introduction

Recommender systems utilize various types of input, specifically explicit and implicit data [13]. The most accurate among these is explicit feedback, where users provide direct input about their preferences for specific items (e.g., Netflix uses a five-star rating system to collect user ratings). For a recommender system to accurately capture a user's preferences and offer dependable recommendations, it is essential that the user rates a sufficient number of items. However, in real-world scenarios, only a few users actually rate items, resulting in rating

matrices that are mostly empty. This problem is known as sparsity [128]. After collecting these initial ratings, a traditional RS will attempt to predict the rating R.

$$R: User \times Item \rightarrow Rating \quad (3.1)$$

Traditional recommender systems primarily utilize two fundamental dimensions: the user and the item dimensions, to generate personalized recommendations. In the content-based approach, a more nuanced method is employed that leverages item attributes and user profiles during the recommendation process. By examining the characteristics of items alongside users' preferences and behaviors, content-based systems deliver recommendations specifically tailored to each user based on the content. On the other hand, collaborative filtering methods adopt a different strategy. These methods do not require detailed information about users or items; rather, they rely on user ratings to determine preferences for specific items. However, these approaches have their limitations because they fail to consider various factors that could influence a user's preferences, such as contextual elements. For instance, if a user typically enjoys sci-fi movies when watching alone, a traditional recommender system (RS) would suggest a sci-fi film based on his past preferences. However, when this user is with his young son and looking for a movie to watch together, the recommender system might still recommend a sci-fi movie. While this genre aligns with the father's preferences, it may not be appropriate for his son, as sci-fi movies often include scenes of violence and are generally not suitable for children under the age of 16. This example highlights the significance of contextual factors in recommendation systems. These factors can greatly influence preferences and turn an otherwise suitable recommendation into an inappropriate one based on the viewing context.

To resolve this issue, the development of Context-Aware Recommender Systems (CARS) represents a significant advancement. These systems utilize contextual data to enhance the accuracy and relevance of recommendations, ensuring they are tailored to the specific circumstances and preferences of each user. After incorporating this additional data, the rating function R evolves from a two-dimensional model to a multi-dimensional one.

$$R: User \times Item \times Contexts \rightarrow Rating \quad (3.2)$$

However, a major challenge for Context-Aware Recommender Systems (CARS) is the collection of contextual information, which often includes sensitive personal data, raising privacy concerns as outlined in Chapter 1. Privacy issues are prevalent across many domains and there are no straightforward solutions due to the complexity involving multiple stakeholders such as companies, users, and governments. In this work, we do not aim to address the root of the problem directly; instead, we focus on identifying new data sources that could provide relevant contextual information.

User reviews are a form of unstructured data that offer a rich source of insights for various applications, including recommender systems. This source of data represents an effective solution to one of the main challenges encountered by recommender systems which is data scarcity. Recent researches have introduced approaches that exploit reviews in order improve recommendations, acknowledging their capability to derive valuable insights while avoiding privacy concerns. For example, Levi et al. [129] introduced a Cold Start RS that utilizes group contexts extracted from reviews. Their approach includes multiple components: a weighted text mining algorithm, sentiment analysis to assess sentiments related to hotel features, clustering to develop a specific vocabulary for each hotel, and nationality-based groupings. Zheng et al. [130] developed a new model called Deep-CONN, which utilizes word vectorization techniques alongside two convolutional neural networks (CNN). The first CNN extracts insights into user behaviors from user reviews, while the second is dedicated to deriving item characteristics. The outputs from these two networks are then merged and processed through a factorization machine algorithm to generate predictions. Similarly, R. Catherine & W. Cohen [131] presented Trans-Nets model that features two parallel CNN to process selected reviews and handle associated text from user/item pairwise. Similar to the Deep-CONN model, Trans-Nets uses a Factorization Machine to predict rating. A distinctive feature of Trans-Nets, is its extra layer designed to represent user-item pairwise. his layer significantly improves the model's capacity to harness and utilize contextual data to personalize recommendations. McAuley & Leskovec [132] proposed a Hidden Factor and Hidden Topics (HFT) model that integrates user-provided reviews and rating scores to create recommendations. The proposed method uses Latent-Factor RS for rating predictions then, applies Latent Dirichlet Allocation (LDA) to reveal hidden aspects from reviews. By merging

these approaches, the HFT adeptly combines scores and data extracted from text, delivering recommendations that are enriched by underlying factors and themes. Zhang et al. [133] introduced an Explicit Factor Models (EFM) to provide recommendations that are both understandable and justifiable. The EFM works by deriving user sentiments and explicit item characteristics from reviews. It uses these insights to decide whether items should be recommended or not, based on the hidden attributes it uncovers, the characteristics of the items, and the expressed interests of the users. By integrating user sentiments and item features, EFM strives to deliver recommendations that are not only precise but also interpretable, offering users clear explanations for the recommendations. The studies mentioned earlier have certainly utilized reviews to improve recommender systems; however, they frequently neglect the advantages of integrating contextual data, which could significantly boost the accuracy and personalization of recommendations. Only a limited number of studies have been conducted on extracting contextual data from reviews such as the case of Aciar [134], which introduced a technique that employs classification rules and text mining to detect user's interest and contexts in reviews. Even so, it is crucial to point out that this method mainly aims to identify sentences that hold contextual clues but might not thoroughly extract or exploit this data for recommending purposes. Although the method acknowledges the presence of context, it may not fully utilize this essential information to enhance recommendation outcomes. Hariri et al. [135] presented a CARS noted for its effective use of user reviews to extract contextual information. This method integrates these insights with the user's historical ratings to create an all-encompassing utility function. The system approaches context as a supervised topic modeling challenge, employing a classifier built on labeled LDA for context categorization. Although traditional recommendation algorithms are employed to predict ratings, it is essential to emphasize that the primary objective of this system is to predict the utility function, rather than just the ratings. Lahlou et al. [127] introduced a review-based RS that leverages reviews to generate contextual suggestions. Their designed architecture automatically utilizes contextual information from reviews to form recommendations. This approach has demonstrated promising outcomes in improving recommendation accuracy. However, one drawback is that it considers the entire review as context, rather than isolating specific contextual elements. In practical settings,

only a minority of reviews contain relevant contextual details, potentially limiting the effectiveness of their method in situations where such rich contextual information is sparse.

To tackle the critical issue of gathering contextual data for CARS, we propose our first approach that introduces a new model for contextual information extraction. This model is paired with a specialized mechanism designed to assimilate the extracted data, aiming to improve prediction accuracy. Our innovative strategy emphasizes extracting context from unstructured sources such as user reviews and seamlessly incorporating this data into the recommendation process. The proposed model utilizes a pre-trained language representation method called BERT, which is trained on extensive datasets from free open-source data, combined with a custom Named Entity Recognition system. Although the model has achieved commendable results, it struggles to fully extract contexts from reviews due to its reliance on public corpora for training the CustomNER. These corpora are primarily designed for entity extraction rather than context extraction. To address this limitation, we propose a second approach that includes a specialized component for building a corpus from DBpedia, enhancing the model's ability to identify and utilize contextual information effectively.

3.2 Context dimensions definition

The term "context" refers to the circumstances, conditions, or factors that surround and give meaning to an event, an idea, or a statement, thereby influencing how it is understood and interpreted. Context can encompass a wide range of elements, from the immediate physical surroundings and historical background to cultural settings and specific details of a situation. Context as multifaceted concept, is extensively studied within various disciplines such as computer science, Psychology, Linguistics, Sociology and Anthropology.

To apply the proposed approach for identifying context information in reviews, it is essential to first define the context dimensions (also known as context categories or context variables) and their corresponding values. The literature features numerous proposals for context modeling, especially within the fields of computer science (Artificial intelligence and ubiquitous computing). For example, Chen et al. [136] proposed an ontology for ubiquitous computing called SOUPA and its implementation within the Context Broker Architecture

(CoBrA). CoBrA represents a novel agent architecture designed to support pervasive contextual systems in smart space environments. The SOUPA ontology, crafted using the Web Ontology Language (OWL), incorporates modular component vocabularies that cover a range of elements including intelligent agents, desires, and intentions, as well as concepts of time, space, events, user profiles, actions, and security. Castelli et al. [137] proposed a comprehensive framework for representing and processing global data through a uniform model designed to handle inputs from diverse sources. Their model centers on the use of a four-field (Who, What, Where, When) to represent world facts in a structured manner. These tuples, referred to as knowledge atoms, serve as atomic units of factual knowledge and can accommodate context and incomplete information. The data model facilitates complex semantic querying over large datasets and enables simple pattern-matching queries, enhancing user interaction with context-aware services. Chaari et al. [138] introduced a framework designed to ensure that applications adapt effectively within a pervasive computing environment. Their approach encompasses a comprehensive adaptation strategy that includes content, presentation, and core service adaptations to align with various contextual factors. The authors categorize essential context descriptors into seven types: user context, device context, network context, activity context, service context, location context, and resource context. This classification helps in tailoring applications more precisely to the specific conditions and requirements of their operating environment, thereby enhancing functionality and user experience. Kim et al. [139] presented an ontology-based context-aware modeling technique that utilizes a framework for the efficient specification of contextual information, aiming to enhance context management and reasoning within intelligent services. Their approach leverages the "5Ws and 1H" maxim (why, who, what, where, when, and how) to process both physical and logical contextual data effectively. This methodology allows for a clear, intuitive schema that facilitates the sharing and integration of contextual information. The proposed model maps these elements, goal, role, action, status, location, and time, onto corresponding aspects of context-aware processing, demonstrating the model's high applicability and utility in dynamic environments. Campos et al. [126] proposed a method and accompanying tool designed to construct a context taxonomy using Linked Data. This taxonomy is intended for use by context-aware

applications beyond just recommendation systems. Additionally, the method is versatile and can be adapted to build other types of taxonomies, such as those describing features or aspects of items, which could then be used in feature-based opinion mining and recommendation applications. Based on their analysis, the authors have incorporated four major context dimensions into the taxonomy: Location, Time, Environmental, and social contexts, each encompassing a variety of specific categories. Following a thorough examination of prior research, it is evident that a significant approach to context modeling involves employing taxonomy or ontology. Based on insights gleaned from these models, we have chosen to integrate four essential dimensions of context: Location, Companion, Weather, and Time. Table 3.1 provides a summary of the principal modeling techniques used for contextual dimensions definition.

3.3 Proposed methods

The primary objective of both methodologies is to derive contextual insights from user

Table 3.1: Categories proposed from principal context modeling techniques

Ref.	Categories
[136]	Time, Space, Person, Event, Geo-spatial, Agent, Action, Policy
[137]	Location, Time, Fact, Person.
[138]	Location, User, Resource, Network Service, Device, Activity.
[139]	Location, Time, Status, Goal, Role, Action.
[126]	Time, Social context, Time, Environmental context

reviews. Although these methods share numerous similarities, the second model is notably superior in terms of its performance capabilities and its specialized focus. This section begins with a description of the initial model, followed by a comprehensive discussion on the advancements and specific features of the enhanced second model.

3.3.1 Custom NER for context extraction: First method

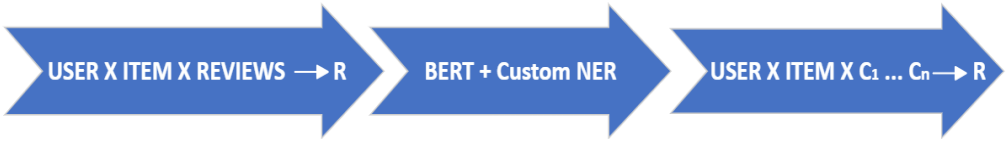

Figure 3.1: The transition from (USER, ITEM, REVIEW) to (USER, ITEM, CONTEXTS).

To accomplish our main objective, we designed a novel model to extract context from user reviews. As illustrated in Figure 3.1, our goal is to transition from the basic triple format (USER X ITEM X REVIEWS) that is incompatible with recommender systems, to a more complex, multi-dimensional representation (USER X ITEM X C_n). This model comprises two key components: a custom Named Entity Recognition (NER) [140] system and a BERT model [141].

- **Named Entity Recognition (NER):** NER is an essential technique in Natural Language Processing (NLP) aimed at identifying and categorizing specific pieces of information, known as named entities, within unstructured text. These entities typically include categories such as names of people, organizations, locations, time, etc. The NER process begins with tokenization, where the input text is segmented into discrete units or tokens, which can be words or phrases. The core of a typical NER system consists of a neural network architecture composed of three primary layers. The first is the word embedding layer, which converts each token into a dense numerical vector. These vectors are designed to capture the semantic properties of words, meaning that words with similar meanings have similar vector representations. This transformation is crucial as it shifts the data from a textual form to a numerical form that machine learning models can operate on. Following the word embedding layer is the context encoder layer, often implemented using a Bidirectional LSTM (Bi-LSTM). This layer processes the sequences of embeddings to extract contextual relationships within the text. The bidirectionality of the LSTM allows the model to gather context from both the past and the future tokens relative to the current token being processed, which is instrumental in understanding the full context of each word. The final layer in the NER architecture is the decoder layer. This component of the

model uses the context-rich representations generated by the Bi-LSTM to classify each token into one of the predefined categories of named entities. The decoder predicts not only the type of entity each token represents but also where each entity begins and ends in the text. This structured approach allows NER systems to effectively parse and categorize data within large volumes of text, facilitating tasks such as information extraction, content categorization, and the automation of document processing. Moreover, the accurate recognition and classification of named entities serve as a foundation for more complex NLP applications, including machine translation, question answering, and the emerging field of Context Aware Recommender Systems, where understanding the specific context and content of user interactions can significantly enhance recommendation accuracy.

- **BERT:** Bidirectional Encoder Representations from Transformers significantly advances the state of the art in natural language processing (NLP) by leveraging the Transformer architecture, a model designed primarily around self-attention mechanisms. This architecture differs from previous models by abandoning the reliance on sequence-aligned recurrent layers (like those found in RNNs or LSTMs) and instead uses attention to weigh the importance of different words in a sentence, regardless of their positional order. Unlike GLoVe [142] and Word2Vec [143], The key innovation of BERT is its bidirectional training, which is a departure from earlier models that processed text in a single direction (either left-to-right or right-to-left). BERT, however, reads the entire sequence of words at once, thereby learning the context of a word based on all of its surroundings (both left and right of the word). This is implemented through the "masked language model" (MLM) pre-training objective, where random words in a sentence are replaced with a [MASK] token, and the model is trained to predict the original word based on its context. Furthermore, BERT also employs a second training objective known as "next sentence prediction" (NSP), where the model learns to predict if a given sentence logically follows another. This dual training approach enables the model to understand language context and relationships more deeply. BERT's architecture consists of multiple layers of these Transformer units stacked on top of each other, with each layer processing the entire

input sequence simultaneously and independently. This allows the model to capture complex syntactic and semantic relationships within the text. The depth of these layers (often 12 for the base model and 24 for the larger variant) is a key factor in its ability to model complex patterns and nuances in language, making it highly effective for a variety of tasks. Figure 3.2 represents BERT architecture; The arrows show the flow of information from layer to layer. E1, Trm and T1 respectively represent the embedding representation, the intermediate representations and the final output for a given word.

Our model is structured into three distinct layers, as depicted in Figure 3.3. These layers include a word embedding layer, depicted as a Bidirectional Long Short-Term Memory (Bi-

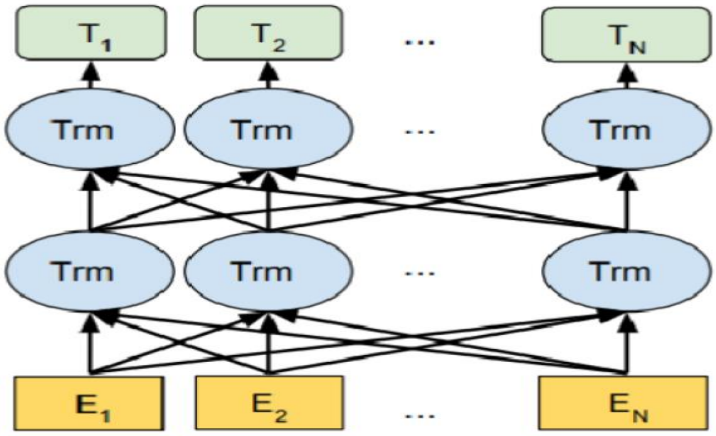

Figure 3.2: BERT architecture [141].

LSTM) [144] layer, and a Conditional Random Fields (CRF) [145] layer. Initially, the BERT pre-trained model processes a sequence of n words (w_1, w_2, ..., w_n) in the word embedding layer, producing a contextual embedding vector for each word. Subsequently, these embeddings are fed into the Bi-LSTM layer. The Bi-LSTM, an enhancement of the traditional LSTM that incorporates both forward and backward processing networks, is designed to mitigate issues such as gradient vanishing and exploding, enhancing the handling of long sequences. This layer concatenates outputs from both directions to form a comprehensive feature vector for each word [H_l, H_r]. The final layer, a CRF layer, is then employed to determine the most probable sequence of tags for the tokens. As a probabilistic discriminative model, the CRF effectively learns and enforces labeling constraints, such as those in the BIO

(Beginning, Inside, Outside) tagging format, to ensure sequence validity. For instance, it automatically learns that a sequence should not start with an "I" tag, thereby maintaining the logical structure of the annotations. Figure 3.3 shows the architecture of the custom NER.

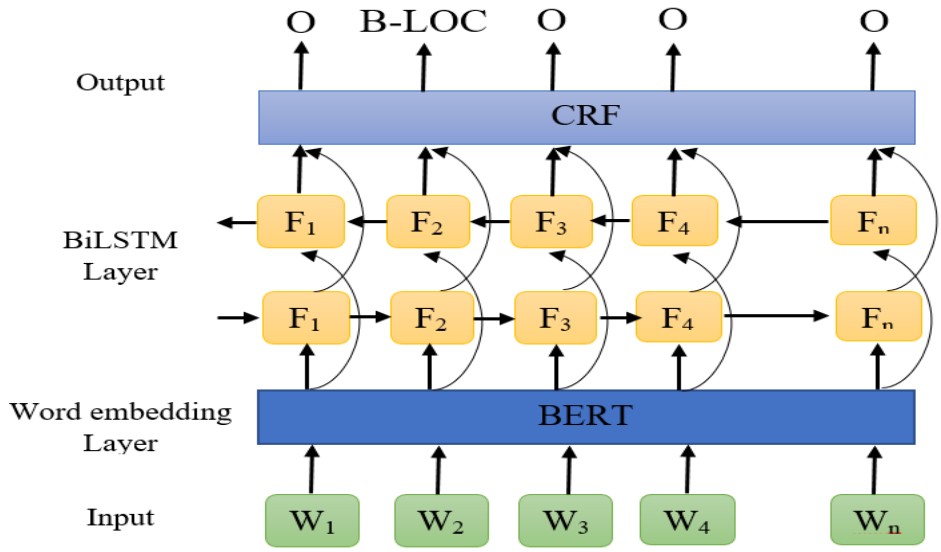

Figure 3.3: The architecture of the custom NER.

To train our custom NER model, we begin by integrating three diverse corpora, each selected to address specific aspects of entity recognition. First, we utilize the CoNLL-2003 NER Dataset, which includes a broad range of common entities across 18,453 sentences. This dataset helps establish a strong baseline by focusing on fundamental entity types like persons, locations, and organizations. Next, we incorporate the Groningen Meaning Bank (GMB) to introduce more complexity and diversity with its eight distinct entity types, including Geopolitical Entities and Natural Phenomena, spread across 63,256 sentences. This expansion allows our model to handle a wider variety of texts and contexts.

In our model, sequences are processed through layers, resulting in a matrix P from the previous layer and a transition matrix T. The scoring of a tag sequence is calculated as follows:

$$\text{Score}(X, Y) = \sum_{i=1}^{n} P_i, y_i + \sum_{i=1}^{n} T_y, y_{i+1} \qquad 3.3$$

Where:

- X represents the sequence and Y his label.
- P_{i,y_i} represents the score from the matrix P for the label y_i.
- T_{y,y_i} represents the transition score from matrix T for moving from label y_i.

When evaluating the performance of our model, we observe impressive results in certain contexts, yet it underperforms in others. This inconsistency is not due to the model's structure or its inherent capabilities, but rather stems from the characteristics of the training corpora. These datasets are specifically curated for entity extraction and may not fully encapsulate the diversity or specificity of entities required for optimal performance across all contexts.

To address these specific deficiencies observed in preliminary model outputs, such as the inadequate recognition of relational and situational entities, we built a custom corpus. This corpus is specially designed with 3,500 sentences to fine-tune our model's ability to discern nuanced entities like Companion and Environmental Contexts in everyday scenarios, ensuring a more refined and context-aware entity recognition capability. This comprehensive training approach leverages the strengths of each corpus to build a robust and adaptable NER model.

Despite improvements from our custom training approach, the model still faces significant limitations. Firstly, building a custom corpus is a manual and labor-intensive process requiring extensive effort in feature engineering and data preprocessing. Secondly, the relatively small size of the corpus adversely affects the quality of training, limiting the model's ability to learn and generalize effectively across diverse scenarios.

3.3.2 Custom NER for context extraction: Second method

To address the observed deficiencies in preliminary model outputs, particularly the inadequate recognition of relational and situational entities as outlined in section 4.3.1, and to enhance the overall performance of our custom NER, we propose a novel methodology. Initially, we compiled a large corpus by extracting data from Dbpedia [146] using SPARQL queries. This data was then converted into BIO-format tags, which are used for training Named Entity Recognition (NER) models. Our customized NER model utilizes a combination of BERT, Bi-LSTM, and BiCRF technologies to more effectively identify contextual elements. Our methodology, depicted in Figure 3.4, involves three key

subprocesses: building the corpus, extracting context, and predicting ratings. In this section, we will discuss the first two subprocesses, with the third step detailed in next chapters.

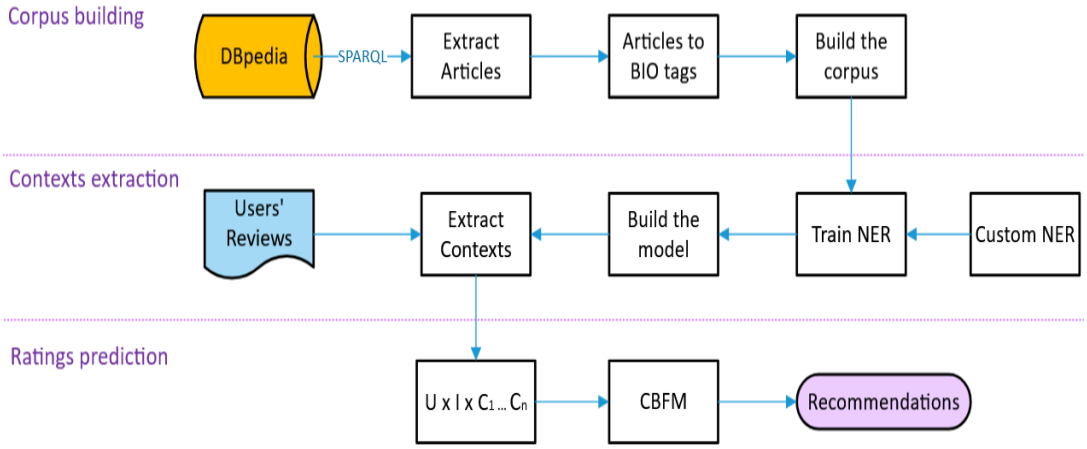

Figure 3.4: Main process steps from our proposed CARS.

- **Corpus building:** Once the context dimensions are established, as outlined in the previous section, the initial step is to extract textual content from DBpedia. Notably, DBpedia utilizes the Simple Knowledge Organization System (SKOS) extensively for organizing data. Leveraging SKOS relations allows for navigation through the interconnected DBpedia categories in a hierarchical fashion, covering both primary categories and their respective subcategories. This involves selectively identifying relevant "root" categories in DBpedia, such as 'dbc:Places' for locations. Table 3.2 displays the various contexts and their respective categories.

Table 3.2: Contexts and their associated categories in DBpedia.

Context	**Description**
Time context	dbc: Time
Location context	dbc: Places dbc: Buildings and structures dbc: Educational institutions dbc: Transport by mode
Companion context	dbc: Interpersonal relationships
Weather context	dbc: Weather dbc: Meteorological phenomena

For retrieving and manipulating data stored in Resource Description Framework (RDF) format, we use SPARQL [147], a dedicated query language. RDF data is structured as triples, each consisting of a subject, predicate, and object, forming a graph of data. It enables complex queries across diverse data sources, treating them as part of a global database. SPARQL supports several query forms, including `SELECT` for retrieving variables, `CONSTRUCT` to return RDF triples, `ASK` for boolean results, and `DESCRIBE` for RDF data about resources. Its capabilities extend to graph pattern matching, optional and alternative matching with `OPTIONAL` and `UNION` clauses, and expressive filters using various functions. Additionally, SPARQL allows aggregations, grouping, sorting, and executing queries across different databases through federation. The update capability in SPARQL 1.1 enhances its utility, permitting users to insert, delete, and modify data in RDF graphs. For example, a simple SPARQL query might involve selecting the names of books and their authors by matching patterns in the data graph, demonstrating its power in facilitating complex data integration tasks necessary for linking large volumes of distributed data in semantic web applications.

Figure 3.5 depicts an example of a SPARQL query. This query is designed to retrieve all articles that fall under the 'places' category and its subcategories, up to six levels deep, and are written in English.

```
from SPARQLWrapper import SPARQLWrapper, JSON
# Set up the SPARQL endpoint URL
endpoint_url = "http://dbpedia.org/sparql"
# Set up the SPARQL query
query = """
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dct: <http://purl.org/dc/terms/>

SELECT DISTINCT ?article ?abstract WHERE {
  ?subcategory skos:broader{1,6} <http://dbpedia.org/resource/Category:Places> .
  ?article dct:subject ?subcategory .
  ?article dbo:abstract ?abstract .
  FILTER (lang(?abstract) = 'en')
  }
"""
```

Figure 3.5: Example of SPARQL langage.

- **Context extraction:** Following the construction of the corpus, a critical component of our work, the next step involves identifying and categorizing contexts within the text. This is achieved through the use of a carefully designed custom NER system, which leverages nuanced capabilities for precise analysis. To capture contextual information from reviews, each sentence undergoes a three-step processing sequence. The first step involves word embedding, where word sequences are transformed into embedding sequences using a pre-trained BERT model. The outputs from this step serve as inputs for the second step, which utilizes a Bi-LSTM network. The Bi-LSTM comprises two LSTMs: the first processes the input sequence in a forward direction (from beginning to end), while the second processes it in a backward direction (from end to beginning). This bidirectional approach allows the model to capture contextual information for each token by considering both preceding and following tokens. The outputs of the two LSTMs are typically concatenated or otherwise combined to create a richer contextual representation for each token. The key distinction between the two proposed models lies in the third step, which involves the layer used. Linear chain CRFs have a known limitation in their ability to capture label dependencies, primarily directed forward. For instance, in complex entities like "Hassan II University," these models might mistakenly classify "Hassan" simply as a name because they cannot consider the "University" token that appears later. To overcome the limitation of linear chain CRFs, which are predominantly forward-looking and may struggle with complex entities, employing a BiCRFs proves to be a beneficial approach. This allows for capturing context from both before and after each token, enhancing accuracy in entity recognition. In the third step, Bi-CRFs consider label dependencies from both directions, meaning they take into account the labels both preceding and following the current token. This dual-directional approach is particularly useful for handling complex entities or scenarios where label transitions aren't strictly left-to-right. By incorporating this methodology, the Bi-CRF enhances the accuracy and coherence of labeling across different segments of an entity, even those that are distantly related within the sequence. The output of this step is the final prediction of word sequence with their BIO tags.

3.4 Experiments

This section aims to evaluate the effectiveness of the proposed models comprehensively. Initially, we will describe the datasets and corpora used to train the models, ensuring a thorough understanding of the empirical foundation for the study. Following this, we will present, compare, and discuss the results obtained from these models.

3.4.1 Corpora and datasets

Various corpora have been utilized in training the models discussed in this work, as well as in extracting contextual data.

To train the first CustomNER model, we utilized three distinct corpora:

- **Corpus 1**, employed in the training of the CustomNER, is based on the CoNLL-2003 Named Entity Recognition (NER) dataset [148]. This dataset is composed of 18,453 sentences, totaling 254,983 tokens. It is specifically structured to identify and categorize four types of entities: persons, locations, organizations, and miscellaneous items. The rich diversity of entity types within this dataset provides a comprehensive foundation for training NER models, enhancing their capability to recognize and classify a broad spectrum of entities accurately in text. Therefore, it lacks of some contextual entities or dimensions, which impact the final predication for same contexts.
- **Corpus 2** used in the CustomNER model training is the Groningen Meaning Bank (GMB) [149] developed at the University of Groningen. This corpus is specifically tailored for named entity classification, containing 63,256 sentences and 1,388,847 tokens. It is meticulously annotated with eight distinct entity categories: Geographical Entity, Organization, Person, Geopolitical Entity, Time, Artifact, Event, and Natural Phenomenon. The broad range of entities covered by the GMB enables a more nuanced understanding and classification of complex data structures, thus enhancing the model's ability to accurately categorize and respond to diverse linguistic inputs. This depth and variety make it an invaluable resource for training sophisticated NER

systems. This corpus helps to address deficiencies in certain contexts; however, it remains insufficient on its own.

- **Corpus 3** is a bespoke corpus that we developed specifically to address gaps identified in the first two corpora. During the initial phases of training our CustomNER, we noticed that it struggled to accurately extract certain categories, such as Companion context and Environmental context. For example, in the sentence "I watched the movies with my friend at the cinema," the words "friend" and "cinema" should be annotated as Companion and Location respectively, but were not. To rectify this issue, we crafted this new corpus in the BIO (Beginning, Inside, Outside) annotation format. It comprises 3,500 sentences and 43,565 tokens, focusing on three key entities. This targeted corpus allows for more precise fine-tuning of our CustomNER, enhancing its ability to recognize and categorize these previously missed contexts more effectively.

To train CustomNER2, a specialized contextual corpus was constructed to address the deficiency in contextual entity representation found in the previously mentioned corpora. This newly developed corpus adheres to the same structural format as its predecessors, utilizing the BIO (Beginning, Inside, Outside) tagging format to maintain consistency and compatibility with our CustomNER system. Comprising 500,000 sentences and over 20 million tokens, this corpus is enriched with annotations across four distinct contexts: Location, Time, Weather, and Companion. Due to constraints in computational resources, the training was limited to just 10% of the data, which was selectively sourced from DBpedia.

To effectively extract contextual information from user reviews and evaluate our models, we selected two comprehensive datasets: the Amazon Customer Reviews Dataset [150] and the Yelp Dataset [151]. The Amazon dataset is particularly extensive, comprising over 233.1 million customer reviews, ratings, and detailed product metadata such as price, brand, and descriptions. These reviews span from 1996 to 2018 and encompass 21 different product categories, making it the largest publicly available dataset for product ratings. We selected video games category for this work. On the other hand, the Yelp dataset, which is freely available for academic and personal use, includes more than 8.5 million reviews and data on

over 160,000 businesses. This dataset provides a rich source of consumer feedback and business attributes, which are essential for testing and refining our models in realistic service and retail environments. The type of business utilized to extract contextual data is the restaurants.

3.4.2 Results

To ensure optimal performance of our model on a laptop equipped with a 6GB GPU and 16GB of RAM, without exceeding the GPU's memory limitations, we have made strategic configuration choices. We've set the mini-batch size to 32, allowing the model to process 32 sequences simultaneously during each training iteration, effectively managing GPU memory usage. Additionally, we've capped the maximum sequence length at 512; any input sequences exceeding this limit are truncated or divided into smaller segments to prevent memory overloads. The model's architecture features a single Bi-LSTM layer with a hidden size of 256, balancing computational complexity with the hardware's capabilities. Furthermore, we've configured the learning rate at 0.1, a crucial hyperparameter that determines the step size during training, tailored to the task and dataset requirements to facilitate effective model convergence.

To rigorously assess the performance of our custom Named Entity Recognition (NER) model, we employed three essential metrics: Precision, Recall, and F1 Score. Each metric provides a unique insight into the model's effectiveness:

- **Precision** measures the accuracy of the predictions, indicating the proportion of correctly identified contexts out of all contexts the model identified as such. High precision means that the model is reliable in its identification, with fewer false positives.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \qquad 3.4$$

- **Recall** assesses the model's ability to capture all relevant instances in the dataset. It reflects the proportion of actual contexts that the model correctly identified. A high recall indicates that the model is thorough, missing few actual contexts.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad 3.5$$

- **F1 Score** combines precision and recall into a single metric using the harmonic mean. It provides a balanced measure of the model's accuracy and thoroughness, making it especially valuable when comparing the performance of different models or when seeking a balance between precision and recall.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad 3.6$$

By analyzing these metrics, we can comprehensively evaluate the model's performance, understanding both its strengths and areas for improvement in identifying contexts accurately and completely across various texts.

As discussed previously, our methodology is distinctive as we have not encountered any existing studies that extract contextual data and then evaluate the outcomes for each specific context. Consequently, to demonstrate the effectiveness of our approach, we will compare the results of the two developed models: the first CustomNER and the second CustomNER. This comparison will help highlight the unique strengths and potential improvements of each model in handling contextual information.

Table 3.3 presents a comparative analysis of the performance of our two custom NER models. Despite the computational constraints that limited our training to only 10% of the data we collected, both models exhibited exceptional performance across various contexts. Specifically, in the Location context, while the improvement our second model achieved was modest compared to the first model, it was nonetheless significant. The customNER2 attained precision, recall, and F1 scores of 92.66%, 93.25%, and 93.12%, respectively, surpassing the CustomNER1 model which registered scores of 91.47%, 87.23%, and 90.70%. This indicates our models' enhanced ability to accurately identify and classify Location entities.

The performance disparity becomes more pronounced in the Weather context, where our second model significantly outperformed the first proposed model, achieving precision, recall, and F1 scores of 90.02%, 91.32%, and 91.50%, respectively, compared to customNER1' scores of 86.37%, 85.43%, and 85.90%. This substantial improvement underscores our models' superior capability in interpreting and categorizing Weather-related entities, a direct benefit of employing a corpus developed from Dbpedia.

Table 3.3: Comparision between our first and second custom NER.

	CustomNER1			**CustomNER2**		
	Precision	**Recall**	**F1 score**	**Precision**	**Recall**	**F1 score**
Time	77.74%	76.35%	77.04%	94.33%	92.56%	93.04%
Location	91.47%	87.23%	90.70%	92.66%	93.25%	93.12%
Companion	91.59%	92.15%	91.34%	92.16%	94.15%	93.34%
Weather	86.37%	85.43%	85.90%	90.02%	91.32%	91.50%

The most marked difference was observed in the Time context, where our second model demonstrated far superior precision, recall, and F1 scores of 94.33%, 92.56%, and 93.04%, respectively, in contrast to customNER1, which lagged significantly behind with scores of 77.74%, 76.35%, and 77.04%. This leap in performance highlights our models' enhanced ability to handle temporal information, likely benefitting from a more comprehensive and varied dataset that includes a broader range of temporal expressions.

While the improvements in the Location context were less dramatic compared to those in the Weather and Time contexts, the latter's substantial advancements suggest potential weaknesses in the customNER1 processing capabilities, possibly due to an inadequate training corpus. Remarkably, the high performance of our second model, achieved with only a fraction of the available data, not only demonstrates their efficiency and effectiveness but also underscores the significant impact of constructing a corpus from knowledge databases rather than relying solely on the volume of training data.

3.4.3 Discussion

Our research aims to address the gaps in contextual data collection by employing large-scale datasets from platforms like Amazon and Yelp, alongside sophisticated model evaluation techniques. Previous studies have often overlooked the comprehensive extraction of contextual information in these domains. To overcome this, our study leverages custom Natural Language Processing (NLP) models, specifically designed Named Entity Recognition (NER) models specialized in extracting contextual data for recommendation, which demonstrate remarkable efficiency in identifying and extracting relevant data points.

These NER models are particularly adept at recognizing and categorizing entities (contexts) related to weather, time, location and companion aspects within user reviews. This proficiency is reflected through enhanced metrics of precision, recall, and F1 scores, surpassing previous benchmarks in the field. The results achieved by the second cutom NER model is particularly noteworthy. It utilizes a vast corpus derived from open data sources such as Dbpedia, enabling more precise and accurate categorization of weather and time-related entities which was a challenge for the first proposed cutom NER, due to the lack of time and weather data within public corpora utilized for the training.

This approach not only fills a crucial research gap but also showcases the potential of integrating expansive open-source data in improving the granularity and accuracy of data extraction processes in NLP applications. By doing so, our study provides a robust framework for future research in contextual data collection, offering insights that could significantly enhance recommendation processes.

Our study is extensive but recognizes some limitations. Limited computational resources necessitated training our models on only a subset of the available data, which may have restricted their learning potential. Moreover, our models were trained with a specific focus on certain contexts and datasets, which might impact their ability to generalize to different situations. Despite these constraints, the successful integration of contextual data highlights the opportunity for further research. This includes broadening the models' training datasets and contexts to enhance their applicability and robustness across varied scenarios.

3.5 Conclusion

This chapter introduces a custom NER designed for extracting contextual data from user reviews. The model comprises three layers: an embedding layer, a Bi-LSTM layer, and a CRF layer. Its primary function is to input sequences of words and predict their contextual tags. Initial outcomes demonstrate proficiency in extracting certain contexts; however, the model struggles with specific contexts due to the nature of the corpora used for training. To address this issue, a second model was proposed that utilizes a personalized corpus specifically crafted for context-aware recommendation systems (RS). This corpus is constructed from data sourced from DBpedia, tailored to match predefined contextual dimensions. By training the custom NER with this specialized corpus, the model shows improved performance across all contexts. Additionally, enhancements were made to the initial model, particularly in overcoming the limitations of CRFs to enhance the overall effectiveness of the proposed model.

In the upcoming chapter, we will explore how to utilize the extracted contextual information to improve the accuracy of recommendations. Integrating this contextual data presents challenges, including issues related to data sparsity and increased dimensionality. To address these challenges, we propose two methods that employ advanced techniques to develop robust context-aware recommender systems.

Chapter 4

Rating prediction

In Chapter 3, we introduce models that extract contextual data from reviews, which is typically sparse and high-dimensional, making its incorporation a significant challenge. Recent research has shown that representing user context as a latent vector can effectively mitigate these issues. Models such as Factorization Machines (FM) have gained popularity for their ability to address sparsity and condense the feature space into a more manageable latent space. However, FM fall short in effectively distinguishing between different contexts. By applying a uniform latent space across all features, FMs fail to capture the subtle differences that specific contexts introduce to feature interactions. To overcome this limitation, we propose a new model, called Context-Based Factorization Machines (CBFM). CBFM improves on traditional FM by efficiently incorporating contextual data into the recommendation process. Despite its success, CBFM struggles with higher-order feature interactions due to its design focusing primarily on second-order interactions.

To address this issue, we introduce a second model that leverages deep learning techniques to manage non-linear feature interactions. This model extends beyond the capabilities of CBFM, offering a more flexible and powerful approach to capturing complex relationships in the data. This deep learning-based model is designed to handle the intricacies of higher-order interactions more effectively, providing a robust solution to the limitations observed in CBFM.

4.1 Introduction

Traditional Recommender Systems typically operate on two principal dimensions: users and items, to deliver recommendations. Introducing additional contextual data significantly

enhances the accuracy of these systems. However, the recommendation domain faces two primary challenges: sparsity and high dimensionality. Sparsity occurs due to limited interactions or ratings between users and items, while high dimensionality emerges from incorporating contextual data, thereby increasing the dataset's complexity. Ironically, adding contextual data can intensify sparsity by expanding the dataset's dimensions, complicating the recommendation process further.

FM is a supervised algorithm known for effectively addressing these issues and delivering notable performance. This method transforms user-item dimensions into latent spaces, creating a compact representation of the data. Much research in Context-Aware Recommender Systems (CARSs) focuses on enhancing FMs. A significant limitation of FM is its dependency on a fixed interaction function, typically an inner product, to predict high-order interactions between users and items. This method may fail to capture the detailed and varied dynamics of real-world user-item relationships.

High-order interactions involve three or more features. For instance, in a movie recommendation system, a second-order interaction might analyze how a movie's genre and director affect its rating, while a high-order interaction could explore how the genre, director, user age, and viewing time collectively influence the rating. Such interactions can offer deeper insights and boost the predictive accuracy of the model. However, effectively capturing and utilizing these complex interactions is challenging due to computational demands, data sparsity, and the potential for model overfitting. The computational cost rises exponentially with the interaction order, making it impractical for large datasets. This complexity is exacerbated by data sparsity, as higher-order feature combinations are rarely observed, complicating the learning process. Additionally, increased model complexity heightens the risk of overfitting, especially when the dataset is too small to support a model trained on numerous parameters, compromising the model’s generalizability and effectiveness. Moreover, FM uses a single latent vector even when handling features from different contextual dimensions. This approach may overlook the unique behaviors exhibited by features when they interact across various contexts, limiting FM's ability to discern the nuanced interactions that occur between features from different contexts.

Recent studies have emphasized integrating contextual data into recommender systems to enhance their performance. For example, Baltrunas et al. [152] proposed a context-aware model using Matrix Factorization, which efficiently captures the relationship between contexts and item ratings with minimal computational overhead. This approach allows for detailed granularity in depicting item-context interactions. On the other hand, Casillo et al. [153] introduced a CARS that incorporates contextual data directly within the model rather than treating it as an external input. This method uses the computational strengths of matrix factorization to maintain scalability. Despite the benefits, both Matrix Factorization and Factorization Machines face difficulties in addressing non-linear feature interactions during the learning process. Many researchers have explored the use of DL to develop sophisticated CARS, seeking to tackle the challenges of non-linear complexities inherent in recommendation models. For instance, Jeong et al. [154] developed a CARS utilizing DL methods. This model improves recommendation accuracy by incorporating contextual features. It combines a neural network with an auto-encoder, using well-known deep learning structures to extract unique features and predict scores while restoring input data. The model is also versatile, capable of handling various types of contextual information. Sattar et al. [155] developed a Contextual Graph Convolutional Matrix technique, which leverages the complex graph structures to incorporate user preferences and contextual signals represented by edges. This method utilizes a graph encoder to create detailed representations of users and items, factoring in contextual cues, inherent attributes, and user opinions. These representations are then aggregated and processed by a decoder to predict ratings. Vu et al. [156] presented a novel approach that utilizes DL to enhance context-aware multi-criteria recommender systems. In their method, DL are crucial for predicting contextual multi-criteria ratings and learning how to aggregate these ratings. The research successfully illustrates how DNNs models can be applied to Multi Criteria Recommender Systems (MCRS) to predict ratings in a context-aware manner and to understand the aggregation function in depth. Vaghari et al. [157] proposed a CARS framework designed to enhance the accuracy of machine learning models in detecting disease patterns. Their method integrates multi-modal data sources, combining genetic, clinical, and environmental information to maximize synergistic effects. This comprehensive approach not only boosts diagnostic precision but

also customizes treatments for individual patients, advancing personalized medicine. However, the model encounters challenges such as the high computational demand of processing large multimodal datasets and potential biases from incomplete or unrepresentative data. Overcoming these obstacles is essential for the framework's scalability and reliability in various clinical environments. Dilekh et al. [158] introduced a CARS tailored for smart homes, utilizing the FPGrowth algorithm and a Generalized Linear Method (GLM) to analyze and apply user behavior association rules. The study focused on a single-user scenario, demonstrating high accuracy and robustness, which attest to its effectiveness. However, the research also recognized certain limitations, notably its restriction to single-user settings, which might not address the complexities found in multi-user environments. Additionally, the system's adaptability to changing user behaviors and the balance between personalization and privacy remain underexplored. Further investigation is needed to integrate this system effectively into broader smart home ecosystems and ensure its compatibility with various devices and platforms. Previous studies predominantly used DL to create RS but did not sufficiently overcome issues like sparsity and high dimensionality in real-world datasets. Contrary to these approaches, our goal is to develop innovative context-aware recommender systems that addresses the limitations of FM by utilizing the power of Deep Neural Networks. Furthermore, we seek to reveal an advanced FM variant that is specially optimized for CARS.

The major contributions of the proposed models include:

- **New Variant of FM for CARSs**: We introduce a novel variant of Factorization Machines specifically designed for Context-Aware Recommender Systems (CARSs) called CBFM. This model is adept at discerning the variations across different contexts while also effectively capturing low-order feature interactions.
- **Hybrid Context-Aware Recommender Model**: A significant advancement is the development of a new context-aware recommender model that merges Deep Learning (DL) with Factorization Machines (FMs). This hybrid model leverages the strengths of both technologies to improve the performance of CARSs, incorporating

additional contextual information such as time, companion, and location to deliver personalized recommendations.

- **Utilization of DL Techniques**: Another key contribution is the application of deep learning techniques to model non-linear feature interactions. DL's ability to detect complex patterns in data helps overcome some of the limitations faced by traditional Factorization Machines, particularly in handling higher-order interactions.

4.2 Proposed models

Our primary objective is to create a sophisticated recommender system that goes beyond using conventional user and item data by incorporating contextual information. This allows for a more nuanced understanding of user behaviors in various scenarios. By adopting this comprehensive approach, we aim to provide personalized recommendations that are specifically tailored to the preferences and needs of individual users. To achieve this, we proposed two models: the first is Context-based Factorization Machines (CBFM) and the second, the Deep Context-Based Factorization Machines (DeepCBFM).

4.2.1 Context-based Factorization Machines (CBFM)

In the literature, numerous studies have utilized various methods to predict ratings. For instance, Linear Regression (LR) have been widely utilized [159][160] due to its simplicity which allows a straightforward interpretation and implementation, making it a popular choice in studies aiming to understand and predict user preferences based on a set of observable features. Linear Regression predicts ratings by calculating a linear combination of features. This relationship is typically represented by the following equation:

$$y_{\text{LR}} = w_0 + \sum_{i=1}^{d} w_i x_i \qquad 4.1$$

Where, y_{LR} represents the predicted output from the LR model. w_i are the coefficients for each feature x_i. d is the number of features. However, LR models typically struggle with accuracy because they do not incorporate the interactions among features, which are essential for

understanding complex relationships in the data. This limitation makes them less effective in scenarios where the interplay between variables significantly influences the outcomes.

To address this limitation, Poly2 [161] model which is an extension of traditional LR incorporates both primary feature effects and their pairwise interactions. This approach introduces second order feature interactions into the equation, enhancing the model's ability to capture complex relationships between features. It is particularly valuable when the relationships between variables are not merely additive but interactive, which can be crucial in many practical applications like predictive modeling, machine learning, and recommendation. Poly2 formula is defined as follows:

$$y_{\text{Poly2}} = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} x_i x_j w_{ij} \qquad 4.2$$

Where, y_{poly2} is the predicted rating, w_0 is the bais term and w_{ij} represents the coefficients for the interaction between feature i and j. This method, while effective, has notable limitations. Specifically, the interaction parameters between features can only be learned if those features co-occur in the same data record. This constraint means that predictions involving unseen features may be weak or insignificant, as the model lacks the necessary data to train these interaction terms effectively. This can limit the model's predictive power in scenarios where new or rare features are introduced.

Matrix Factorization (MF) model typically performs better than the Poly2 model, particularly in scenarios involving sparse data. This advantage arises because MF are specifically designed to effectively handle and interpret the sparse matrices commonly found in many real-world datasets. MF is a class of collaborative filtering algorithms used in recommendation systems [162]. Its goal is to fill in the missing entries of a user-item association matrix, typically representing some form of interaction such as ratings or clicks, based on the assumption that both items and users can be described by a smaller set of latent factors inferred from the observed interactions.

As shown in figure 4.1 [163], MF techniques decompose the original large and sparse user-item matrix into lower-dimensional representations in terms of latent factors associated with

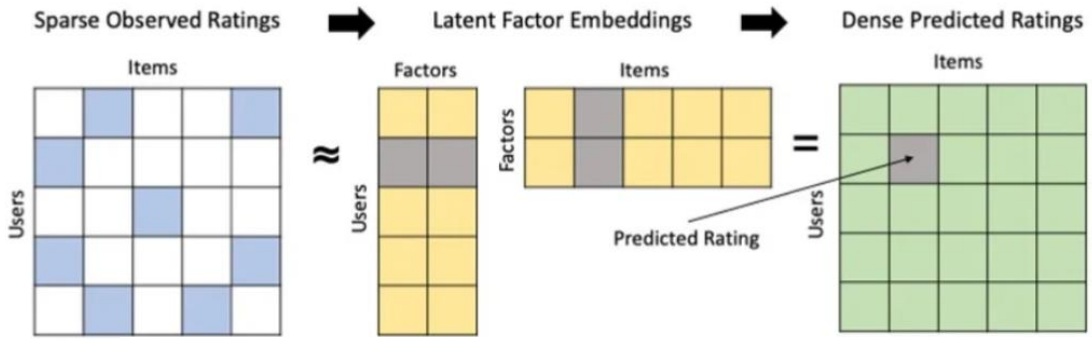

Figure 4.1: Matrix Factorization decomposition process.

users and items. Essentially, it transforms both items and users into the same latent factor space, thus enabling the prediction of missing entries through the interactions of these latent factors.

Mathematically, given a matrix R of dimensions $m \times n$, where, m is the number of users and n is the number of items, Matrix Factorization techniques aim to find two lower-dimensional matrices, U (of dimensions $m \times k$) and V (of dimensions $k \times n$), such that their product approximates R:

$$R \approx U \times V^T \qquad 4.3$$

Where, U represents the user matrix where each row is a k-dimensional latent vector corresponding to a user. V represents the item matrix where each column is a k-dimensional latent vector corresponding to an item. K is the number of latent factors and is typically much smaller than m or n.

MF effectively overcomes limitations of Poly2 and linear regression models, it also has its own set of drawbacks. Initially, MF was designed to handle two-dimensional data, primarily focusing on user-item interactions. Consequently, it lacks native support for incorporating additional types of data, such as contextual information, directly into the recommendation process. MF is mainly used in recommendation systems to decompose user-item matrices into latent factors, focusing primarily on user-item interactions. It excels in simplicity and effectiveness but is limited to matrix-like data and struggles with sparse matrices without auxiliary features.

Factorization Machines (FM) extend MF by modeling interactions between any features, not just user-item pairs. They can handle additional features like user demographics, item characteristics and contextual data, making them more flexible and applicable to a wider range of tasks, including regression and classification in high-dimensional sparse datasets. Figure 4.2 illustrates the transformation of recommendation data into a prediction problem using Sparse Feature Vector Representation from real-valued features. This representation converts user/item and additional data such as contextual information into a format suitable for predictive modeling, emphasizing how sparse data is structured and utilized to enhance the prediction accuracy in models that handle high-dimensional spaces.

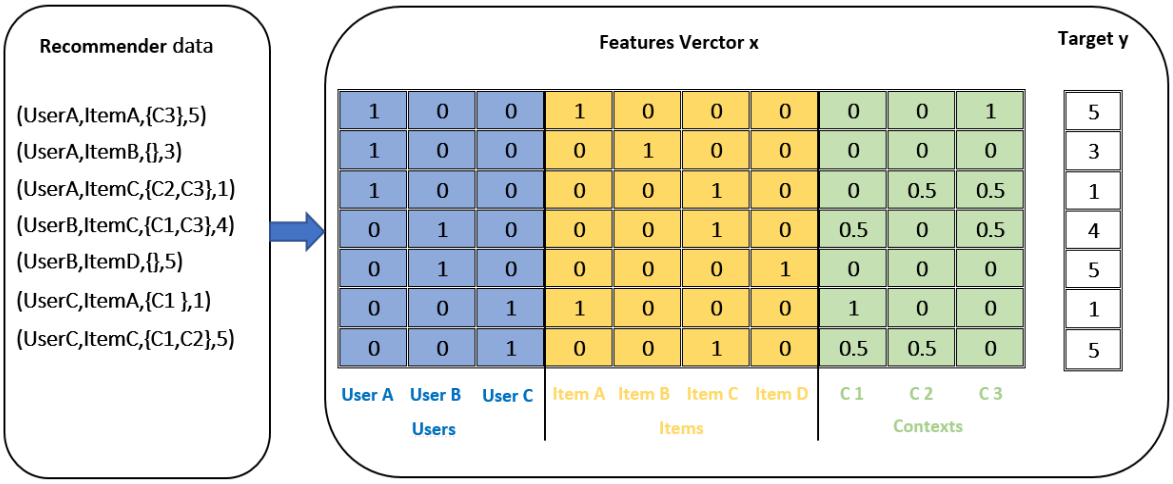

Figure 4.2: Data representation used by FM.

FM was first introduced by Rendle [164] as a sophisticated predictive modeling technique specifically designed to manage the complexities of high-dimensional sparse datasets where conventional methods often struggle. This model is distinctive in its ability to fully capture interactions between features through an elegant mathematical setup that utilizes latent factors. In this setup, each feature is associated with a latent vector, and the interactions between any two features are modeled as the dot product of these vectors. This design allows FMs to predict interactions effectively even in datasets where many feature combinations are rarely observed together. FM equation is denoted as follows:

$$\hat{y}_{FM}(x) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} \langle v_i, v_j \rangle x_i x_j \qquad 4.4$$

Where, w_0 is the global bias term, w_i are the linear interaction weights of each feature, v_i and v_j are the latent vectors corresponding to features i and j respectively. $\langle v_i, v_j \rangle$ represents the dot product of the latent vectors for features i and j

The versatility of FM is one of their key strengths, allowing them to be applied across a range of tasks involving real-valued features, such as regression, classification, and ranking. This adaptability is further supported by the model's capacity to handle both numerical and categorical data, typically encoded as binary vectors. Utilizing latent factors, FM significantly reduce the dimensionality of interaction spaces, effectively mitigating the computational challenges and the curse of dimensionality often associated with high-dimensional data.

FM excels in recommendation systems where they predict user preferences by leveraging past interactions and seamlessly integrate additional data. This integration is accomplished without complex feature engineering, enhancing the model's ability to provide tailored content and recommendations. The training of FMs usually involves gradient-based optimization techniques, such as stochastic gradient descent (SGD), which is favored for its effectiveness with large-scale data. To combat overfitting, particularly in sparse environments, regularization strategies like L2 regularization are commonly implemented. Since their inception, FM has been broadly utilized in various applications beyond just recommendation systems, including click-through rate prediction in digital advertising and intricate ranking operations in search technologies. Their widespread adoption underscores their effectiveness and resilience in managing intricate interaction models across diverse industrial settings.

FM employs a single latent vector for each feature to model the latent interactions across various contexts. For example, in scenarios involving three contexts such as Time, Companion, and Location, FM uses the same embedding vector for "Tuesday" to represent its latent effects in interactions with both "Companion" (e.g., $<V_{Tuesday}, V_{Friend}>$) "Location"

(e.g., <$V_{Tuesday}$, V_{AtHome}>), regardless of the distinct nature of these contexts. This method might not fully capture the complex behaviors that manifest when features interact in different situational contexts. To investigate this issue further, we apply ANOVA [165], a statistical technique pioneered by Ronald Fisher in the early 20th century. ANOVA is used to examine and highlight similarities or differences within specific attributes of a population by analyzing variances. ANOVA equation is:

$$F = \frac{mean\ sum\ of\ squares\ due\ to\ treatment}{mean\ sum\ of\ squares\ due\ to\ error} \qquad 4.5$$

Initially, we employ ANOVA to confirm that specific contexts behave differently when they interact with other distinct contexts. This application of ANOVA is crucial for quantifying the varying strengths of feature interactions within a dataset. Our experimental analysis is conducted on the DePaulMovie dataset, as detailed in Chapter 6. The results, presented in Figure 4.3, illustrate the outcomes from analyzing the interactions among three specific contexts: Time, Location, and Companion.

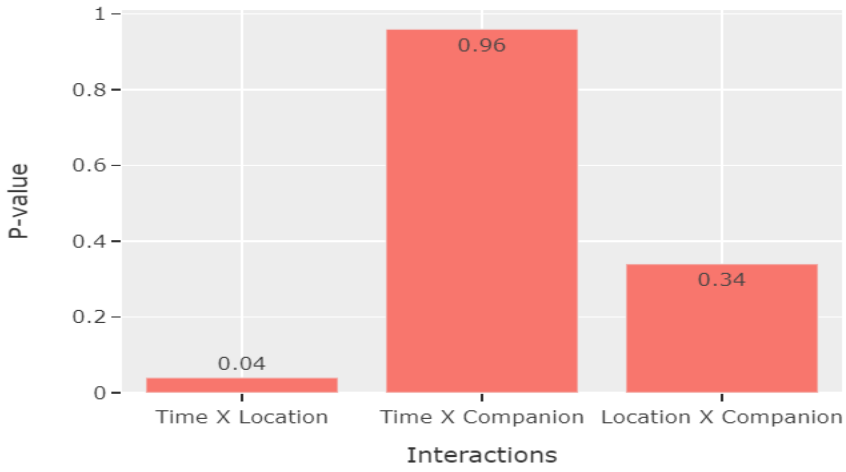

Figure 4.3: Context interactions.

Using two-way ANOVA, this statistical approach enables us to systematically evaluate the combined effects of pairs of these contexts *(Ci, Cj)* on a dependent variable *Y*. By doing so, we can identify significant interactions that affect the outcome, such as how the time of day and location together influence movie preferences when accompanied by different companions. This method not only highlights the individual contributions of each context but

also captures the complex interplay between them, thereby providing a deeper understanding of how contextual factors jointly impact user behavior.

In our analysis, the interaction between Time and Location demonstrated a significant effect with a P-value of 0.04, which is below the threshold of 0.05, indicating a statistically significant interaction between these two contexts. However, the interactions involving Time and Companion, as well as Location and Companion, yielded P-values higher than 0.05, suggesting that these context pairs do not exhibit significant interactions.

This variability in interaction patterns among different context pairs highlights the complexity of feature interactions in datasets. To address this complexity and the limitations of traditional Factorization Machines (FM), which might not adequately differentiate between varying contexts, we have developed an advanced model called Context-Based Factorization Machines (CBFM). This enhanced model introduces additional weights that help differentiate the effects of distinct contexts, allowing for a more nuanced representation of interactions. Specifically, CBFM adjusts the latent vectors according to the contexts they interact with, thereby better capturing the unique effects of different feature combinations.

The primary objective of the CBFM model is to accurately model second-order interactions while specifically accounting for contextual variations that affect these interactions. This refinement aims to provide a deeper understanding of the data by recognizing and quantifying the influence of contextual information on feature interactions, thereby improving the predictive performance of the model. The equation for CBFM, which encapsulates these enhancements, is formulated to include context-specific adjustments to the standard FM equation, ensuring a more tailored and effective approach to modeling complex datasets. CBFM equation is defined as follows:

$$\hat{y}_{CBFM}(x) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} \langle v_i \cdot wc(i), v_j \cdot wc(j) \rangle x_i x_j \qquad 4.6$$

Where, w_0 and w represent the bias term and the weights for the feature vectors, respectively. The vectors v_i and v_j are the latent vectors associated with features i and j, which capture the

underlying interactions between these features. The weights wc(i) and wc(j) are coefficients generated randomly to differentiate importance of interactions between contexts.

While the Context-based Factorization Machine (CBFM) has made significant strides in incorporating contextual data to improve model accuracy, it faces several limitations. Firstly, the model assigns additional weights somewhat arbitrarily to capture the importance of context interactions, rather than employing empirical values derived from analytical tools like ANOVA, which could more accurately measure the significance of feature interactions. Secondly, the inclusion of extra weights adds to the model's complexity, potentially making it more cumbersome to implement and optimize. Lastly, like traditional Factorization Machines, CBFM does not effectively model high-order interactions, which could be crucial in understanding more complex relational dynamics.

To address these issues, we propose a new model designed to overcome the shortcomings of the CBFM. The model is presented in more details in the next section.

4.2.2 Deep Context-based Factorization Machines (DeepCBFM)

The DeepCBFM combines Factorization Machines (FM) and deep learning techniques to overcome the limitations of its predecessor, the Collaborative Filtering Based Model (CBFM), thereby enhancing recommendation systems by incorporating contextual data. As shown in Figure 4.4, the architecture features two core components: the primary CBFM component, which is an improved variant of the first proposed model. It extends traditional FM to capture second-order interactions for a foundational understanding of user-item relationships, and the secondary deep component, designed to probe into more complex, higher-order feature interactions. Both components share common input and embedding layers, ensuring that data is uniformly processed into a dense vector format suitable for detailed analysis. The model’s efficacy culminates in the integration of outputs from both components, delivering personalized recommendations that finely cater to individual user preferences and needs, thereby achieving a holistic approach in modern recommendation systems.

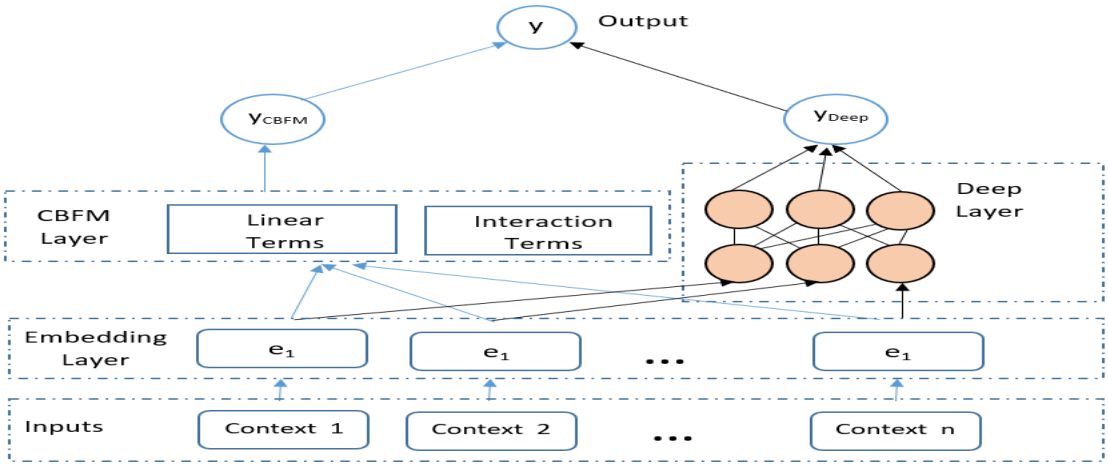

Figure 4.4: DeepCBFM architecture.

4.2.2.1 CBFM component

The primary objective of the CBFM component within our DeepCBFM framework is to effectively capture second-order context interactions. Originally introduced in our preliminary work, the CBFM now functions as an integral part of our comprehensive solution. This enhanced model addresses the limitations of its predecessor by overcoming the challenges associated with randomly generated weights that fail to accurately represent the significance of each feature interaction. Instead of using arbitrary weights, the revised CBFM component utilizes P-values generated by ANOVA to assign weights to the interactions, thereby accurately measuring the significance of each context interaction. This approach not only ensures more meaningful weight assignments but also simplifies the mathematical operations involved, significantly reducing the model's complexity. The redefined equation for the CBFM, incorporating these improvements, is structured to optimize both performance and efficiency in capturing contextual relationships. The output of the CBFM component is calculated as follows:

$$y_{CBFM}(x) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} S_{ij} \langle v_i, v_j \rangle x_i x_j \qquad 4.7$$

Where, w_0 is the bias term, w_i are the weights associated with each feature x_i. S_{ij}, derived from ANOVA, adjusts the impact of each interaction,

4.2.2.2 Deep component

In the initial phase of the deep component of a feedforward neural network, sparse input features are transformed into dense vectors via an embedding process. Given the original input vector's high sparsity and dimensionality, which includes a variety of data types such as categorical and continuous, organized by contextual groupings like time and weather, an embedding layer is essential. This layer compacts the input into a dense format, setting the stage for processing in subsequent hidden layers.

It is crucial to note that both the CBFM and deep components of the model utilize the same feature embeddings for two main reasons: The first reason is by employing identical embeddings, both components operate with a consistent interpretation of input features. This uniformity ensures that features are processed similarly throughout the model, enhancing the coherence of the learning process. The embeddings encapsulate the core attributes of each feature within a lower-dimensional space, maintaining the semantic relationships among them. This shared foundation allows the FM component to effectively capture second-order interactions and the deep component to handle complex, non-linear relationships. The second reason is that the sharing embeddings between the FM and deep components reduces the model's complexity. This centralization avoids duplicative mechanisms for converting raw features for each component, minimizing the number of parameters to be learned. Such consolidation not only conserves computational resources but also optimizes the training process by establishing a singular representation standard for context. The resulting embedding vectors are concatenated into a single dense vector, denoted as $S^{(0)}$, which forms the input to the neural network. This vector serves as a compact, lower-dimensional embodiment of the original input data, ensuring essential information is retained while reducing overall dimensionality.

$$S^{(0)} = [e_1, e_2, e_3, \dots, e_l] \qquad 4.8$$

Where each e_i is a dense vector representing the embedded form of an input feature (context). l is the feature' number.

The concatenated embeddings serve as input to the FNN, which comprises several layers. Each layer in this network is structured to progressively identify and learn more intricate patterns and interactions within the input features.

$$S^{(j)} = f\left(W^{(j)}s^{(j-1)} + b^{(j)}\right) \quad 4.9$$

Where, $S^{(j)}$ represents the output vector of the jth layer. $S^{(j-1)}$ Denotes the output vector from the previous layer (j – 1). $W^{(j)}$ the weight matrix associated with jth layer. $b^{(j)}$ The bias vector for the jth layer. f is the activation function applied to the linear combination of the inputs and weights.

The final result of DeepCBFM model is calculated by first summing the outputs from CBFM component and Deep component, then the result of this sum is then passed through a sigmoid function, which transforms the combined output into a probability value.

$$y = \sigma\left(y^{(CBFM)} + y^{(DNN)}\right) \quad 4.10$$

Where, y the final output or prediction of the DeepCBFM model. σ sigmoid function.

4.3 Experiments

This chapter is dedicated to assessing the effectiveness of our proposed models. We begin by detailing the datasets employed to train these models, offering a comprehensive overview of the empirical basis. Subsequently, we compare our models against several baseline models, conducting an in-depth analysis of their performance across different scenarios. This comparison not only elucidates the advantages and constraints of our methods but also situates them within the larger landscape of ongoing research in recommendation systems.

4.3.1 Datasets

The Amazon and Yelp datasets, as discussed in the previous chapter, are not only valuable for extracting contextual data from user reviews but are also crucial for evaluating the performance of ratings prediction models. These datasets include extensive user-generated reviews and ratings, which are instrumental in developing and testing predictive algorithms.

For more comprehensive testing, we incorporate a variety of additional datasets to evaluate our models. This diverse range of datasets enables us to conduct extensive experiments, providing a deeper and more nuanced analysis of the proposed models' performance across different scenarios and data types. This approach ensures that our findings are robust and our models are adaptable to various real-world applications. The incorporated additional datasets are from the CARSKit collection [166]. Specifically, we use two distinct datasets that enable an in-depth evaluation of contextual factors on user preferences:

- **DePaulMovie Dataset**: This dataset contains a total of 5,044 movie ratings, split into 1,449 non-contextual ratings and 3,595 contextual ratings that reflect specific viewing circumstances. It captures three primary contextual dimensions—Location (with the options of Home or Cinema), Companion (including Alone, Family, or Friends), and Time (Weekend or Weekday). These dimensions allow for a detailed analysis of how different viewing environments and social settings influence movie watching experiences, providing valuable insights into contextual effects on user behavior.
- **InCarMusic Dataset**: This dataset consists of 4,012 music ratings from 43 users over 138 unique music items. It includes 1,004 non-contextual ratings and 3,010 contextual ratings, which are influenced by the user's environment and specific situations during music listening. The dataset features seven varied contextual dimensions: Driving Style (Relaxed or Sport driving), Landscape (Coastline, Countryside, Mountains, or Urban), Mood (Active, Happy, Lazy, or Sad), Natural Phenomena (Afternoon, Daytime, Morning, or Night), Road Type (City, Highway, or Serpentine), Sleepiness (Awake or Sleepy), and Weather (Cloudy, Rainy, Snowing, or Sunny). Each of these factors is crucial for understanding the complex interplay between the user's context and their music preferences, allowing for nuanced model evaluations and the development of more adaptive recommendation systems. These contextual dimensions in the InCarMusic dataset offer a comprehensive framework for investigating how different driving environments influence music preferences. This detailed analysis enables a deeper understanding of user behavior under various

driving conditions, which is crucial for tailoring music recommendations in dynamic settings.

In RS, data is conventionally organized into a rating matrix format where users are represented by columns, items by rows, and the observed ratings fill the matrix cells. A common challenge with this setup is the significant sparsity due to many items not being rated by most users. To address the complex needs of context-aware recommendations, which involve multiple dimensions, the traditional matrix format is often replaced by tensors, allowing for a richer representation of data dimensions. However, when implementing the CBFM and DeepCBFM models, the data must be formatted as Sparse Feature Vectors, commonly utilized in deep neural network models and known as One Hot Encoding. This method enables the efficient handling of categorical data by converting it into a binary format. For effective data management and manipulation, we leverage the Pandas library [167], which offers a comprehensive suite of tools for data analysis, cleaning, exploration, and manipulation. The Sparse Feature Vector format transforms the recommendation data into tuples of (x, y), where 'x' represents a real-valued feature vector and 'y' is the observed rating. This setup redefines the recommendation task as a typical machine learning prediction problem, facilitating the application of standard machine learning algorithms. Additionally, this format easily integrates additional contextual dimensions, significantly enhancing the model's ability to manage and analyze complex, multi-dimensional datasets effectively. Table 4.1 provides detailed insights into the structured data of the two datasets.

Table 4.1: Statistics about the two datasets.

	DePaulMovie	InCarMusic
Users	124	43
Items	80	138
Ratings	5044	4014
Dimensions	4	7
Sparsity	94%	99%

4.3.2 Results

To rigorously assess the effectiveness of our models, the evaluation process is structured into two distinct phases. In the first phase, we utilize the DePaulMovie and InCarMusic datasets to conduct an initial evaluation of both the CBFM and DeepCBFM models. This phase aims to determine the performance of each model under controlled conditions using well-defined, context-rich datasets.

Following this preliminary assessment, the second phase involves deploying the superior model determined from the first phase on datasets that were originally used to extract contextual data. This strategic approach allows us to not only test the model in a setting similar to its training environment but also to verify its robustness and adaptability across different data contexts.

By employing this two-step evaluation strategy, we can generate more comprehensive and realistic results. It enables a thorough analysis of the models' capabilities in handling diverse datasets and contexts, thereby providing a clearer understanding of their practical applicability and effectiveness in real-world scenarios. This methodical approach helps in identifying potential areas of improvement and in understanding the models' strengths and limitations in various data environments.

To evaluate the performance of our proposed models, we employ two fundamental statistical metrics: Root Mean Squared Error (RMSE) and the R-Squared (R^2). RMSE is calculated as the square root of the Mean Squared Error (MSE), which itself is determined by averaging the squared differences between the predicted values ($\hat{y}_i$) and the actual target values (y_i). This metric is expressed mathematically as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad 4.11$$

where n is the number of observations. RMSE is particularly useful because it provides a measure of how accurately the model predicts the target variable, with lower values indicating better performance.

On the other hand, R-Squared (R^2) is a measure of the proportion of the variance in the dependent variable that is predictable from the independent variables, calculated using the formula:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \qquad 4.12$$

where $\bar{y}$ is the mean of the actual values. This metric helps assess the explanatory power of the model, with higher values indicating a better fit to the observed data. Together, these metrics provide a comprehensive evaluation of a model's accuracy and effectiveness in fitting and predicting data.

To validate the effectiveness of the CBFM and DeepCBFM models, we compare them against five established baseline models in the field of recommendation systems:

- **FM [164]**: This well-known technique is extensively used in Collaborative Filtering systems. It operates by decomposing the rating matrix into a pair of low-rank matrices through a specialized transformation process, effectively capturing interactions between variables in a low-dimensional space.
- **CBMF [153]**: CBMF enhances traditional Matrix Factorization by incorporating contextual information into the model, allowing it to adapt recommendations based on specific situational factors.
- **TCAFM [127]**: TCFM is model based on FM to incorporate contextual data to enhance recommendation quality.
- **CANCF [168]**: This hybrid model enhances recommendation systems by integrating a prefiltering approach that includes contextual data, providing a more nuanced understanding of user preferences.
- **CAMCRS [154]**: CAMCRS leverages deep learning techniques to predict ratings while taking into account contextual factors. It also focuses on effectively aggregating this information to improve recommendation accuracy.

By comparing CBFM and DeepCBFM with these models, we aim to demonstrate their respective strengths and advancements in handling contextual data within recommendation systems.

For the development of our models, we utilize TensorFlow [169], a robust framework for deep learning, to implement CBFM and DeepCBFM. The computational environment consists of a Windows 10 computer equipped with 16GB of RAM, and our model implementation draws inspiration from [170]. We organize each dataset by splitting it into training data (80%) and testing data (20%), ensuring a balanced approach to training and validation. For optimization, we employ the Adam optimizer, setting a low learning rate of 0.00001 to ensure gradual and steady convergence. To handle large volumes of data efficiently, we process it in mini-batches of 4096. To counteract the risk of overfitting, L2 regularization is applied, a common technique that introduces a penalty proportional to the magnitude of the coefficients. We specifically fine-tune two parameters to optimize the performance. First, we adjust the embedding size. As indicated in figure 4.5, the RMSE achieves its minimum at an embedding size of 32. This size is found to be optimal as larger embeddings enhance the model's representation capacity, providing a richer and more detailed feature set for predictions.

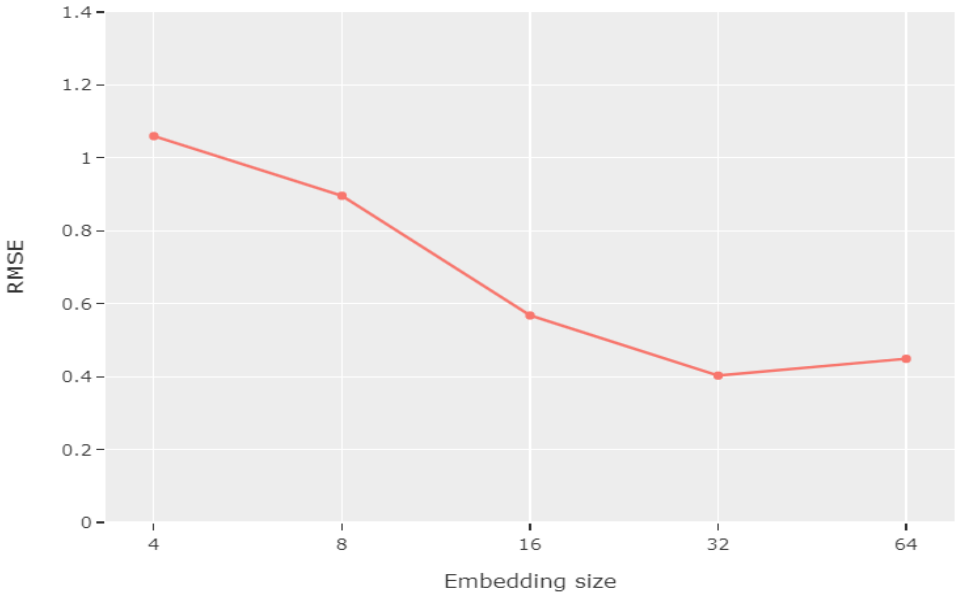

Figure 4.5: RMSE results of DeepCBFM under the embedding size parameter.

Secondly, we scrutinize the dropout rate, a technique designed to improve model generalization by randomly omitting a subset of features at each iteration during training. This method effectively reduces overfitting by preventing complex co-adaptations on training data. Figure 4.6 demonstrates how we systematically adjusted the dropout rate in our

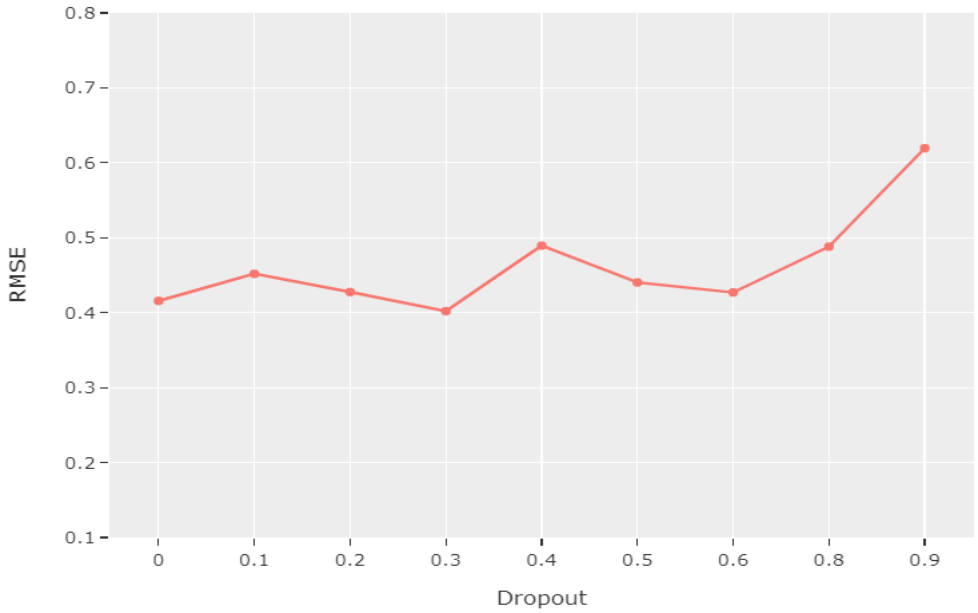

Figure 4.6: RMSE results of DeepCBFM under the dropout parameter.

DeepCBFM model, testing values ranging from 0 to 0.9 in increments of 0.1. This analysis reveals that the model achieves its optimal performance, indicated by the lowest RMSE, when the dropout rate is set at 0.3. This finding underscores the efficacy of the dropout technique in enhancing model generalization and preventing overfitting.

In our study, we also conducted a thorough comparative analysis, pitting the CBFM and DeepCBFM models against five other baseline models across two distinct datasets. The performance of these models was evaluated using the RMSE and R^2 metrics, which provided a quantitative measure of their prediction accuracy and the proportion of variance explained, respectively.

Particular attention was given to the comparison between the CBFM and the traditional FM algorithm. This focused analysis helped to highlight the advantages of integrating contextual information into the model, showcasing how the CBFM approach enhances prediction

accuracy by leveraging additional contextual data. This comparative evaluation not only validated the superiority of our model over conventional methods but also illustrated the practical benefits of our advancements in handling complex data interactions within recommendation systems.

The data presented in Figures 4.7 and 4.8 demonstrate the effectiveness and predictive accuracy of various models using the DePaulMovie dataset. Specifically, the DeepCBFM model achieves the lowest RMSE value at 0.4027, indicating that its predictions are most closely aligned with the actual movie ratings. This low RMSE highlights DeepCBFM's exceptional ability to reduce prediction errors, making it highly effective in providing precise movie rating estimations compared to the other models evaluated. This superior performance underscores DeepCBFM's robustness in predicting user preferences within the dataset. Additionally, the enhanced performance of the CBFM model compared to the traditional FM model provides concrete evidence of its effectiveness. Furthermore, the CBFM model excels beyond variations of the FM model such TCAFM and CBMF, showcasing its robustness in handling the dataset.

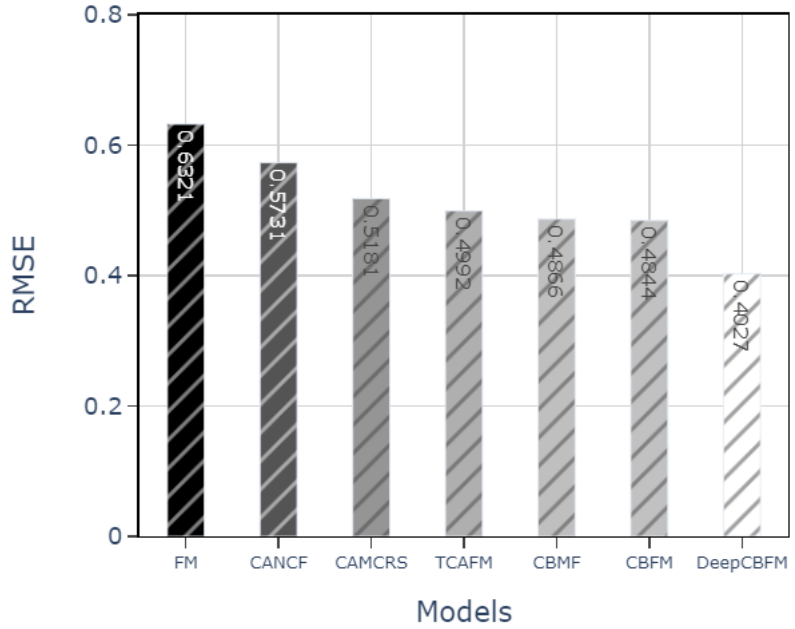

Figure 4.7: RMSE results obtained for DePaulMovie dataset.

Regarding the R^2 metric, which assesses the fit quality of the models, the superiority of the DeepCBFM model becomes increasingly evident. This model not only outstrips the CBFM which the second-best model and traditional FM model in R^2 values but also demonstrates its capacity to account for a greater portion of variance in movie ratings. This indicates that the DeepCBFM model effectively identifies more complex patterns and underlying relationships in the dataset, enhancing its predictive accuracy for movie ratings. The significant lead of the DeepCBFM model over the CBFM and FM and the rest of models in

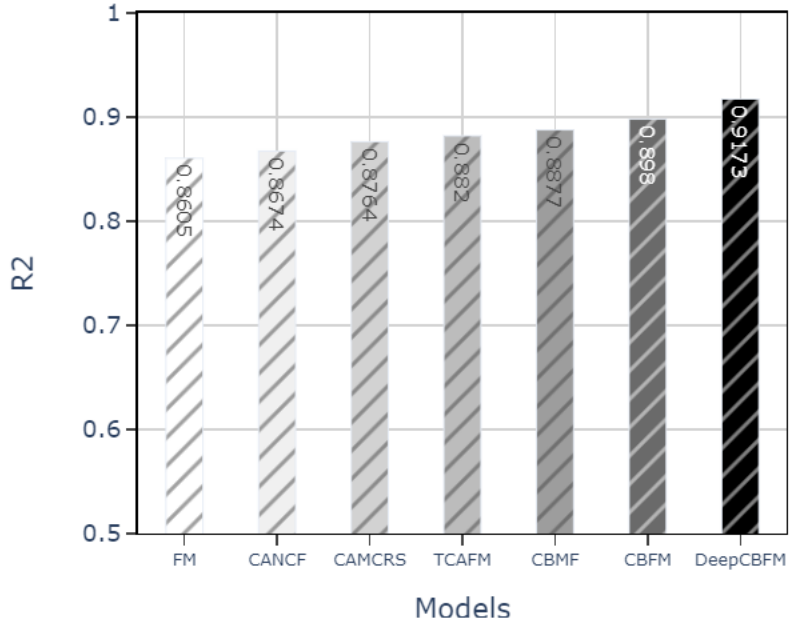

Figure 4.8: R^2 results obtained for DePaulMovie dataset.

terms of R^2 further highlights its advanced capability in capturing the intricate dynamics of movie preferences.

The analysis of the InCarMusic dataset, as shown in Figures 4.9 and 4.10, corroborates the performance rankings observed in the initial dataset, maintaining a consistent order among the models. It is noteworthy that the FM model shows a decline in its performance, which can be attributed to the dataset's inherently high contextual dimensions that challenge simpler models. A notable point is the close performance between the TCAFM, CBMF and CBFM models. This similarity likely stems from their common design emphasis on capturing low-order feature interactions, which suggests a shared strength in handling specific types of data relationships. These findings underscore the outstanding performance of the DeepCBFM

model and highlight the critical role of integrating contextual information and utilizing sophisticated modeling approaches, such as deep learning architectures, in recommendation systems.

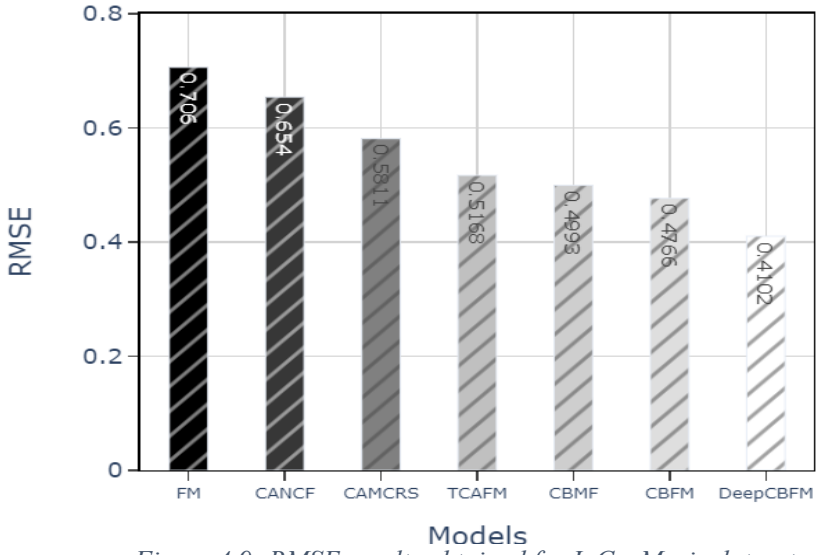

Figure 4.9: RMSE results obtained for InCarMusic dataset.

By incorporating these advanced techniques, recommendation models can more effectively discern the complex and varied factors that influence user preferences. This enhanced capability not only improves the precision of movie recommendations but also significantly boosts their relevance and user satisfaction.

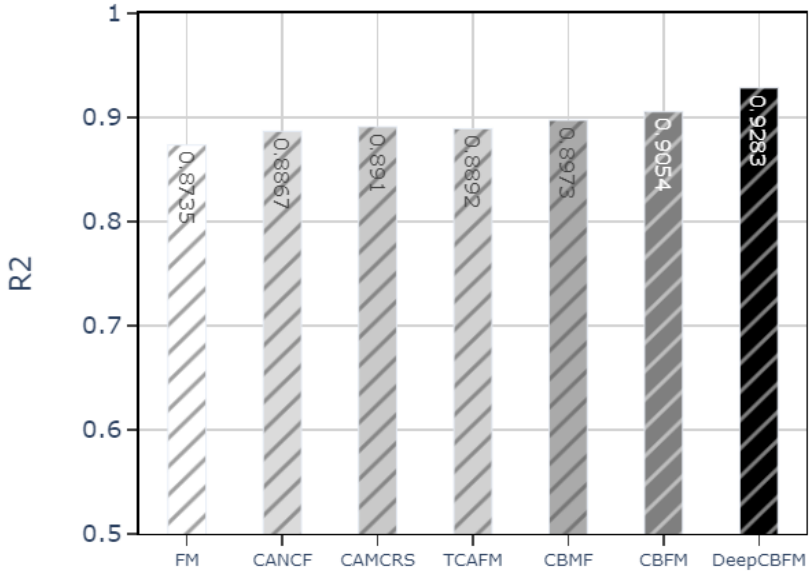

Figure 4.10: R2 results obtained for InCarMusic dataset

To further assess our proposed models, we utilize the Yelp and Amazon datasets, previously utilized to extract contextual information from the reviews contained within these datasets. We employ the same baseline models for evaluating the performance of the CBFM and DeepCBFM models. This approach allows us to consistently compare the effectiveness of our models across different datasets and contexts.

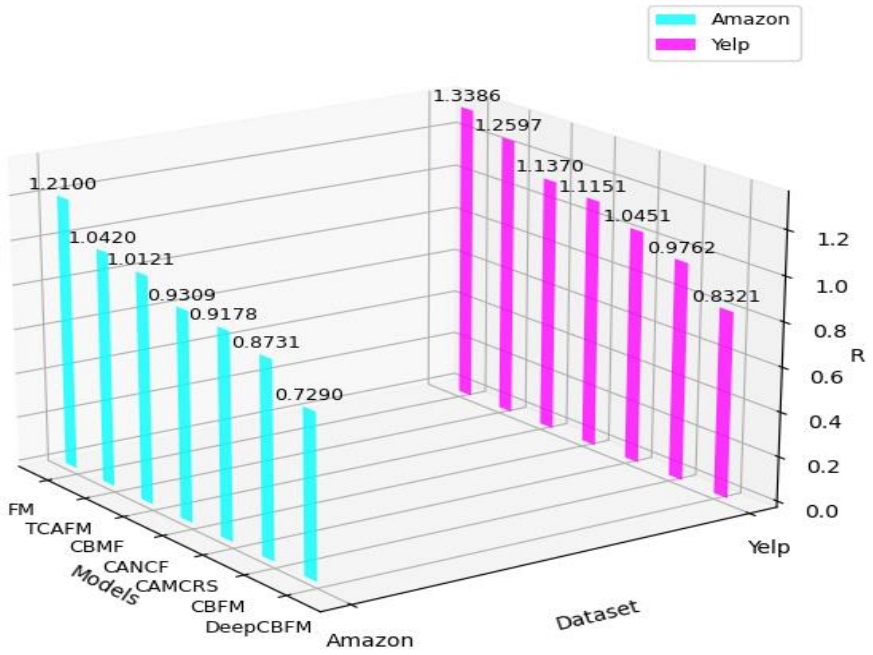

Figure 4.11: RMSE results obtained for Amazon and Yelp datasets.

Figure 4.11 (Cyan color) vividly displays the enhancements in recommendation quality afforded by the new model. This model significantly reduces the RMSE to 0.729, outperforming the CBFM model, which has the next best score at 0.8731. This comparison underscores the superior accuracy of our model, particularly in handling complex datasets enriched with high-order features, where deep learning models typically excel. Extending this analysis to the Yelp dataset, Figure 4.11 (color magenta) reaffirms the new model's effectiveness, as observed with the Amazon dataset. Here, the DeepCBFM model again records the lowest RMSE, scoring 0.8321, with CBFM following. This performance highlights the model's robustness across different datasets. Additionally, the FM model is noted for its higher RMSE, which underscores its limitations in managing complex feature interactions and differentiating contexts effectively. This comparative analysis clearly

establishes the advantages of leveraging advanced, deep learning-based approaches in improving recommendation systems.

Figure 4.12 presents the comparative results of the DeepCBFM model, with and without the incorporation of contextual information. The data clearly demonstrates that the model achieves superior performance when it utilizes contextual data from both the Amazon and Yelp datasets. This enhancement in performance is consistent across the experiments, reinforcing the observation that contextual information plays a crucial role in augmenting the effectiveness of recommender systems. The integration of contextual data not only improves the predictive accuracy but also significantly enhances the overall quality of the recommendations provided by the system. This evidence underscores the value of incorporating rich contextual details to achieve more precise and relevant recommendations in such systems.

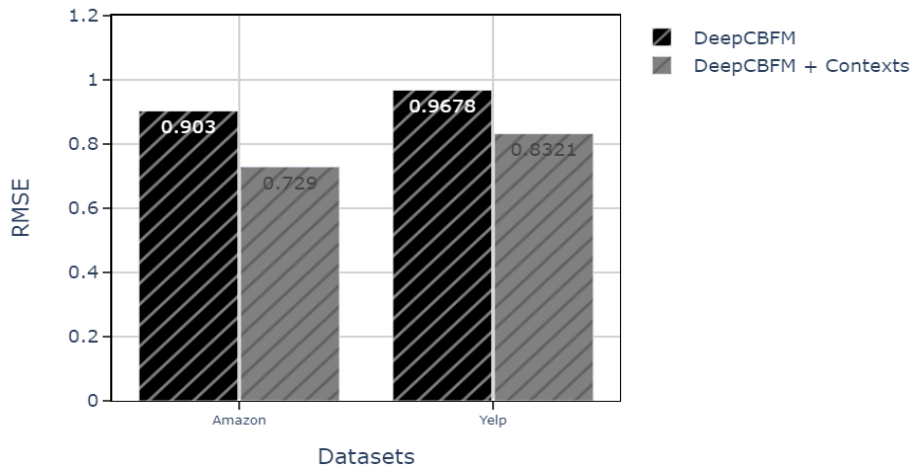

Figure 4.12: Comparison between DeepCBFM with and without using contextual data.

4.3.3 Discussion

This study introduces groundbreaking methods aimed at boosting the efficacy of CARS through the use of a new variant of factorization machines and a deep learning-integrated variant of the same algorithm, specifically named Context-Based Factorization Machine (CBFM) and its more advanced counterpart, DeepCBFM. These models deviate significantly from traditional recommendation algorithms by excelling in capturing the complex interactions among various features within different contextual settings. This capability is

essential, as it facilitates more personalized and precise recommendations by understanding how features interact differently across diverse contexts.

Our research highlights that combining deep learning with factorization machines substantially enhances the handling of datasets with intricate contextual dimensions. The improvements are particularly evident when comparing the performance of DeepCBFM against traditional FM methods, showcasing the older models' limitations in managing the complexities found in modern recommendation systems. Furthermore, the comparison between the CBMF and CBFM models reveals a significant insight; while these models perform similarly in terms of low-order feature interactions, CBFM gains performance by deploying mechanisms that capture the significant of interactions.

This progression is critically important in the field of RS, where accurately predicting user preferences amidst complex contexts can significantly enhance user satisfaction and engagement. Our findings suggest a major shift towards incorporating sophisticated deep learning elements into recommendation models, aiming to fully leverage the spectrum of feature interactions. This shift could potentially lead to transformative improvements in the way RS comprehend and meet user needs, thereby boosting overall effectiveness and enhancing user experience.

4.4 Conclusion

In this chapter, we introduced two approaches aimed at enhancing personalized recommendation systems by integrating contextual data. The initial method is a variant of Factorization Machines known as CBFM, was designed to address certain shortcomings of traditional FM. Upon evaluation, the CBFM model demonstrated limitations, notably its inability to model nonlinear interactions between contextual features and its reliance on randomly generated weights, which may not accurately reflect the true significance of feature interactions. Additionally, the incorporation of extra weights increases the model's complexity, complicating both its implementation and optimization especially when dealing with high order feature interactions. Given these challenges, we proposed a second, more robust model that leverages Deep Learning alongside an enhanced version of CBFM, named

DeepCBFM. This model aims to significantly improve the capabilities of our recommender system. The experiments conducted in this chapter have provided significant insights into the performance and scalability of our proposed models under various conditions. Through rigorous testing across multiple datasets, including those sourced from CARSkits, Amazon and Yelp, we have systematically evaluated the efficacy of our novel algorithms compared to existing standards. Key findings from these experiments demonstrate that the advanced features integrated into our models, for such as contextual data extraction, incorporation and analysis, enhanced by deep learning techniques, contribute markedly to improving accuracy and processing efficiency. The results clearly show our models' superiority in handling complex datasets, with improved precision and recall in context extraction tasks and more accurate rating predictions as evidenced by lower RMSE scores.

Furthermore, these experiments have highlighted the importance of robust data preprocessing and the impact of model tuning on overall performance. By optimizing parameters and incorporating adaptive learning rates, we achieved significant gains in model responsiveness and predictive power.

Conclusion and future work

This chapter concludes our study, reviewed the research questions initially posed at the beginning of this thesis and the solutions we have developed in response. Additionally, it highlights the principal contributions of our research and proposes potential avenues for future work

Conclusion

In the opening chapter, we explored the domain of context-aware recommender systems and identified existing shortcomings within this field. Building on this analysis, we formulated the following research questions:

Q1. How can contextual data be gathered indirectly or inferred from user interactions rather than directly solicited from users and What are the potential sources from which contextual data can be extracted without compromising user privacy?

Q2. How can we develop effective models that collect contextual data and require minimal manual input in terms of feature engineering, particularly those that utilize publicly available data?

Q3. How can contextual information be seamlessly integrated into recommender systems while addressing issues like data sparsity and high dimensionality?

Q4. Does the contextual data collected enhance the quality of recommendations?

To address Q1, we need a data source that fulfills specific criteria: it must contain contextual information and be user-generated. Upon reviewing available data sources, user reviews emerge as the most suitable choice for our requirements. User reviews are detailed evaluations or feedback from users regarding a product, service, or experience. These reviews generally consist of a narrative description of the user's experience, coupled with a numerical

rating reflecting their level of satisfaction. User reviews are essential for aiding potential customers in their decision-making processes and have a considerable impact on a business's reputation and success. In practice, upon analyzing the user reviews, we found that approximately 10% of them include contextual information. This subset of reviews proves to be a crucial resource, offering a significant enhancement to our dataset and addressing the existing shortfall in contextual data.

Chapter 3 answers Q2, which focuses on determining efficient and low-effort methods to process the identified data source. In this chapter, we introduced two methodologies employing a custom Named Entity Recognition (NER) system, specifically designed to capture contextual information. Notably, the enhanced performance of the second model is attributed to its training on a substantial, specialized corpus derived from Dbpedia. This extensive corpus has equipped the model with the necessary capabilities to effectively discern contextual information within user reviews. The results from our experiments underscore the effectiveness of these custom NER models. Their ability to extract contextual data from user reviews is evidenced by their outstanding performance metrics, including high precision, recall, and F1 scores. These outcomes highlight the models' adeptness in identifying and processing relevant contextual information, which significantly contributes to the depth and utility of the data analysis.

To answer the Q3, two models have been developed namely CBFM and DeepCBFM as described in chapter 4. While the CBFM model represented a variant of the traditional Factorization machines that is designed to incorporate effectively contextual data and tackle FM short comes. The DeepCBFM exploited deep learning capabilities to provide a more sophisticated model that not only integrate contexts efficiently and addresses the difficulties of FM to process non-linear features, but also improve predication accuracy. The results demonstrated significant enhancements in predictive accuracy compared with baseline models.

The response to the final research question is discussed in Section 3.4 and 4.3, where we presented the outcomes of experiments comparing the performance of the DeepCBFM model with and without the incorporation of contextual information. The results clearly show a

substantial decrease in RMSE for the model utilizing contextual data compared to its counterpart lacking this enhancement. These findings confirm the substantial benefits of integrating advanced modeling techniques and contextual data within predictive frameworks. This integration not only significantly improves the accuracy of predictions but also demonstrates the value of contextual information in refining the performance of computational models.

Future work

The field of context-aware recommender systems represents an active area that of many opportunities for future research. When we look to our proposed models which cover the contextual information collection and rating prediction, while these models demonstrated strong performance, we must acknowledge certain limitations, particularly the constraints of computational resources. These constraints limited our ability to fully train our models on the extensive datasets available, which likely capped their learning capabilities. Future investigations should focus on harnessing more powerful computational resources to enable comprehensive training on larger datasets. This will help in evaluating the true potential of our models and allow us to explore their scalability more effectively. Enhancing computational capacity could lead to more nuanced models that can handle larger and more complex data sets, providing richer insights and more accurate recommendations. Moreover, broadening the contextual frameworks by testing and validating these models across diverse environments and user interactions will help assess their generalizability and identify necessary context-specific modifications to maintain their effectiveness.

Additionally, Continued efforts are needed to refine the efficiency and adaptability of our models. This includes optimizing algorithmic efficiency and enhancing the models' ability to adapt to different datasets and contextual nuances. Research should explore algorithmic improvements that reduce resource consumption while maintaining or increasing predictive accuracy. Moreover, developing techniques that allow models to quickly adapt to new datasets without extensive retraining could significantly enhance their practical application. Another direction for further exploration is extending the use of user reviews to infer implicit

ratings through techniques like sentiment analysis, which could mitigate data sparsity issues and enhance the predictive capabilities of our systems. Advancements in context-sensitive computational models are also essential, focusing on developing sophisticated models that dynamically adjust to users' changing preferences and behaviors.

Finally, our proposed models have demonstrated versatility across various domains, as evidenced by their successful testing with datasets from diverse fields including music, movie, product, and restaurant recommendations. Extending these models to sectors such as healthcare, finance, or e-commerce could greatly improve personalized and context-aware recommendations, significantly boosting user engagement and satisfaction across these industries.

Publications

In this section, we have listed our published articles. All models proposed in this thesis were initially published in international journals or as international conferences papers.

International journals

The list of articles published in international journals is presented as follows:

- R. Madani, A. Ez-Zahout, and F. Omary, “Extracting contextual insights from user reviews for recommender systems: a novel method,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 35, no. 1, Art. no. 1, Jul. 2024, doi: 10.11591/ijeecs.v35.i1.pp542-550.
- R. Madani, A. Ez-Zahout, F. Omary, and A. Chedmi, “Advancing Context-Aware Recommender Systems: A Deep Context-Based Factorization Machines Approach,” International Journal of Computing and Digital Systems, vol. 16, no. 1, pp. 353–363, Apr. 2024, doi: 10.12785/ijcds/160128.
- R. Madani and A. Ez-zahout, “A Review-based Context-Aware Recommender Systems: Using Custom NER and Factorization Machines,” International Journal of Advanced Computer Science and Applications (IJACSA), vol. 13, no. 3, Art. no. 3,

42/30 2022, doi: 10.14569/IJACSA.2022.0130365.

- A. Ez-zahout, H. Gueddah, A. Nasry, R. Madani, and F. Omary, “A hybrid big data movies recommendation model based k-nearest neighbors and matrix factorization,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 26, no. 1, Art. no. 1, Apr. 2022, doi: 10.11591/ijeecs.v26.i1.pp434-441.

International conferences

The list of articles published in international conferences are:

- R. Madani, A. Idrissi, and A. Ez-Zahout, “A New Context-Based Factorization Machines for Context-Aware Recommender Systems,” in Modern Artificial Intelligence and Data Science: Tools, Techniques and Systems, A. Idrissi, Ed., Cham: Springer Nature Switzerland, 2023, pp. 15–23. doi: 10.1007/978-3-031-33309-5_2.
- R. MADANI, A. EZ-ZAHOUT, and A. IDRISSI, “An Overview of Recommender Systems in the Context of Smart Cities,” in 2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech), Nov. 2020, pp. 1–9. doi: 10.1109/CloudTech49835.2020.9365877.

References

[1] Resnick, P. and Varian, H. R. (1997). Recommender systems. Communications of the ACM, 40(3):56–58.

[2] Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to recommender systems handbook. In Recommender Systems Handbook, pages 1–35. Springer.

[3] C. Basu, H. Hirsh, et W. Cohen,« Recommendation as classification: using social and content-based information in recommendation», in Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence, USA, juill. 1998, p. 714-720.

[4] Resnick, N. Iacovou, M. Suchak, P. Bergstrom, et J. Riedl, « GroupLens: an open architecture for collaborative filtering of netnews », in Proceedings of the 1994 ACM conference on Computer supported cooperative work, New York, NY, USA, oct. 1994, p. 175-186. doi:10.1145/192844.192905.

[5] G. Adomavicius, B. Mobasher, F. Ricci, et A. Tuzhilin, «Context-Aware Recommender Systems», AI Magazine, vol. 32, no 3, Art. no 3, 2011, doi: 10.1609/aimag.v32i3.2364.

[6] R. Andhikaputra, S. M. A. Tumbel, J. Vida, A. Gui, I. G. M. Karmawan, and Y. Ganesan, “User’s awareness of personal data leakage in E-commerce application,” E3S Web Conf., vol. 426, p. 02063, 2023, doi: 10.1051/e3sconf/202342602063.

[7] R. Fatima, A. Yasin, L. Liu, J. Wang, W. Afzal, and A. Yasin, “Sharing information online rationally: An observation of user privacy concerns and awareness using serious game,” Journal of Information Security and Applications, vol. 48, p. 102351, Oct. 2019, doi: 10.1016/j.jisa.2019.06.007.

[8] D. Malandrino, A. Petta, V. Scarano, L. Serra, R. Spinelli, and B. Krishnamurthy, “Privacy awareness about information leakage: who knows what about me?” in Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society, in WPES ’13. New York, NY, USA: Association for Computing Machinery, Nov. 2013, pp. 279–284. doi: 10.1145/2517840.2517868.

[9] Khusro, S., Ali, Z., & Ullah, I. (2016). Recommender systems: Issues, challenges, and research opportunities. Information Science and Computing.

[10] Ziegler, C.-N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving recommendation lists through topic diversification. Proceedings of the 14th international conference on World Wide Web.
[11] Mooney, R.J., & Roy, L. (2000). Content-based book recommending using learning for text categorization. Proceedings of the fifth ACM conference on Digital libraries.
[12] Pazzani, M. J. and Billsus, D. (2007). Content-based recommendation systems. In The adaptive web, pages 325–341. Springer.
[13] Adomavicius, G., & Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734-749.
[14] Nguyen, T. T., Hui, P. M., Harper, F. M., Terveen, L., & Konstan, J. A. (2014). Exploring the filter bubble: the effect of using recommender systems on content diversity. Proceedings of the 23rd international conference on World wide web.
[15] Cantador, I., Bellogín, A., & Castells, P. (2015). Ontology-based personalised and context-aware recommendations of news items. IEEE Transactions on Systems, Man, and Cybernetics: Systems.
[16] Salton, G. (1989). Automatic Text Processing: The Transformation, Analysis, and Retrieval of. Addison-Wesley.
[17] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, Dec. 1992, doi: 10.1145/138859.138867.
[18] T. Saracevic, A. Spink, and M.-M. Wu, "Users and Intermediaries in Information Retrieval: What Are They Talking About?," in *User Modeling*, A. Jameson, C. Paris, and C. Tasso, Eds., Vienna: Springer, 1997, pp. 43–54. doi: 10.1007/978-3-7091-2670-7_6.
[19] Y. Koren, S. Rendle, and R. Bell, "Advances in collaborative filtering," Recommender systems handbook, 2021.
[20] Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, pages 43–52. Morgan Kaufmann Publishers Inc.
[21] Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999). An Algorithmic Framework for Performing Collaborative Filtering. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (pp. 230-237)
[22] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-Based Collaborative Filtering Recommendation Algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295). ACM
[23] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994).cGrouplens: an open architecture for collaborative filtering of netnews. IncProceedings of the 1994 ACM conference on Computer supported cooperative work, pages 175–186. ACM. Knowledge-Based approach
[24] Koren, Y. and Bell, R. (2011). Advances in collaborative filtering. In Recommender Systems Handbook, pages 145–186. Springer.
[25] Shani, G., Brafman, R. I., and Heckerman, D. (2002). An mdp-based recommender system. In Proceedings of the Eighteenth conference on Un- certainty in artificial intelligence, pages 453–460. Morgan Kaufmann Publishers Inc.
[26] Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory.

[27] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2000). Application of dimensionality reduction in recommender system-a case study. Technical report, DTIC Document.
[28] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. Computer, 42(8), 30-37.
[29] Burke, R. (2000). Knowledge-based Recommender Systems. Encyclopedia of Library and Information Systems, 69(Supplement 32), 180-200.
[30] Bridge, D., Goeker, M., McGinty, L., Smyth, B.: Case-based recommender systems. Knowledge Engineering Review 20(3), 315–320 (2005)
[31] Felfernig, A., Burke, R.: Constraint-based recommender systems: technologies and research issues. In: 10th International Conference on Electronic Commerce, ICEC08, pp. 1-10. ACM, New York, NY, USA (2008)
[32] Reilly, J., McCarthy, K., McGinty, L., Smyth, B.: Dynamic Critiquing. In: 7th European Conference on Case-based Reasoning, ECCBR 2004, pp. 763–777. Madrid, Spain (2004)
[33] Tsang, E.: Foundations of Constraint Satisfaction. Academic Press, London (1993)
[34] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4), 331-370. DOI: 10.1023/A:1021240730564.
[35] Luce, R. D. (1959). Individual Choice Behavior: A Theoretical Analysis. New York: Wiley.
[36] Kahneman, D., & Tversky, A. (1979). Prospect Theory: An Analysis of Decision under Risk. Econometrica, 47(2), 263-291.
[37] R., and Raiffa, H. Decision with Multiple Objectives: Preferences and Value Tradeoffs. John Willey & Sons, New York, 1976.
[38] B. Mobasher, H. Dai, T. Luo, M. Nakagawa, Using sequential and non-sequential patterns in predictive web usage mining tasks, in Proceedings of IEEE International Conference on Data Mining, ICDM'02 (2002), pp. 669–672.
[39] R. Ragno, C.J.C. Burges, C. Herley, Inferring similarity between music objects with application to playlist generation, in Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval, MIR'05 (2005), pp. 73–80
[40] G. Shani, D. Heckerman, R.I. Brafman, An MDP-based recommender system. J. Mach. Learn. Res. 6, 1265–1295 (2005)
[41] G. Bonnin, D. Jannach, Automated generation of music playlists: survey and experiments. ACM Comput. Surv. 47(2), 26:1–26:35 (2014)
[42] S. Chen, J.L. Moore, D. Turnbull, T. Joachims, Playlist prediction via metric embedding, in Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'12 (2012), pp. 714–722
[43] F. Garcin, C. Dimitrakakis, B. Faltings, Personalized news recommendation with context trees, in Proceedings of the ACM Confererence on Recommender Systems, RecSys'13 (2013), pp. 105–112
[44] N. Hariri, B. Mobasher, R. Burke, Context-awaremusic recommendation based on latent topic sequential patterns, in Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys'12 (2012), pp. 131–131
[45]. D. Jannach, M. Jugovac, L. Lerche, Supporting the design of machine learning workflows with a recommendation system. ACM Trans. Interact. Intell. Syst. 6(1), 1–35 (2016)
[46] D. Jannach, L. Lerche, M. Jugovac, Adaptation and evaluation of recommendations for shortterm shopping goals, in Proceedings of the ACM Conference on Recommender Systems, RecSys'15 (2015), pp. 211–218.
[47] G.-E. Yap, X.-L. Li, P.S. Yu, Effective next-items recommendation via personalized sequential pattern mining, in Proceedings International Conference on Database Systems for

Advanced Applications, DASFAA'12 (2012), pp. 48–64
[48] Han, J., Pei, J., & Kamber, M. (2011). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann.
[49] R. Agrawal, T. Imieli´nski, A. Swami, Mining association rules between sets of items in large databases, in Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (1993), pp. 207–216
[50] B. Hidasi, M. Quadrana, A. Karatzoglou, D. Tikk, Parallel recurrent neural network architectures for feature-rich session-based recommendations, in Proceedings ACM Conference on Recommender Systems, RecSys'16 (2016), pp. 241–248
[51] M. Ludewig, D. Jannach, Evaluation of session-based recommendation algorithms. User-Model. User-Adapt. Interact. 28(4–5), 331–390 (2018)
[52] N.R. Mabroukeh, C.I. Ezeife, A taxonomy of sequential pattern mining algorithms. ACM Comput. Surv. 43(1), 1–41 (2010)
[53] He, X. et al., 2017. Neural collaborative filtering. Proceedings of the 26th International Conference on World Wide Web
[54] Hidasi, B. et al., 2015. Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939
[55] Oord, A. van den et al., 2013. Deep content-based music recommendation. Advances in Neural Information Processing Systems
[56] Vaswani, A. et al., 2017. Attention is all you need. Advances in neural information processing systems.
[57] Tintarev, N., Masthoff, J.: A survey of explanations in recommender systems. In: 2007 IEEE 23rd International Conference on Data Engineering Workshop, pp. 801–810, April 2007
[58] J. Chen, W. Geyer, C. Dugan, M.G.I. Muller, I. Guy, Make new friends, but keep the old: recommending people on social networking sites, in Proceedings of the International Conference on Computer-Human Interaction (CHI) (2009)
[59] M. Fazel-Zarandi, H.J. Devlin, Y. Huang, N. Contractor, Expert recommendation based on social drivers, social network analysis, and semantic data representation, in Proceedings of the Second International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec) (ACM, New York, 2011), pp. 41–48
[60] L. Brozovsky, V. Petricek, Recommender system for online dating service, in Proceedings of Znalosti Conference, Ostrava (2007)
[61] L. Pizzato, T. Rej, T. Chung, I. Koprinska, and J. Kay, "RECON: a reciprocal recommender for online dating," in Proceedings of the fourth ACM conference on Recommender systems, in RecSys '10. New York, NY, USA: Association for Computing Machinery, Sep. 2010, pp. 207–214. doi: 10.1145/1864708.1864747.
[62] J. Akehurst, I. Koprinska, K. Yacef, L. Pizzato, J. Kay, T. Rej, CCR: A content-collaborative reciprocal recommender for online dating, in Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI), Barcelona (2011)
[63] S. Bull, J.E. Greer, G.I. McCalla, L. Kettel, J. Bowes, User modelling in I-help: what, why, when and how, in Proceedings of the International Conference on User Modeling (2001), pp. 117–126
[64] C.-T. Li, Mentor-spotting: Recommending expert mentors to mentees for live trouble-shooting in codementor, in Knowledge and Information Systems, vol. 61 (Springer, Berlin, 2020), pp.799–820
[65] J. Malinowski, T. Keim, O. Wendt, T. Weitzel, Matching people and jobs: A bilateral recommendation approach, in Proceedings of the 39th Annual Hawaii International

Conference on System Sciences (2006)
[66] W. Hong, L. Lei, T. Li, W. Pan, iHR: An online recruiting system for xiamen talent service center, in Proceedings of the 19th ACM SIGKDD International
[67] P.-Y. Chen, Y.-C. Chou, and R. J. Kauffman, "Community-Based Recommender Systems: Analyzing Business Models from a Systems Operator's Perspective," presented at the 2009 42nd Hawaii International Conference on System Sciences, IEEE Computer Society, Dec. 2009, pp. 1–10. doi: 10.1109/HICSS.2009.630.
[68] S. K. Gorripati and V. K. Vatsavayi, "A Community Based Content Recommender Systems," vol. 12, no. 22, 2017.
[69] B. D. Deebak and F. Al-Turjman, "A novel community-based trust aware recommender systems for big data cloud service networks," Sustainable Cities and Society, vol. 61, p. 102274, Oct. 2020, doi:10.1016/j.scs.2020.102274.
[70] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning (MIT Press, 2016). https://www.deeplearningbook.org
[71] G. CANBULUT, "Analysis of The Travelling Time According to Weather Conditions Using Machine Learning Algorithms," Oct. 11, 2023. doi: 10.21203/rs.3.rs-3407758/v1.
[72] D. D. Sanandaj and S. H. Alizadeh, "A hybrid recommender system using multi layer perceptron neural network," in 2018 8th Conference of AI & Robotics and 10th RoboCup Iranopen International Symposium (IRANOPEN), Apr. 2018, pp. 7–13. doi: 10.1109/RIOS.2018.8406624.
[73] A. K. Yengikand, M. Meghdadi, S. Ahmadian, S. M. J. Jalali, A. Khosravi, and S. Nahavandi, "Deep Representation Learning using Multilayer Perceptron and Stacked Autoencoder for Recommendation Systems," in 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Oct. 2021, pp. 2485–2491. doi: 10.1109/SMC52423.2021.9658978.
[74] A. Angadi, S. Keerthi Gorripati, V. Rachapudi, Y. Krishna Kuppili, and P. Dileep, "Image-based Content Recommendation System with CNN," in 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Nov. 2021, pp. 1260–1264. doi: 10.1109/I-SMAC52330.2021.9641026.
[75] R. Patel, P. Thakkar, and V. Ukani, "CNNRec: Convolutional Neural Network based recommender systems - A survey," Engineering Applications of Artificial Intelligence, vol. 133, p. 108062, Jul. 2024, doi: 10.1016/j.engappai.2024.108062.
[76] K. V. Dudekula et al., "Convolutional Neural Network-Based Personalized Program Recommendation System for Smart Television Users," Sustainability, vol. 15, no. 3, Art. no. 3, Jan. 2023, doi: 10.3390/su15032206.
[78] N. X. Bach, D. H. Long, and T. M. Phuong, "Recurrent convolutional networks for session-based recommendations,"Neurocomputing,vol. 411, pp. 247–258, Oct. 2020, doi: 10.1016/j.neucom.2020.06.077.
[79] R. Mesuga and B. J. Bayanay, "A Deep Transfer Learning Approach on Identifying Glitch Wave-form in Gravitational Wave Data," May 09, 2022, arXiv: arXiv:2107.01863. doi: 10.48550/arXiv.2107.01863.
[80] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). "Attention is All You Need." In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017).
[81] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805.
[82] Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). "Deep Learning Based Recommender System: A Survey and New Perspectives." ACM Computing Surveys (CSUR), 52(1), 1-38.

[83] Choi, K., Lee, H., & Provost, F. (2020). "Scalable and Efficient Recommendation with Scalable Collaborative Filtering." IEEE Transactions on Knowledge and Data Engineering, 32(4), 650-662.

[84] D. Charte, F. Charte, M. J. del Jesus, and F. Herrera, "An analysis on the use of autoencoders for representation learning: Fundamentals, learning task case studies, explainability and challenges," Neurocomputing, vol. 404, pp. 93–107, Sep. 2020, doi: 10.1016/j.neucom.2020.04.057.

[85] C. C. Aggarwal, J. L. Wolf, K.-L. Wu, and P. S. Yu, "Horting hatches an egg: a new graph-theoretic approach to collaborative filtering," in Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, San Diego California USA: ACM, Aug. 1999, pp. 201–212. doi: 10.1145/312129.312230.

[86] Katz, L.: A new status index derived from sociometric analysis. Psychometrika 18(1), 39–43 (1953)

[87] Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 30(1-7), 107-117.

[88] Fouss, F., Pirotte, A., Renders, J. M., & Saerens, M. (2007). Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Transactions on Knowledge and Data Engineering, 19(3), 355-369

[89] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep Learning Based Recommender System: A Survey and New Perspectives," ACM Comput. Surv., vol. 52, no. 1, p. 5:1-5:38, Feb. 2019, doi: 10.1145/3285029.

[90] U. Panniello, A. Tuzhilin, M. Gorgoglione, C. Palmisano, and A. Pedone, "Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems," in Proceedings of the third ACM conference on Recommender systems, in RecSys '09. New York, NY, USA: Association for Computing Machinery, Oct. 2009, pp. 265–268. doi: 10.1145/1639714.1639764.

[91] K. Rama, P. Kumar, and B. Bhasker, "Deep Learning to Address Candidate Generation and Cold Start Challenges in Recommender Systems: A Research Survey." arXiv, Jul. 17, 2019. doi: 10.48550/arXiv.1907.08674.

[92] Y. Li, Q. Wu, L. Hou, and J. Li, "Entity knowledge transfer-oriented dual-target cross-domain recommendations," Expert Systems with Applications, vol. 195, p. 116591, Jun. 2022, doi: 10.1016/j.eswa.2022.116591.

[93] H. Oh and C. Kim, "Fairness-aware recommendation with meta learning," Sci Rep, vol. 14, no. 1, p. 10125, May 2024, doi: 10.1038/s41598-024-60808-x.

[94] J. Xu, Y. Li, Y. Jiang, and S.-T. Xia, "Adversarial Defense Via Local Flatness Regularization," in 2020 IEEE International Conference on Image Processing (ICIP), Oct. 2020, pp. 2196–2200. doi: 10.1109/ICIP40778.2020.9191346.

[95] C. Palmisano, A. Tuzhilin, and M. Gorgoglione, "Using Context to Improve Predictive Modeling of Customers in Personalization Applications," IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 11, pp. 1535–1549, Nov. 2008, doi: 10.1109/TKDE.2008.110.

[96] B. L. van Kortenhof and L. van Kortenhof, "Context-Aware Recommender Systems in the E-commerce Domain," 2017. Accessed: May 21, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Context-Aware-Recommender-Systems-in-the-Ecommerce-Kortenhof-Kortenhof/f9d2d0b4029da195faa9073b7edd962ee5028bea

[97] L. Baltrunas, B. Ludwig, S. Peer, F. Ricci, Context-aware places of interest recommendations for mobile users, in Design, User Experience, and Usability. Theory,

Methods, Tools and Practice, ed. by A. Marcus. Lecture Notes in Computer Science, vol. 6769 (Springer, Berlin, 2011), pp. 531–540

[98] F. Cena, L. Console, C. Gena, A. Goy, G. Levi, S. Modeo, I. Torre, Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide. AI Commun. 19(4), 369–384 (2006)

[99] K. Oku, S. Nakajima, J. Miyazaki, S. Uemura, Context-aware SVM for context-dependent information recommendation, in Proceedings of the 7th International Conference on Mobile Data Management (2006), p. 109

[100] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, S. Wang, Learning graph-based poi embedding for location-based recommendation, in Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (2016), pp. 15–24

[101] A. Karatzoglou, X. Amatriain, L. Baltrunas, N. Oliver, Multiverse recommendation: Ndimensional tensor factorization for context-aware collaborative filtering, in Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10 (ACM, New York, 2010), pp. 79–86

[102] A. Karatzoglou, L. Baltrunas, K. Church,M. Böhmer, Climbing the app wall: enabling mobile app discovery through context-aware recommendations, in Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12 (ACM, New York, 2012), pp. 2527–2530

[103] Q. Wang, H. Yin, T. Chen, Z. Huang, H. Wang, Y. Zhao, N.Q.V. Hung, Net point-of-interest recommendation on resource-constrained mobile devices, in Proceedings of The Web Conference 2020 (2020), pp. 906–916

[104] R. Cai, C. Zhang, C. Wang, L. Zhang, W.-Y. Ma, Musicsense: contextual music recommendation using emotional allocation modeling, in Proceedings of the 15th International Conference on Multimedia, MULTIMEDIA '07 (ACM, New York, 2007), pp. 553–5

[105] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM, New York, 2008), pp. 426–434

[106] L. Baltrunas, X. Amatriain, Towards time-dependant recommendation based on implicit feedback, in Workshop on Context-Aware Recommender Systems (CARS 2009), New York (2009)

[107] E. Alpaydin, Introduction to Machine Learning (The MIT Press, London, 2004) behavior in recommendation, in Workshop on Context-Aware Recommender Systems (CARS 2009), New York (2009)

[108] S. Lombardi, S. Anand, et M. Gorgoglione, « Context and customer behaviour in recommendation », oct. 2009. Consulté le: 26 novembre 2024. [En ligne]. Disponible sur: https://www.semanticscholar.org/paper/Context-and-customer-behaviour-in-recommendation-Lombardi-Anand/7ce834607412a69f0d188019556ca4094f6ad129

[109] V. Codina, F. Ricci, L. Ceccaroni, Exploiting the semantic similarity of contextual situations for pre-filtering recommendation, in User Modeling, Adaptation, and Personalization, ed. By S. Carberry, S. Weibelzahl, A. Micarelli, G. Semeraro. Lecture Notes in Computer Science, vol. 7899 (Springer, Berlin, 2013), pp. 165–177

[110] U. Panniello and M. Gorgoglione, "Incorporating context into recommender systems: an empirical comparison of context-based approaches," Electron Commer Res, vol. 12, no. 1, pp. 1–30, Mar. 2012, doi: 10.1007/s10660-012-9087-7.

[111] U. Panniello, A. Tuzhilin, M. Gorgoglione, Comparing context-aware recommender

systems in terms of accuracy and diversity. User Model. User-Adapt. Interact. 24(1–2), 35–65 (2014)
[112] Z. Yu, X. Zhou, D. Zhang, C.Y. Chin, X. Wang, J. Men, Supporting context-aware media recommendations for smart phones. IEEE Pervasive Comput. 5(3), 68–75 (2006)
[113] S. Abbar, M. Bouzeghoub, S. Lopez, Context-aware recommender systems: a service-oriented approach, in VLDB PersDB Workshop (2009)
[114] N. Hariri, B. Mobasher, R. Burke, Query-driven context aware recommendation, in Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13 (ACM, NewYork, 2013), pp. 9–16
[115] E. Frolov, I. Oseledets, Tensor methods and recommender systems. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 7(3), e1201 (2017)
[116] A. Rettinger, H. Wermser, Y. Huang, V. Tresp, Context-aware tensor decomposition for relation prediction in social networks. Soc. Netw. Anal. Min. 2(4), 373–385 (2012)
[117] H. Chen, J. Li, Adversarial tensor factorization for context-aware recommendation, in Proceedings of the 13th ACM Conference on Recommender Systems (2019), pp. 363–367
[118] W. Wu, J. Zhao, C. Zhang, F. Meng, Z. Zhang, Y. Zhang, Q. Sun, Improving performance of tensor-based context-aware recommenders using bias tensor factorization with context feature auto-encoding. Knowl. Based Syst. 128, 71–77 (2017)
[119] Z. Batmaz, A. Yurekli, A. Bilge, C. Kaleli, A review on deep learning for recommender systems: challenges and remedies. Artif. Intell. Rev. 52(1), 1–37 (2019)
[120] X. Xin, B. Chen, X. He, D. Wang, Y. Ding, J. Jose, CFM: convolutional factorization machines for context-aware recommendation, in Proceedings of the 28th International Joint Conference on Artificial Intelligence (AAAI Press, Palo Alto, 2019), pp. 3926–3932
[121] M. Unger, A. Tuzhilin, A. Livne, Context-aware recommendations based on deep learning frameworks. ACM Trans. Manag. Inf. Syst. 11(2), 1–15 (2020)
[122] L. Mei, P. Ren, Z. Chen, L. Nie, J.Ma, J.-Y. Nie, An attentive interaction network for contextaware recommendations, in Proceedings of the 27th ACM International Conference on Information and KnowledgeManagement, CIKM '18 (Association for ComputingMachinery, New York, 2018), pp. 157–166
[123] X. Ding, J. Tang, T. Liu, C. Xu, Y. Zhang, F. Shi, Q. Jiang, D. Shen, Infer implicit contexts in real-time online-to-offline recommendation, in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19 (Association for Computing Machinery, New York, 2019), pp. 2336–2346.
[124] A. Beutel, P. Covington, S. Jain, C. Xu, J. Li, V. Gatto, Ed.H. Chi, Latent cross: making use of context in recurrent recommender systems, in Proceedings of the Eleventh ACMInternational Conference on Web Search and Data Mining (2018), pp. 46–54.
[125] Q. Liu, S. Wu, D. Wang, Z. Li, L. Wang, Context-aware sequential recommendation, in 2016 IEEE 16th International Conference on Data Mining (ICDM) (IEEE, Piscataway, 2016), pp. 1053–1058
[126] P. Campos, N. Rodr´ıguez-Artigot, and I. Cantador, "Extracting Context Data from User Reviews for Recommendation: A Linked Data Approach," 2017.
[127] F. Z. Lahlou, H. Benbrahim, and I. Kassou, "Review Aware Recommender System: Using Reviews for Context Aware Recommendation,"IJDAI, vol.
[128] Oard, D. W., Kim, J., et al. (1998). Implicit feedback for recommender systems. In Proceedings of the AAAI workshop on recommender systems, pages 81–83. Wollongong.
[129] A. Levi, O. Mokryn, C. Diot, and N. Taft,"Finding a needle in a haystack of reviews: cold start context-based hotel recommender system," in Proceedings of the sixth ACM

conference on Recommender systems, New York, NY, USA, Sep. 2012, pp. 115–122. doi: 10.1145/2365952.2365977.

[130] L. Zheng, V. Noroozi, and P. S. Yu, "Joint Deep Modeling of Users and Items Using Reviews for Recommendation," arXiv:1701.04783 [cs], Jan. 2017, Accessed: Nov. 11, 2021. [Online]. Available: http://arxiv.org/abs/ 1701.04783

[131] R. Catherine and W. Cohen, "TransNets: Learning to Transform for Recommendation," in Proceedings of the Eleventh ACM Conference on Recommender Systems, New York, NY, USA, Aug. 2017, pp. 288–296. doi: 10.1145/3109859.3109878.

[132] J. McAuley et J. Leskovec, « Hidden factors and hidden topics: understanding rating dimensions with review text », in Proceedings of the 7th ACM conference on Recommender systems, in RecSys '13. New York, NY, USA: Association for Computing Machinery, oct. 2013, p. 165-172. doi: 10.1145/2507157.2507163

[133] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, et S. Ma, « Explici factor models for explainable recommendation based on phrase-level sentiment analysis », in Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, in SIGIR '14. New York, NY, USA: Association for Computing Machinery, juill. 2014, p. 83-92. doi: 10.1145/2600428.2609579.

[134] S. Aciar, "Mining Context Information from Consumer's Reviews. "https:www.semanticscholar.org/paper/Mining-Context-Information-from-Consumer%E2%80%99-s-Aciar/3c0977e396849c345ff5283772ecd5c1b7097c08.

[135] N. Hariri, B. Mobasher, R. Burke, and Y. Zheng, "Context-Aware Recommendation Based On Review Mining," 2011.

[136] Chen, H., Finin, T. and Joshi, A. 2005. The SOUPA ontology for pervasive computing. In Tamma, V., Cranefield, S., Finin, T. W., and Willmott, S. (Eds.), Ontologies for Agents: Theory and Experiences, pp. 233–258.

[137] Castelli, G., Rossi, A., Mamei, M., and Zambonelli, F. 2007. A simple model and infrastructure for context-aware browsing of the world. In Proceedings of the 5th IEEE Conference on Pervasive Computing and Communications, 1–11.

[138] Chaari, T., Dejene, E., Laforest, F., and Scuturici, V.-M. 2007. A comprehensive approach to model and use context for adapting applications in pervasive environments. International Journal of Systems and Software 80(12): 1973–1992.

[139] Kim, J. D., Son, J., and Baik, D.K. 2012. CA5W1H onto: Ontological context-aware model based on 5W1H. International Journal of Distributed Sensor Networks, article ID 247346, 11.

[140] L. F. Rau, "Extracting Company Names from Text," in Proceedings of the Seventh IEEE Conference on Artificial Intelligence Applications, Miami, FL, USA, 1991, pp. 29-32.

[141] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, June 2019, pp. 4171-4186.

[142] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, pp. 1532-1543.

[143] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in Proceedings of the International Conference on Learning Representations (ICLR), Scottsdale, AZ, USA, 2013.

[144] M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, November 1997.
[145] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in Proceedings of the 18th International Conference on Machine Learning (ICML), Williamstown, MA, USA, 2001, pp. 282-289.
[146] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellman, S., Morsey, M., van Kleef, P., Auer, S. and Bizer, C. 2015. DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web 6(2): 167–195.
[147] J. Pérez, M. Arenas, and C. Gutierrez, "Semantics and Complexity of SPARQL," in The Semantic Web - ISWC 2006, I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. M. Aroyo, Eds., Berlin, Heidelberg: Springer, 2006, pp. 30–43. doi: 10.1007/11926078_3.
[148] E. F. T. K. Sang and F. De Meulder, "Introduction to the CoNLL- 2003 Shared Task: Language Independent Named Entity Recognition," arXiv:cs/0306050, Jun. 2003, Accessed: Nov. 20, 2021. [Online]. Available:http://arxiv.org/abs/cs/0306050
[149] "Home - Groningen Meaning Bank." https://gmb.let.rug.nl/ (accessedNov. 20, 2021).
[150] "Amazon review data." https://jmcauley.ucsd.edu/data/amazon/ (accessed Nov. 20, 2021).
[151] "Yelp Dataset." https://www.yelp.com/dataset (accessed Nov. 20, 2021).
[152] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in Proceedings of the fifth ACM conference on Recommender systems, ser. RecSys '11. New York, NY, USA: Association for Computing Machinery, Oct. 2011, pp. 301–304.
[153] M. Casillo, B. B. Gupta, M. Lombardi, A. Lorusso, D. Santaniello, and C. Valentino, "Context Aware Recommender Systems: A Novel Approach Based on Matrix Factorization and Contextual Bias," Electronics, vol. 11, no. 7, p. 1003, Jan. 2022, number: 7 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2079-9292/11/7/1003
[154] S.-Y. Jeong and Y.-K. Kim, "Deep Learning-Based Context-Aware Recommender System Considering Contextual Features," Applied Sciences, vol. 12, no. 1, p. 45, Jan. 2022, number: 1 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2076-3417/12/1/45
[155] A. Sattar and D. Bacciu, "Graph Neural Network for Context-Aware Recommendation," Neural Processing Letters, vol. 55, no. 5, pp. 5357–5376, Oct. 2023. [Online]. Available: https: //doi.org/10.1007/s11063-022-10917-3
[156] S.-L. Vu and Q.-H. Le, "A Deep Learning Based Approach for Context-Aware Multi-Criteria Recommender Systems," Computer Systems Science and Engineering, vol. 44, no. 1, pp. 471–483, 2023. [Online]. Available: https://www.techscience.com/csse/v44n1/48080
[157] H. Vaghari, M. H. Aghdam, and H. Emami, "Diarec: Dynamic Intention-Aware Recommendation with Attention-Based Context- Aware Item Attributes Modeling," Journal of Artificial Intelligence and Soft Computing Research, vol. 14, no. 2, pp. 171–189, Mar. 2024. [Online]. Available: https://www.sciendo.com/article/10.2478/jaiscr-2024-0010
[158] T. Dilekh, S. Benharzallah, A. Mokeddem, and S. Kerdoudi, "Dynamic Context-Aware Recommender System for Home Automation Through Synergistic Unsupervised and Supervised Learning Algorithms," Acta Informatica Pragensia, vol. 13, no. 1, pp. 38–61, Apr. 2024, publisher: Acta Informatica Pragensia. [Online]. Available: https://doi.org/10.18267/j.aip.228

[159] O. Chapelle, E. Manavoglu, and R. Rosales, "Simple and Scalable Response Prediction for Display Advertising," ACM Transactions on Intelligent Systems and Technology, vol. 5, no. 4, pp. 61:1–61:34, Dec. 2015. [Online]. Available: https://doi.org/10.1145/2532128

[160] M. Richardson, E. Dominowska, and R. Ragno, "Predicting clicks: estimating the click-through rate for new ads," in Proceedings of the 16th international conference on World Wide Web, ser. WWW'07. New York, NY, USA

[161] Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, and C.-J. Lin, "Training and Testing Low-degree Polynomial Data Mappings via Linear SVM," The Journal of Machine Learning Research,vol. 11, pp. 1471–1490, Aug. 2010.

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

[163] M. Timmi, L. Laaouina, A. Jeghal, S. E. Garouani, and A. Yahyaouy, "Recommendation System based on User Trust and Ratings," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 13, no. 7, Art. no. 7, 31 2022, doi: 10.14569/IJACSA.2022.0130723.

[164] S. Rendle, "Factorization Machines," in 2010 IEEE International Conference on Data Mining, Dec. 2010, pp. 995–1000, iSSN:2374-8486.

[165] K. Iskandar, Noprianto, B. S. Abbas, B. Soewito, and R. Kosala, "Two-Way ANOVA with interaction approach to compare content creation speed performance in knowledge management system," in 2016 11th International Conference on Knowledge, Information and Creativity Support Systems (KICSS), Nov. 2016, pp. 1–5. doi: 10.1109/KICSS.2016.7951453.

[166] Y. Zheng, B. Mobasher, and R. Burke, "CARSKit: A Java-Based Context-Aware Recommendation Engine," in 2015 IEEE International Conference on Data Mining Workshop (ICDMW), Nov. 2015, pp. 1668–1671, iSSN: 2375-9259.

[167] W. McKinney et al., "Data structures for statistical computing in python," in Proceedings of the 9th Python in Science Conference, vol. 445. Austin, TX, 2010, pp. 51–56.

[168] I. M. A. Jawarneh, P. Bellavista, A. Corradi, L. Foschini, R. Montanari, J. Berrocal, and J. M. Murillo, "A Pre-Filtering Approach for Incorporating Contextual Information Into Deep Learning Based Recommender Systems," IEEE Access, vol. 8, pp. 40 485–40 498, 2020, conference Name: IEEE Access. [Online]. Available: https://ieeexplore.ieee.org/document/9004579

[169] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker,V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensor-Flow: a system for large-scale machine learning," in Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation, ser. OSDI'16. USA: USENIX Association, Nov. 2016, pp. 265–283.

[170] W. Shen, "Deepctr: Easy-to-use,modular and extendible package of deep-learning ctr models," https://github.com/shenweichen/ deepctr, 2017.

Abstract

Recommender systems (RS) are a subset of filtering systems primarily designed to deliver personalized recommendations by analyzing user-specific preferences. Traditional RS primarily utilize user and item data to generate recommendations but often neglect critical factors such contextual factors that could enhance the quality of these recommendations. Context-aware recommender systems (CARS) address this issue by integrating contextual information, which enables a deeper understanding of user preferences in various scenarios. However, implementing CARS poses several challenges, including the difficult task of obtaining contextual data due to users' privacy concerns. Additionally, incorporating contextual information into RS can lead to increased sparsity and dimensionality, posing further complications. In this thesis, we introduce a range of models that leverage advanced techniques like Factorization Machines and deep learning to surmount the inherent challenges of CARS and develop robust, cross-domain recommender systems. These models are designed around two principal functions: the collection of contextual information function which aims to address the scarcity of such data, and the ratings prediction function that incorporates this data in order to provide more precise and accurate recommendations. The results from our experiments demonstrate that our proposed models significantly outperform the baseline models, thereby confirming their effectiveness.

Keywords: Recommender Systems, Context-aware recommender systems, Deep learning Factorization machines, Named Entity Recognition.

Résume

Les systèmes de recommandation (SR) sont une sous-catégorie des systèmes de filtrage, principalement conçus pour fournir des recommandations personnalisées en analysant les préférences spécifiques des utilisateurs. Les SR traditionnels utilisent principalement des données sur les utilisateurs et les articles pour générer des recommandations, mais ils négligent souvent des facteurs critiques tels que les facteurs contextuels qui pourraient améliorer la qualité de ces recommandations. Les systèmes de recommandation sensibles au contexte (SRSC) résolvent cette problématique en intégrant des informations contextuelles, ce qui permet une compréhension plus approfondie des préférences des utilisateurs dans divers scénarios. Cependant, la mise en œuvre des SRSC présente plusieurs défis, y compris le problème de l'obtention de ces données en raison des préoccupations des utilisateurs concernant la confidentialité. De plus, l'incorporation d'informations contextuelles dans les SR peut compliquer le problème de la rareté des données et augmente considérablement le nombre des dimensions. Dans cette thèse, nous proposons une gamme de modèles qui exploitent des techniques avancées telles que les machines de factorisation et l'apprentissage profond pour surmonter les défis inhérents aux SRSC et développer des systèmes de recommandation multi-domaines robustes. Ces modèles sont conçus autour de deux fonctions principales : la fonction de collecte d'informations contextuelles qui vise à répondre à la rareté de ces données, et la fonction de prédiction des scores qui intègre ces données afin de fournir des recommandations plus précises et plus exactes. Les résultats de nos expériences démontrent que nos modèles proposés surpassent considérablement les modèles de base, confirmant ainsi leur efficacité.

Mots-clefs: Recommender Systems, Context-aware recommender systems, Deep learning Factorization machines, Named Entity Recognition.