

8-2019

# An explainable recommender system based on semantically-aware matrix factorization.

Mohammed Sanad Alshammari  
*University of Louisville*

Follow this and additional works at: <https://ir.library.louisville.edu/etd>

 Part of the [Other Computer Engineering Commons](#)

---

## Recommended Citation

Alshammari, Mohammed Sanad, "An explainable recommender system based on semantically-aware matrix factorization." (2019).  
*Electronic Theses and Dissertations*. Paper 3273.  
<https://doi.org/10.18297/etd/3273>

This Doctoral Dissertation is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact [thinkir@louisville.edu](mailto:thinkir@louisville.edu).

AN EXPLAINABLE RECOMMENDER SYSTEM BASED ON  
SEMANTICALLY-AWARE MATRIX FACTORIZATION

By

Mohammed Sanad Alshammari

Bachelor of Education in Computer Science, University of Ha'il, KSA, 2008

M.Sc., University of Leicester, UK, 2011

A Dissertation

Submitted to the Faculty of the

J.B. Speed School of Engineering of the University of Louisville

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy in Computer Science and Engineering

Department of Computer Engineering and Computer Science

University of Louisville

Louisville, Kentucky

August, 2019

Copyright 2019 © by Mohammed Sanad Alshammari

All rights reserved



AN EXPLAINABLE RECOMMENDER SYSTEM BASED ON  
SEMANTICALLY-AWARE MATRIX FACTORIZATION

By

Mohammed Sanad Alshammari  
Bachelor of Education in Computer, University of Ha'il, KSA, 2008  
M.Sc., University of Leicester, UK, 2011

A Dissertation Approved On

July 15, 2019

by the following Dissertation Committee

---

Dr. Olfa Nasraoui, Dissertation Director

---

Dr. Hichem Frigui

---

Dr. Nihat Altiparmak

---

Dr. Antonio Badia

---

Dr. Scott Sanders

## DEDICATION

To my parents, Sanad Suhaiman Alshammari and Aljazi Abdullah Alshammari.

## ACKNOWLEDGMENT

First of all, I would like to praise and thank Almighty God (Allah) for everything he blessed me with; without his grace, mercy, and well, nothing would have happened. He alone helped me to succeed.

Then, I appreciate my sponsor, the government of Saudi Arabia and Northern Border University or supporting me to pursue the Ph.D. degree. Also, I pay recognition to the University of Louisville and the Department of Computer Engineering and Computer Science for allowing me to join the graduate program and advance in my studies and research.

Next, I would like to express my sincere gratitude to my advisor Prof. Olfa Nasraoui for the continuous support of my Ph.D. study and related research, for her patience, motivation, and immense knowledge. Her guidance helped me at all the times of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D. study. Besides my advisor, I would like to thank the rest of my thesis committee for serving as my committee members and for their insightful comments and encouragement throughout my Ph.D. Study: Prof. Hichem Frigui, Dr. Antonio Badia, Dr. Nihat Altiparmak, and especially Dr. Scott Sanders who has been very helpful in the user study evaluation. Also, I thank my fellow friends and labmates for supporting and encouraging me to strive towards my goals.

A special thanks to my family. Words cannot express how grateful I am to my mother Aljazi and father Sanad for all of the sacrifices that they have made on my behalf. Their unconditional love, caring, and support since day one back in 1992 were what brought me where I am today.

I would like to express appreciation and gratitude to my dear wife, Alwadhhaa, who spent sleepless nights taking care of our children and was always my closest support in the moments when there was no one to answer my needs.

My beloved daughters and sons, Jood, Fahad, Abdullah, and Fatimah, thank you for being in my

life. You are my main source of motivation and encouragement to achieve my goals. I am grateful to Allah that you are in my life.

Last but not least, I would like to thank my siblings: my brothers Abdulrahman and Salman and sisters Mona, Nawal, Fatimah and Tahani and their families for supporting me spiritually throughout writing this thesis and my life in general.



## ABSTRACT

### AN EXPLAINABLE RECOMMENDER SYSTEM BASED ON SEMANTICALLY-AWARE MATRIX FACTORIZATION

Mohammed Sanad Alshammari

July 15, 2019

Collaborative Filtering techniques provide the ability to handle big and sparse data to predict the ratings for unseen items with high accuracy. Matrix factorization is an accurate collaborative filtering method used to predict user preferences. However, it is a black box system that recommends items to users without being able to explain why. This is due to the type of information these systems use to build models. Although rich in information, user ratings do not adequately satisfy the need for explanation in certain domains. White box systems, in contrast, can, by nature, easily generate explanations. However, their predictions are less accurate than sophisticated black box models. Recent research has demonstrated that explanations are an essential component in bringing the powerful predictions of big data and machine learning methods to a mass audience without a compromise in trust. Explanations can take a variety of formats, depending on the recommendation domain and the machine learning model used to make predictions. Semantic Web (SW) technologies have been exploited increasingly in recommender systems in recent years. The SW consists of knowledge graphs (KGs) providing valuable information that can help improve the performance of recommender systems. Yet KGs, have not been used to explain recommendations in black box systems. In this dissertation, we exploit the power of the SW to build new explainable recommender systems. We use the SW's rich expressive power of linked data, along with structured information search and understanding tools to explain predictions. More specifically, we take advantage of semantic data to learn a semantically aware latent space of users and items in the matrix factorization

model-learning process to build richer, explainable recommendation models. Our off-line and on-line evaluation experiments show that our approach achieves accurate prediction with the additional ability to explain recommendations, in comparison to baseline approaches. By fostering explainability, we hope that our work contributes to more transparent, ethical machine learning without sacrificing accuracy.

# TABLE OF CONTENTS

	Page
DEDICATION	iii
ACKNOWLEDGMENT	iv
ABSTRACT	vi
LIST OF TABLES	xi
LIST OF FIGURES	xiii
1 INTRODUCTION AND MOTIVATION	1
1.1 Introduction . . . . .	1
1.2 Problem Statement . . . . .	2
1.3 Assumptions and Research Scope . . . . .	2
1.4 Research Contribution . . . . .	3
1.5 Document Organization . . . . .	4
2 BACKGROUND AND LITERATURE REVIEW	5
2.1 Semantic Web . . . . .	5
2.1.1 The Early use of the Semantic Web in User Profile Building . . . . .	5
2.1.2 RDF . . . . .	7
2.1.3 SPARQL . . . . .	8
2.2 Recommender Systems . . . . .	9
2.2.1 Introduction . . . . .	9
2.2.2 Recommendation Techniques . . . . .	10

2.2.3	Semantic-Aware Content-based Recommender Systems . . . . .	11
2.2.4	Collaborative Filtering (CF) . . . . .	18
2.2.5	Explanations . . . . .	22
2.2.6	Inference of Facts . . . . .	39
2.3	Summary . . . . .	40
3	<b>PROPOSED WORK</b> . . . . .	41
3.1	Introduction . . . . .	41
3.1.1	Semantic Knowledge Graphs (KGs) . . . . .	41
3.1.2	The Semantic Web’s Effect in Latent Space . . . . .	42
3.2	Asymmetric Semantic Explainable Matrix Factorization (ASEMF) . . . . .	43
3.2.1	ASEMF Semantic KGs . . . . .	43
3.2.2	Model Building . . . . .	46
3.2.3	Explanation Style . . . . .	49
3.3	Semantic Explainable Matrix Factorization (SemEMF) . . . . .	50
3.3.1	SemEMF Semantic KGs . . . . .	51
3.3.2	Model Building . . . . .	51
3.3.3	Explanation Style . . . . .	53
3.4	Merged Semantic Explainable Matrix Factorization (MergedSemEMF) . . . . .	53
3.4.1	MergedSemEMF Semantic KGs . . . . .	53
3.4.2	Model Building . . . . .	53
3.4.3	Explanation Style . . . . .	55
3.5	Linked Data Semantic Distance Matrix Factorization (LDSDMF) . . . . .	55
3.5.1	Linked Data Semantic Distance (LDS) . . . . .	56
3.5.2	Model Building . . . . .	60
3.5.3	Explanation Style . . . . .	61
3.6	Inferred Fact Style Explanation (IFSE) . . . . .	61
4	<b>EXPERIMENTAL EVALUATION</b> . . . . .	63
4.1	Experimental Setting . . . . .	63
4.2	Accuracy Evaluation . . . . .	65

4.3	Recommendation Evaluation . . . . .	67
4.4	Explainability Evaluation . . . . .	68
4.4.1	Semantic Explainability Metrics . . . . .	70
4.4.2	Neighborhood Explainability Metrics . . . . .	74
4.5	Analysis of Results . . . . .	76
4.6	Real User Study . . . . .	78
4.6.1	Hypothesis . . . . .	78
4.6.2	Methods . . . . .	78
4.6.3	Subject Recruitment . . . . .	79
4.6.4	Sample Size Estimation . . . . .	79
4.6.5	Procedures . . . . .	80
4.6.6	Analysis of Results . . . . .	84
4.6.7	Hypothesis Testing . . . . .	91
4.7	Summary of Experimental Evaluation . . . . .	95
5	CONCLUSION AND FUTURE WORK	96
	REFERENCES	98
	CURRICULUM VITAE	115

## LIST OF TABLES

2.1	Explanation Styles. . . . .	23
2.2	A comparison of some recommender systems based on the (a) domain, (b) RS style, (c) availability of explanation, (d) system awareness of semantics, (e) user study, and (f) CSP solution to the cold start problem. . . . .	26
2.3	Explanation objectives. . . . .	38
4.1	The effect of applying the inferred fact mechanism on the semantic KGs' sizes in the movie domain; the first and second rows show the sizes before and after the application, respectively. . . . .	65
4.2	The effect of applying the inferred fact mechanism to the semantic KGs' sizes in the book domain; the first and second rows show the sizes before and after the application, respectively. . . . .	65
4.3	RMSE versus the number of latent factors $K$ in the movie domain. SemEMF, MergedSemEMF, ASEMf_UIB, and LDSDMF denote our proposed methods. . .	66
4.4	RMSE versus the number of latent factors $K$ in the book domain. SemEMF, MergedSemEMF, ASEMf_UIB, and LDSDMF denote our proposed methods. . . . .	66
4.5	RMSE significance test results in the movie domain ( $K = 10$ ). . . . .	67
4.6	RMSE significance test results in the book domain ( $K = 50$ ). . . . .	67
4.7	MAP significance test results in the movie domain ( $K = 10$ ). . . . .	68
4.8	MEP@10 significance test results in the movie domain ( $K = 10$ and $\theta^s = 0.25$ ) using semantic KGs to calculate explainability metrics. . . . .	70
4.9	MER@10 significance test results in the movie domain ( $K = 10$ and $\theta^s = 0.25$ ) using semantic KGs to calculate explainability metrics. . . . .	70

4.10 xF-score@10 significance test results in the movie domain ( $K = 10$ and $\theta^s = 0.25$ ) using semantic KGs to calculate explainability metrics. . . . .	72
4.11 MEP@10 significance test results in the book domain ( $K = 50$ and $\theta^s = 0.25$ ) using semantic KGs to calculate explainability metrics. . . . .	72
4.12 MER@10 significance test results in the book domain ( $K = 50$ and $\theta^s = 0.25$ ) using semantic KGs to calculate explainability metrics. . . . .	72
4.13 xF-score@10 significance test results in the book domain ( $K = 50$ and $\theta^s = 0.25$ ) using semantic KGs to calculate explainability metrics. . . . .	72
4.14 MEP@10 significance test results in the movie domain ( $K = 10$ and $\theta^n = 0.25$ ) using the neighborhood based explainability metrics. . . . .	74
4.15 MER@10 significance test results in the movie domain ( $K = 10$ and $\theta^n = 0.25$ ) using the neighborhood based explainability metrics. . . . .	74
4.16 xF-score@10 significance test results in the movie domain ( $K = 10$ and $\theta^n = 0.25$ ) using the neighborhood based explainability metrics. . . . .	76
4.17 Likert scale survey questions. . . . .	81
4.18 Demographic questions. . . . .	81
4.19 Mean and standard deviation for all groups for regarding all three explanation aspects.	93
4.20 Categorization of the survey questions from Table 4.17 according to the research questions. . . . .	93
4.21 Tukey multiple comparisons of means at 95% family-wise confidence interval for <b>satisfaction</b> . . . . .	94
4.22 Tukey multiple comparisons of means at 95% family-wise confidence interval for <b>transparency</b> . . . . .	94
4.23 Tukey multiple comparisons of means at 95% family-wise confidence interval for <b>effectiveness</b> . . . . .	94

## LIST OF FIGURES

1.1	Black Box Matrix Factorization on the left and Explainable Semantic Matrix Factorization on the right. . . . .	1
2.1	A simple RDF graph. . . . .	7
2.2	The high level architecture of a content-based recommender [1]. . . . .	11
2.3	Matrix Factorization Flow Chart. . . . .	20
2.4	An Example of dbrec’s Output, Source: [2]. . . . .	28
2.5	An Example of TasteWeight’s Output, Source: [3]. . . . .	28
2.6	An Example of Entity-based RS’ Output, Source: [4]. . . . .	28
2.7	An Example of SEP’s Output, Source: [5]. . . . .	29
2.8	An Example of RecExp’s Output, Source: [6]. . . . .	29
2.9	An Example of LOD-TRS’ Output, Source: [7]. . . . .	30
2.10	An Example of SemAuto’s Output, Source: [8]. . . . .	31
2.11	An Example of MovieExplain’s Output, Source: [9]. . . . .	32
2.12	An example of Tagsplanation’s Output, Source: [10]. . . . .	32
2.13	An Example of Hybrid-RS’ Output, Source: [11]. . . . .	33
2.14	An Example of Embedding-RS’ Output, Source: [12]. . . . .	33
2.15	EMF, Source: [13] [14] [15] neighborhood style explanation examples; the example on the left is user-based, and the example on the right is item-based. . . . .	37
2.16	Snapshot of the camera recommender system, Source: [16]. . . . .	40
3.1	Proposed methods and other approaches. . . . .	42
3.2	Asymmetric Semantic Explainable Matrix Factorization (ASEMF). . . . .	44



3.3	A histogram-like semantic ISE (left) and a plain text semantic ISE (right) recommendation for sample user 1. . . . .	50
3.4	A histogram-like semantic ISE (left) and a plain text semantic ISE (right) recommendation for sample user 2. . . . .	50
3.5	Semantic Explainable Matrix Factorization. . . . .	51
3.6	Merged Semantic Explainable Matrix Factorization. . . . .	54
3.7	LDSDMF flowchart. . . . .	56
3.8	A Generic Semantic Knowledge Graph. . . . .	57
3.9	Inferred Facts: Movie Example. . . . .	62
4.1	The mapping process for the movie domain. . . . .	64
4.2	The mapping process for the book domain. . . . .	64
4.3	The graph shows the MAP@10 results of all methods while varying $K$ in the movie domain. . . . .	68
4.4	This graph shows the MAP@10 results of all methods while varying $K$ in the book domain. . . . .	69
4.5	The upper graph shows the results of MEP@10 for all methods, while the middle one shows MER@10 for all methods, and the lower graph illustrates the results of all methods using the xF-score metric. All explainability metrics utilize semantic KGs. All the results are in the movie domain. . . . .	71
4.6	The upper graph shows the results of MEP@10 for all methods, while the middle one shows MER@10 for all methods, and the lower graph illustrates the results of all methods using the xF-score metric. All explainability metrics utilize semantic KGs. All the results are in the book domain. . . . .	73
4.7	The upper graph shows the results of MEP@10 for all methods, while the middle one shows the MER@10 results for all methods, and the lower graph illustrates the results of all methods. All explainability metrics are based on the neighborhood explainability graph. All the results are in the movie domain. . . . .	75

4.8	The upper graph shows the results of MEP@10 for all methods, while the middle one shows MER@10 for all methods, and the lower graph illustrates the results of all methods using the xF-score metric. All explainability metrics use semantic KGs. All the results are in the book domain. . . . .	77
4.9	A comparison of two explanations, high on the left and low on the right, that are exposed to the user during the experiment. The explanation on the left shows more semantic attributes than the one on the right. . . . .	79
4.10	A snapshot of the recommender system app showing a list of movies for the user to rate. . . . .	82
4.11	A snapshot of a recommendation and its semantic explanation presented to a user. The share of interest is the likeability degree computed using Equation 3.2 . . . . .	83
4.12	A Vertical bar chart of the answers to the questions in Table 4.17 for all participants.	85
4.13	A Vertical bar chart of the answers to the questions in Table 4.17 for participants in the group "High". . . . .	85
4.14	A Vertical bar chart of the answers to the questions in Table 4.17 for participants in the group "Medium". . . . .	86
4.15	A Vertical bar chart of the answers to the questions in Table 4.17 for participants in the group "Low". . . . .	86
4.16	A Heat-map plot of the answers to the questions in Table 4.17 for all participants. .	86
4.17	A Heat-map plot of the answers to the questions in Table 4.17 for participants in the group "High". . . . .	87
4.18	A Heat-map plot of the answers to the questions in Table 4.17 for participants in the group "Medium". . . . .	87
4.19	A Heat-map plot of the answers to the questions in Table 4.17 for participants in the group "Low". . . . .	87
4.20	Distribution of the participants' satisfaction with the explanation. . . . .	88
4.21	Distribution of the participants' gender. . . . .	89
4.22	Distribution of the participants' age. . . . .	89
4.23	Distribution of the participants' major of study. . . . .	89
4.24	Distribution of the participants' weekly hours watching movies. . . . .	89

4.25	Distribution of the participants' favorite movies' attributes. . . . .	90
4.26	Distribution of the participants' familiarity with recommender systems. . . . .	90
4.27	Distribution of the participants' most used online entertainment services. . . . .	90
4.28	Visualization of differences of mean levels of pairs of groups for <b>satisfaction</b> . . . .	92
4.29	Visualization of differences of mean levels of pairs of groups for <b>transparency</b> . . .	93
4.30	Visualization of differences of mean levels of pairs of groups for <b>effectiveness</b> . . .	94

# CHAPTER 1

## INTRODUCTION AND MOTIVATION

### 1.1 Introduction

Collaborative filtering (CF) is a recommender system technique that uses the explicit preferences of users, such as ratings, to recommend items [17]. In contrast, content-based filtering (CBF) techniques rely only on detailed information about an item’s content or description to make recommendations [18], which is a major advantage in complex domains. However, the most accurate CF methods, such as matrix factorization (MF) [19], lack transparency. Hence, despite its efficient in handling of large data sets and a high degree of accuracy in predicting unseen items’ ratings, MF fails to justify its output. This is because MF only utilizes users’ explicit preferences or ratings to build a prediction model. For this reason, MF is considered a black box recommender system (see the left side of Figure 1.1).

In addition to the lack of explainability, users’ explicit preferences or ratings of past items may not be sufficient for the model to be able to recommend new items for which there is no rating data. This problem is known as the cold start problem.

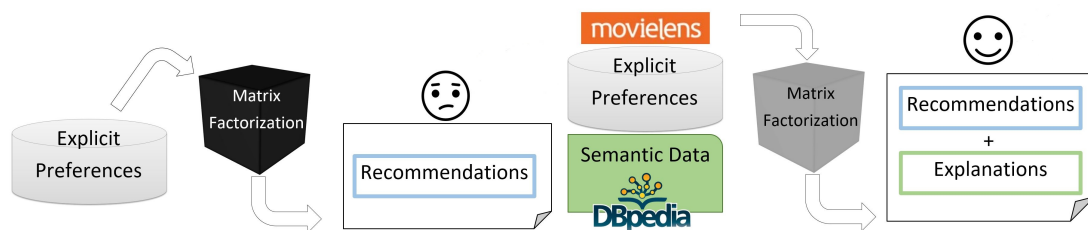


Figure 1.1: Black Box Matrix Factorization on the left and Explainable Semantic Matrix Factorization on the right.

There is, therefore, a need to overcome both the black box and cold start problems. One way to cope with these two problems involves leveraging the semantic web. More specifically, linked open data (LOD) [20] is a project where data is linked, structured, and connected on the web. In recent years, the web has been saturated with data and is a good source of rich information. The goal of LOD is to make this information machine processable and semantically linked. For example, in the movie domain, information about movie stars, such as related producers and writers, is available in a linked manner. When an actor has starred in two movies, these two movies are considered linked. This linkage of information can help infer relationships between movies.

## **1.2 Problem Statement**

The research questions that we attempt to answer are as follows: Can we build a recommender system using matrix factorization (MF) that, in addition to being accurate, succeeds in explaining the recommendations using Semantic Web resources? Are the semantics of users and items effective in building an explainable low dimensional (latent factor) space? Will the recommendations, after exploiting the semantics, be accurate? Do more semantic properties increase the explanation effectiveness?

## **1.3 Assumptions and Research Scope**

The focus of this research is on the matrix factorization technique in addition to the semantic KGs that we rely on for justification purposes. We assume that the recommender system is a black box algorithm that relies on matrix factorization to build the model. We also assume that there is a sophisticated overlap between all knowledge sources. This means that all items (e.g., movies or books) in the dataset that contains users' explicit preferences also exist in the semantic web knowledge source (e.g., DBpedia).

The scope of our research is limited to the collaborative filtering technique for recommendations and to semantic web technologies for explanation generation. Users' preferences are their ratings of movies, whereas the semantic web technology that we incorporate is the simple protocol and resource description framework query language (SPARQL) [21]. We query DBpedia to extract semantic knowledge graphs that link items, such as movies or books to semantic properties, and

then use this information to generate an explanation while simultaneously building the model. This semantic data is expected to cause some items to be projected closer to some users based on how many properties the items and users share based on semantics.

## 1.4 Research Contribution

Our work contributes to the recommender system and semantic web fields by:

1. Designing semantic Knowledge Graphs (KGs) that can be used to interpret and justify big data black box predictors, such as matrix factorization, while preserving prediction accuracy as illustrated in the right side of Figure 1.1.
2. Proposing a two-step model that uses the designed semantic KGs to learn semantically-aware latent spaces of users and items Asymmetric Semantic Explainable Matrix Factorization (ASEMF).
3. Proposing a one-step model that incorporates the semantic KGs in learning the low dimensional latent spaces Semantic Explainable Matrix Factorization (SemEMF).
4. Proposing a model that combines multiple explanation styles, semantics (SemEMF) and neighborhood style Explainable Matrix Factorization (EMF), in learning explanation-aware latent spaces Merged Semantic Explainable Matrix Factorization (MergedSemEMF).
5. Proposing a model that takes advantage of two algorithms Linked Data Semantic Distance (LDSD) and Joint Matrix Factorization (JMF) for building a semantically more comprehensive model Linked Data Semantic Distance Matrix Factorization (LDSDMF).
6. Proposing a model that augments the semantic explanations based on inferred facts about users and semantic attributes, (IFSE) Inferred Fact Style Explanation.
7. Presenting offline evaluation for the proposed models.
8. Performing a user study to evaluate our proposed model (LDSDMF) and proposed explanation style Inferred Fact Style Explanation (IFSE) for online evaluation.

## **1.5 Document Organization**

In the following chapters, we first review previous work that has used the semantic web in the recommendation process in Chapter 2, focusing specifically on user profile building and matrix factorization. Then, we describe our proposed methods in Chapter 3, and present their experimental evaluation in Chapter 4. Finally we present our final conclusions and future work in Chapter 5.

## CHAPTER 2

# BACKGROUND AND LITERATURE REVIEW

### 2.1 Semantic Web

The goal behind the semantic web is to allow systems to intelligently search, combine, process, and interact [22]. It is important to mention that the contents of these systems must be systematically processable by machines [22]. Specifically, content, like media, is processed, combined, and aggregated by humans with no help from systems. The goal of the semantic web is to make systems do these jobs by themselves by converting the information on the web into data that is machine readable and understandable using a specific schema and format [23]. The semantic web enables us to utilize not only keywords when searching on the web but also the underlying semantics, such as synonyms and homonyms, to enrich the user's experience [23]. These technologies are designed to make more data available for systems to process without human intervention. "Anyone can say anything about anything" [23] on the web, which implicitly means that there is a vast amount of data on the web. Therefore, an intelligent way to organize this data is required to help humans manage data more rapidly, accurately, and efficiently. Notably, the semantic web could be referred to as 'linked data' [20], which means structured and connected data on the web.

In the following sections, the resource description framework (RDF) and SPARQL concepts, which are two technologies of the semantic web, are explored in greater detail.

#### 2.1.1 The Early use of the Semantic Web in User Profile Building

In this section, we shed light on some previous studies that have sought to mine user profiles and compare them with our focus in this research. The interaction between the semantic web and



web mining was discussed previously by [24] and [25]. They stated that discovering the meaning of web content could be easier with the help of the semantic web. Moreover, usage mining is a considerable source of knowledge and could be used to present more meaningful information on the web. A survey was conducted by [26], who described the development of the semantic web and web mining. They also aimed to clarify web space evaluation, where multiple factors, including semantic web mining, software agents, and human agents interact. They concluded the survey by predicting that semantics will be more attractive in the research area of web mining because of its ability to make information on the web machine processable and understandable. The combination of the semantic web and web mining is also studied by [27], who found semantic web mining can successfully enhance the output of web mining after taking advantage of semantic web technologies. Building user profiles is a fundamental task in the recommender systems field. For this reason and because the semantic web is a platform of linked data, considerable effort has gone into using the semantic web in this area.

Usage data is one of the most important sources for understanding user interests and desires, which will help in building user profiles [28], [29]. Furthermore, [30] emphasized that it is crucial to utilize both website usage data and the ontological content of items to build richer user profiles. In their work, they built user profiles by integrating semantics obtained from ontologies, website structures, and user activity. However, in this study, the focus was on users' ratings of items and the underlying semantics to increase the accuracy of recommendations and generate explanations.

[31] used webpage textual content data and WordNet ontology [32] to build a semantically-enhanced user model that can help achieve personalization and understand user needs in information retrieval.

In another study, [33] exploited the power of the semantic web for personalized information retrieval within the e-learning domain [34], where they leveraged learning content, such as lectures, and semantic user profiles.

Finally, [35] attempted to create a more robust user profile by preprocessing user usage data as performed earlier in [36], [37], and [38]. This process includes robust knowledge discovery algorithms that result in noise and outlier resistance. The study concluded that although a preprocessing stage may reduce the quality of user profiles, it allows the data mining process to be performed more rapidly.

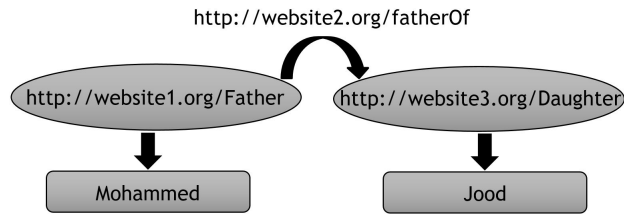


Figure 2.1: A simple RDF graph.

## 2.1.2 RDF

The web currently relies on languages, such as HTML, which allow anyone to publish a document and make it available online. However, since the number of documents and amount of data published on the web have increased rapidly, the need for an intelligent mechanism to handle them arose, and hence, RDF was developed [23]. This language outputs structured information in a formal manner [22]. The original objective of this language was to allow applications, such as travel agents' websites, to exchange data while maintaining their rules [22]. Since HTML is limited, extensible markup language (XML) was developed to organize semantics on the web. This means that XML is the foundation language for RDF. Although RDF has a specific syntax and structure, it is the basic language for semantic web development [22]. A simple RDF graph is visualized in Figure 2.1.

### 2.1.2.1 RDF Data Model

RDF components are [23]:

1. Resources:

Resources, in this context, means objects or things, such as players, machines, notebooks, or mountains, and they each have a unique uniform resource identifier (URI) for identification purpose on the web when pointing to them.

2. Properties:

Properties represent the relationship between resources. For example, in Figure 2.1, Mohammed is the father of Jood. 'Father of,' in this case, is the property that illustrates the relationship between Mohammed and Jood.

### 3. Statements:

Statements are a combination of resources, properties, and values.

### 4. Graph:

Figure 2.1 shows how a statement is represented in a graph.

In conclusion, RDF basically enables data to be formatted in a machine-understandable format. Briefly, RDF works by dividing the concepts contained in web documents into a relationship that includes a subject, predicate, and object, for example, 'Fahad isWorkingAt T-mobile.' This relationship yields additional information regarding Fahad. Since he works at T-mobile, other people who work at the same company could benefit from Fahad's other relationships when conducting an online search. Specifically, the search engine will consider the content of Fahad's profile and then make suggestions to his colleagues.

## 2.1.3 SPARQL

SPARQL is a query language inspired by structured query language (SQL), and it is used to query information organized using RDF syntax. There are many similarities between these two languages [23]. However, the structure of data that SQL and SPARQL query is completely different [22].

RDF is stored in a database called a triple store so that SPARQL can perform its queries on RDF files. An example of a triple store that provides the service of a SPARQL endpoint is DBpedia <sup>1</sup> [23].

As mentioned previously, triple RDF consists of a subject, predicate, and object. These three components are substantial when querying RDF files [23]. SPARQL is similar to SQL in terms of syntax. For example, SELECT has the same functionality in both technologies. Moreover, a WHERE clause in SPARQL acts like pattern matching for the triple. For instance, consider the following triple: '?person dbo:birthPlace :London.' If this is in the WHERE clause, it will return all persons whose birthplace is London. It is important to define the abbreviations such as rdf, foaf, dbpedia, and others at the beginning of the query [23].

---

<sup>1</sup>[dbpedia.org/sparql](http://dbpedia.org/sparql)

## 2.2 Recommender Systems

### 2.2.1 Introduction

Recommender Systems aim to help individual users select the next item based on their previous preferences and choices [39]. For example, the amazon.com e-commerce website personalizes items for each customer, and if we consider books as an example, each customer will see different book suggestions [40]. It is worth mentioning that if two or more users share similar interests, they may see similar item suggestions. In addition, there are non-personalized recommendations, where the recommendation is based on other criteria, such as popularity (e.g., the top five movies based on the number of views). In terms of implementation, these are easier than personalized suggestions [41], and are used in magazines and newspapers since they may not have enough information about the user to make a personalized recommendation. They thus use other factors, such as the number of views or clicks, to rank the recommendations. These type of recommendations are distinct from modern intelligent RSs that are personalized and data-driven, which we consider in this research work [41].

RSs use the ratings of products, which users explicitly express through feedback, to make recommendations. The system may also consider visiting some pages as a sign of interest and, based on this information, RSs take action [41].

Recently, RSs have become important for the following reasons [41]:

- Websites like Amazon, YouTube, Netflix, and others need RSs so they can improve their ability to help a very large number of users navigate and discover relevant objects among many options.
- Conferences and workshops have been dedicated to RSs research.
- University courses have been dedicated to RSs.
- Famous scientific journals have covered the development of RSs.

In addition, there are several benefits that service providers look for when exploiting RSs [41]:

- Increase sales.

- Sell more diverse items.
- Increase users satisfaction.
- Strengthen fidelity.
- Understand user's needs.

### 2.2.2 Recommendation Techniques

Predicting the utility of items that RSs should recommend to the user is important; some items are not worth recommending to users [42], [43]. To model the degree of utility of item  $i$  for user  $u$  as a function  $R(u, i)$ , we consider user ratings. Then, collaborative filtering predicts the utility of items for each user [42], [43].

It is important to note that the utility of items for some users can be influenced by their knowledge level (i.e., expert versus beginner) in a specific field [43], which will have an impact on recommendations.

There are six different classes of recommendation approaches [44]:

- Content-based: The system learns to recommend items that are similar to the ones that the user liked in the past.
- Collaborative filtering: Recommends items to the user that other users with similar preferences have liked in the past. The degree of similarity between two users' preferences is calculated based on the degree of similarity in their rating history. It is the most popular RS technique.
- Demographic: Recommends items based on the user's demographic profile.
- Knowledge-based: Recommends items based on specific domain knowledge; how certain item features meet the user's needs and preferences. Specifically, how the item is preferable for the user.
- Community-based: Recommends items based on the preferences of the user's friends.

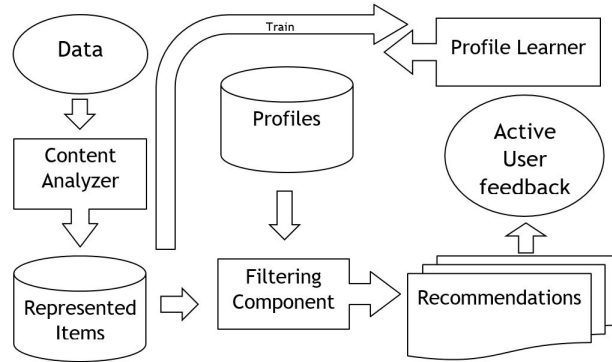


Figure 2.2: The high level architecture of a content-based recommender [1].

- Hybrid recommender systems: These RSs are based on a combination of the above techniques.

### 2.2.3 Semantic-Aware Content-based Recommender Systems

Content-based recommender systems (CBRS) consider all documents and item descriptions that a user has previously shown an interest in via ratings and then builds a user profile that matches the user’s interest in such items. Once the user explicitly shows interest in an item, the algorithm analyzes the item’s features and then recommends similar items to the user [45], [46], [1]. For example, when a user purchases a pillow from amazon.com, the recommender system will suggest a pillow cover as it is a closely related item. Unfortunately, this keyword-based technique faces problems, such as polysemy, synonymy, multi-word expressions, and others. Semantic technologies are a way to overcome these obstacles [1]. Knowledge sources, such as DBpedia and Freebase, assist in the switch from keyword-based techniques to concept-based techniques used to build items and profiles. There are two approaches to the use of semantic technologies in content-based recommender systems: top-down and bottom-up [1]. We will elaborate on them in this chapter. Figure 2.2 shows the high-level architecture of a content-based recommender system.

In the following sections, we discuss how data are represented in content-based RS and which algorithms are used. Then, we describe some strengths and weaknesses of these methods.

### 2.2.3.1 Content-based Recommender Systems

**Item Representation and algorithms** Since many early content-based RSs dealt with textual content [46], [45], retrieval models come into play. A vector space model (VSM) is commonly used to match documents based on keywords. It uses the term frequency-inverse document frequency (TF-IDF) technique. Let  $D = \{d_1, d_2, \dots, d_N\}$  represent a set of documents and  $T = \{t_1, t_2, \dots, t_n\}$  denote the dictionary.  $T$  is obtained through processing operations, such as tokenization and stemming [47]. Each document is represented as  $d = \{w_{1j}, w_{1j}, \dots, w_{nj}\}$  where  $w_{kj}$  is the weight for term  $t_k$  in document  $d_j$ . The definition of TF-IDF [48] is as follows:

$$TF - IDF(t_k, d_j) = TF(t_k, d_j) \cdot \log \frac{N}{n_k} \quad (2.1)$$

The first part represents the term frequency and the log term denotes the inverse document frequency.  $N$  is the number of documents, and  $n_k$  represents the number of documents in which  $t_k$  is found at least once. A normalization formula is needed, thus cosine normalization is typically used [48]. This results in the following weight  $w_{k,j}$  for a term  $k$  in document  $j$ :

$$w_{k,j} = \frac{TF\_IDF(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} TF_{IDF}(t_s, d_j)^2}} \quad (2.2)$$

To compute the similarity between documents, the cosine similarity measure is commonly used, which is given by

$$sim(d_i, d_j) = \frac{\sum_k w_{ki} \cdot w_{kj}}{\sqrt{\sum_k w_{ki}^2} \cdot \sqrt{\sum_k w_{kj}^2}} \quad (2.3)$$

After documents are processed, algorithms are needed to perform the task of learning the profile of the user. The following list is a summary of some of these algorithms and how they work:

1. Naive Bayes (NB): NB classifies data based on the posterior probability of belonging to a class, given by

$$P(c | d) = \frac{P(c)P(d | c)}{P(d)} \quad (2.4)$$

$P(c|d)$  denotes the probability of document  $d$  belonging to class  $c$ , while  $P(c)$  means the probability of observing a document in class  $c$ ,  $P(d)$  is the probability of observing the instance  $d$ , and  $P(d|c)$  represents the probability of observing the document  $d$  given  $c$ . In Content-Based Filtering the class  $c$  can correspond to relevance or non-relevance to a particular user.

## 2. Rocchio's Algorithm:

The Rocchio's algorithm is used to refine and thus personalize the user's query  $q$  by considering the relevance of feedback obtained from the user [49]:

$$q_{ki} = \beta \cdot \sum_{(d_j \in POS_i)} \frac{w_{kj}}{|POS_i|} - \gamma \cdot \sum_{(d_j \in NEG_i)} \frac{w_{kj}}{|NEG_i|} \quad (2.5)$$

$w_{kj}$  represents the weight of term  $t_k$  in document  $d_j$  using TF-IDF.  $POS_i$  and  $NEG_i$  denote the set of positives and negatives instances for class  $c_i$  in the training set.  $\beta$  and  $\gamma$  are controlling parameters to set the importance of positive and negative instances.

## 3. Nearest Neighbor:

The nearest neighbor technique stores the training data and then classifies new items based on nearest neighbors from the stored training data using a similarity function, such as the cosine similarity [50]. Like the Naive Bayes classifier above, classes of interest here are the relevance and the non-relevance to a user.

**Advantages and Disadvantages of Content-based RS** The advantages of content-based RSs (CBRSs) include user independence, transparency, and handling new items. The disadvantages include limited content analysis, over-specialization, and new-user cold-start problems [1].

### 2.2.3.2 Top-Down Approach

In this approach, the idea is to utilize knowledge resources together to form the user profile and then find the information that the user needs [1].

There are three aspects that CBRSs incorporate to increase the accuracy of recommendations:

- Ontological knowledge use [51].



- Unstructured or semi-structured encyclopedic knowledge source utilization [52].
- Linked open data cloud incorporation [53].

These aspects are discussed in the following paragraph.

#### 1. Ontological Resources:

WordNet [54] introduced linguistic knowledge to the public and the research field. The interpretation of the semantic meaning of the content of WordNet was obtained by algorithms of word sense disambiguation (WSD) [55],[56, 57]. Both technologies are utilized to build a user profile [1]. Nonetheless, WordNet is still limited in terms of entities' names, events, and specific knowledge [1]. Thus, the need for better technology arises. One such technology is the semantic web [58]. One of the powerful technologies of the semantic web is ontology, and its role is to handle the domain knowledge in a specific syntax. In RSs-ontology integration, user and item profiles are built using concepts from ontology [1]. Examples of this type of recommendation are explored in a later section.

Although the use of ontologies in the recommendation process results in more relevant recommendations and less ambiguous user profiles [1], ontology needs experts in the design, which consequently means time consumption, in addition to required maintenance efforts [59].

#### 2. Unstructured or semi-structured:

Since the early years of artificial intelligence, the major role of knowledge resources was recognized [60]. For this reason, many knowledge resources, either structured or unstructured, became available on the web, such as the Wikipedia encyclopedia and Yahoo! web directory. These types of knowledge resources have been exploited in the field of CBRs by finding more related concepts and features [1]. It is not surprising that Wikipedia is the most widely used knowledge resource for several reasons and features: it is free, it covers a wide range of topics, it is multilingual [1] and highly accurate [61]. Wikify [62] and Tagme [63] are two projects that exploit Wikipedia to perform the feature selection task.

Explicit semantic analysis (ESA) is a technique used to improve an item's representation by utilizing Wikipedia to create new features [64]. ESA works by giving weighted vectors to the concepts contained in documents retrieved from the encyclopedia. The difference between

this technique and LSA [65] is that LSA deals with latent features, whereas ESA deals with explicit features derived from Wikipedia [1].

Lastly, BabelNet is an encyclopedic dictionary that integrates Wikipedia and WordNet to generate a huge multilingual semantic network [66].

### 3. Linked Open Data

Linked open data became popular in recent years due to the collaborative efforts of the semantic web community [67]. The structure of this enormous amount of data follows the standard of an RDF, as well as the query language SPARQL<sup>2</sup>, which is used to extract information from RDF files [22]. [68], who developed the dbrec system, were some of the first researchers to use semantic web technologies in the field of recommender systems. Their recommender system takes advantage of an the linked data semantic distance (LDS) algorithm [69], as well as DBpedia, the ontological version of Wikipedia, to retrieve more details about artists. Another study [70] exploited user and item connections performed by [71] by converting the resulting RDF graph into a matrix of users-items using recommendation techniques. Di Noia et al. [53] utilized multiple semantically structured data from DBpedia [67], Freebase [72], and LinkedMDB [73] to produce movie recommendations by generating a matrix of subjects-predicates-objects, with the row being the subject and the column representing the object, whereas the cell is filled with the property, weighted using a genetic algorithm, if one exists. The TF-IDF technique is used to give weights for all matrix elements. The similarity measurement used to determine whether any two movies are related is cosine. Afterward, the similarity between the user profile and new movies is calculated and, hence, a movie is recommended.

Following are some projects inspired by the entity linking technique to represent data:

- (a) Babelfy<sup>3</sup>: [74] an integration of word sense disambiguation and entity linking.
- (b) DBpedia Spotlight: this project uses DBpedia to link unstructured text to an LOD cloud [75].

---

<sup>2</sup><https://www.w3.org/TR/rdf-sparql-query/>

<sup>3</sup><http://babelfy.org>

- (c) Alchemy<sup>4</sup>: provides the ability to utilize NLP for the sake of analyzing different types of documents to reach entity recognition.
- (d) Open Calais<sup>5</sup>: Analyzes documents using both the ML and NLP technique to locate entities.
- (e) NERD: An ontology used to merge the above-listed types of entity-linking projects [76].

### 2.2.3.3 Bottom-Up Approach:

This approach relies on the distributional hypothesis, where words and documents are distributed in the space as vectors, and every two words that are close in meaning in the context are also close in the corresponding vector space [1]. Discriminative models (DMs) inspired the notion that any word can be understood by humans by looking at the context where that particular word occurs [1] (e.g., leash, dog, animal [77]). Similarly, words that occur in similar contexts are more likely to also be closer in meaning (e.g., leash, muzzle) this approach is called the distributional hypothesis [78]. The word-context matrix technique is used in DMs instead of the word-document matrix technique presented by VSM [1]. Moreover, [79] explained how vector space models (VSM) can be used to illustrate semantics in the following:

- A term-document matrix.
- A word-context matrix.
- A pair-pattern matrix.

Since DMs utilize similarity measurements, such as cosine, euclidean, and relative entropy-based measurements, these models are known as geometrical models [80]. [81] introduced WordSpace, where similarity depends on an unsupervised method that leads to an expansion in the number of dimensions of the vector space. Hence, a dimensionality reduction technique is needed.

#### 1. Dimensionality reduction techniques:

The idea behind this technique is to reduce the number of dimensions in the vector space [1].

[65] stated that a dimensionality reduction technique is applied to latent semantic indexing

---

<sup>4</sup><http://www.alchemyapi.com/>

<sup>5</sup><http://www.opencalais.com/>

(LSI), which is a method of creating a semantic vector space by using the singular value decomposition (SVD) technique. First, a matrix is formed in which the rows represent words and columns represent documents. Then, it is decomposed into two matrices to reduce the number of dimensions. The LSI technique outperforms other techniques, and this is proven in [82, 83]. Moreover, [84] used a dimensionality reduction technique to refine the user profile in CBRs by reducing the number of features of the user's profile, which produced better recommendations. Nevertheless, this technique could be outperformed by other techniques when working with a small dataset or short texts [85].

## 2. Modeling Negation:

The VSM technique suffers from the problem of not considering negative feedback when building a user profile [1]. However, works such as [86, 87] overcame the issue of negative relevance feedback by subtracting irrelevant vectors to refine the user profile.

## 3. Conclusion and Comparison of approaches:

The two methods aim to solve problems that occur in CBRs, such as limited content analysis and over-capitalization. The top-down approach tends to use external knowledge sources such as ontology, encyclopedias, and LOD clouds to better understand the user's interests to make more accurate recommendations. In contrast, the bottom-up technique starts by analyzing individual terms in a broad context to extract the underlying semantics [1].

Transparency is one factor that differentiates the two approaches. In the top-down technique, user profiles and items are explicitly represented, which decreases ambiguity, resulting in a better estimation of the semantic similarities between an item and a user profile. This will help in making good explanations. However, the bottom-up approach lacks transparency since it analyzes the meaning of a word by looking at its co-occurrences throughout a set of documents [1].

Another difference between the two approaches is that the bottom-up method is superior because it utilizes a dimensionality reduction technique [1]. Lastly, the two methodologies' pros and cons are presented in Table 4.1 [1].

## **2.2.4 Collaborative Filtering (CF)**

### **2.2.4.1 Introduction**

Collaborative filtering focuses more on rating data than the content descriptions of items [17]. In contrast with content-based techniques, there is no need for detailed information about both users and items, which is a major advantage of this method. Consequently, computation complexity will be much lower.

The Netflix competition conducted in 2006 led to major developments in the field of recommender systems, especially collaborative filtering [88], because huge datasets became available to the research community (around 100 million movie rating).

Recommender systems rely on two types of feedback. One is explicit feedback, where users clearly declare their opinion on an item, such as rating a movie [88]. The other type is implicit feedback; although it is less accurate, it is still important. Examples of implicit feedback are the user's search history, purchase history, and mouse movement [89].

Collaborative filtering depends on two approaches when making recommendations: latent factor models, such as SVD, and neighborhood methods [88].

### **2.2.4.2 Neighborhood-based Recommendation Methods**

In recent years, the online shopping rate has increased rapidly, creating retailer websites, such as amazon.com, with a wide variety of items. In recommendations, the ratings of other users are considered in the process of building the RS model, in contrast to content-based models, which rely on an item's specifications [17]. Some advantages of neighborhood-based recommendations include: [90]:

- Simplicity.
- Justifiability.
- Efficiency.
- Stability.

In the following sections, we explore different types of neighborhood styles:

### 2.2.4.3 User-based Rating Prediction

When a user rates an item, the most similar users' interests are considered when recommending the next item. This technique is called the nearest neighbor [50] [91]:

$$\hat{r}_{ui} = \frac{1}{|N_i(u)|} \sum_{v \in N_i(u)} r_{vi} \quad (2.6)$$

$N_i(u)$  is the  $k$ -nearest neighbors of user  $u$  who rated item  $i$ , while  $w_{uv}$  is the similarity between  $u$  and  $k$  users  $v$ .  $r_{ui}$  is the rating of item  $i$  by user  $u$ . Normalization results in

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i(u)} w_{uv} r_{vi}}{\sum_{v \in N_i(u)} |w_{uv}|} \quad (2.7)$$

Note that the denominator, an absolute value, is used to ensure a positive number.

### 2.2.4.4 Item-based Recommendation

In this approach, items that are similar to the item the user has liked are considered in the process of [92]. The formula is expressed as follows:

$$\hat{r}_{ui} = \frac{\sum_{j \in N_u(i)} w_{ij} r_{uj}}{\sum_{j \in N_u(i)} |w_{ij}|} \quad (2.8)$$

$j$  and  $i$  are items, and  $N_u(i)$  denotes the items that are rated by user  $u$  and similar to item  $i$ .  $w_{ij}$  is the similarity between item  $i$  and  $k$  items  $j$ .

### 2.2.4.5 Matrix Factorization Models

Matrix factorization (MF) is a powerful family of techniques used to build recommender systems [19]. MF aims to learn latent space vectors  $p$  and  $q$  for each user and item, respectively. Figure 2.3 shows a flowchart of the MF method in making recommendations.

The idea is to factorize the rating matrix into lower dimensional spaces using a given number of latent features such that the dot product of the two latent space representations should approximate the original ratings, in addition to predicting the ratings of unseen items, as shown in Equation 2.9

$$\hat{r}_{ij} = p_i q_j^T \quad (2.9)$$

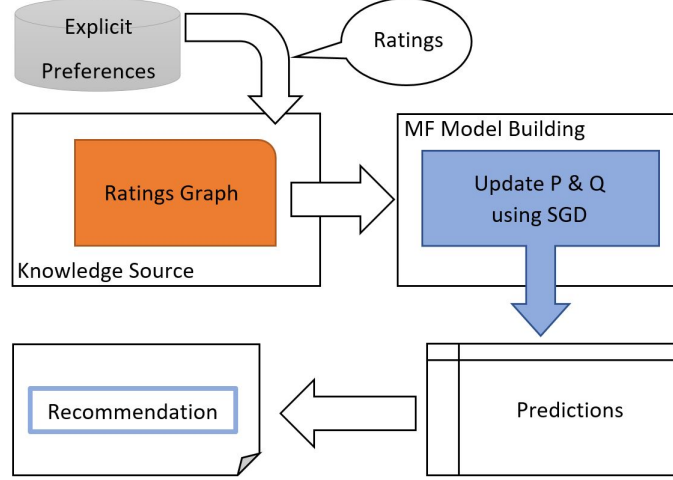


Figure 2.3: Matrix Factorization Flow Chart.

$\hat{r}$  denotes the training set of known ratings, where  $i$  and  $j$  represent a user and an item, respectively. Latent space vectors  $p$  and  $q$  are found by minimizing the following objective function over known ratings [19]:

$$J = \sum_{i,j \in R} (r_{ij} - p_i q_j^T)^2 + \beta (\|p_i\|^2 + \|q_j\|^2) \quad (2.10)$$

A regularization term exists for each unknown parameter (latent vector) to avoid over-fitting, with  $\beta$  being a regularization coefficient that controls the smoothness of the newly added term.  $J$  is not convex with respect to all the unknown parameters, but it is convex with respect to either  $p$  or  $q$  alone. Therefore, stochastic gradient descent [93] is used to solve for the optimal parameters. The update rules for the user and item latent factor parameters  $p$  and  $q$  are given by

$$p_u^{(t+1)} \leftarrow p_u^{(t)} + \alpha (2(R_{u,i} - p_u^{(t)}(q_i^{(t)})^T)q_i^{(t)} - \beta p_u^{(t)}) \quad (2.11)$$

$$q_i^{(t+1)} \leftarrow q_i^{(t)} + \alpha (2(R_{u,i} - p_u^{(t)}(q_i^{(t)})^T)p_u^{(t)} - \beta q_i^{(t)}) \quad (2.12)$$

Next, we will explore two matrix factorization approaches presented by [94, 95], who improved the basic matrix factorization technique, which we described above, and came up with new approaches, JMF and SemJMF. Then, we will explain three other types of matrix factorization models used in the field: the SVD, SVD++, and time-aware factor models.

### 1. Joint Matrix Factorization:

[94] extended [19]’s work by incorporating not only known ratings but also information that comes in two forms, from the user’s side and the item’s side. Side information, such as age and gender, come from the user side, although they are not always available due to privacy issues. Genre, size, color, and movie stars are item-side information, and they are almost always available. [94] took advantage of this additional information to enhance the accuracy of recommendations. They extended the basic matrix factorization cost function to create a joint MF (JMF) that includes additional terms for item-side information. Since they conducted their experiments in the movie domain, they used two types of movie information, mood and plot keyword. Two movie-by-movie similarity matrices were constructed using two similarity methods to be added as new terms to the cost function. They compared their work to several non-context aware approaches and showed that their approach outperformed other approaches by 10

## 2. Semantic Joint Matrix Factorization :

[95] extended [94]’s idea of using item-side information to exploit the power of the semantic web. Working in two domains, music and movies, they extracted semantic information from the DBpedia <sup>6</sup> dataset, which is a semantic version of Wikipedia. They retrieved artist category information from DBpedia using SPARQL to enrich the item-side information. Like [94], they constructed a new matrix for the semantic information, added a new term to the JMF cost function, and obtained better results in comparison to JMF and other techniques.

## 3. Asymmetric Matrix Factorization :

Building low dimensional representations of users and items using multiple sources of knowledge was explored by [96]. In their work, they succeeded in building a model to annotate images using the so-called bag-of-features that belong to images and non-negative matrix factorization (NMF) to build a low dimensional latent vector representation. Later, this approach was used in [97] to propose a solution for the item cold-start problem in collaborative filtering using matrix factorization and utilize multiple domains in the process of building the model. More specifically, item content features, such as genre, are used to build the items’ latent space before learning the user’s latent space using another domain, namely the known

---

<sup>6</sup>dbpedia.org



ratings. Although this approach integrated two sources of data to overcome the cold-start problem, it did not provide explainable recommendations.

#### 4. SVD:

SVD works by mapping items and users to a joint latent factor space within dimensionality  $f$ , making this interaction between users and items modeled as inner products [88].

The rule that is used to predict the rating is expressed as follows:

$$\hat{r}_{ui} = u + b_i + b_u + q_i^T p_u \quad (2.13)$$

where  $\hat{r}_{ui}$  is the estimated rating for item  $i$  by user  $u$ .  $b_i$ ,  $b_u$ ,  $p_u$ , and  $q_i$  are the model parameters. The regularized squared error is minimized to learn the model parameters determined by cross-validation:

$$\min_{b_*, q_*, p_*} \sum_{(u,i) \in K} (r_{ui} - u - b_i - b_u - q_i^T p_u)^2 + \lambda (b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2) \lambda \quad (2.14)$$

$\lambda$  is a parameter that controls the regularization term. The optimization technique used is stochastic gradient descent [93].

## 2.2.5 Explanations

### 2.2.5.1 Introduction

Recommendations, by themselves, aim to guide the user's next move (e.g., which movie to watch next or what item to purchase). [98] has stated that the recommendation process happens inside a black box, meaning that users are not aware of why certain items are recommended to them. Thus, adding more clarifying details is desirable. When a website, such as Netflix.com, recommends a movie to a user and attaches a sentence such as "People who watched this movie also watched . . .)," the user will be encouraged to watch the recommended movie [99]. It is proven that explanations play a major role in gaining user trust and enhancing scrutability, which verifies the recommendation's validity [13].

In the 1990s, explanations were used by so-called expert systems. However, there were no satisfying evaluation techniques for these explanations [100].

### 2.2.5.2 Explanation Styles and Related Approaches

[101] and [99] proposed several explanation styles. Table 2.1 summarizes them.

Table 2.1: Explanation Styles.

[101]'s Styles	[99]'s Styles
Neighbor Style Explanation	Collaborative-based Style Explanation
Influence Style Explanation	Content-based Style Explanation
Keyword Style Explanation	Case-based Reasoning Style Explanation
	Knowledge and Utility-based Style Explanation
	Demographic Style Explanation

Following is a review of some existing systems that use different types of explanation styles in addition to an elaboration of the related studies mentioned in Table 2.2. Herlocher et al. [98] stated that explanations enhance the performance of collaborative filtering recommender systems. In their work, they explored 21 explanation interfaces where they eliminated the recommended items and only retained the explanations for users to choose from and found that, from a promotion point of view, the best interface was the histogram-like explanation interface according to the users' feedback. Other interfaces include past performance, a table of neighbors' ratings, similarity to other movies rated, and other information.

Pineda and Brusilovsky [102] discuss the transparency issue in the educational domain. The concentration is not on the interest of the user, such as in the movie or e-commerce domain, but it is in the level of education, which makes it more challenging. Therefore, the recommendation and explanation process requires more effort to estimate the skill level of the students. The focus of this paper [103] is on the transparency of the hybrid recommender systems. They attempt to overcome this issue by generating visualized and personalized explanations for the outputs. The music domain was used to test the approach by conducting a user study. Sato et al. [104] claim that current explanation styles take advantage of similarities between users, items, items' contents, and demographics. However, context like accompanying persons and usage scenarios are used to generate a different explanation style.

Another study [10] used community tags to explain recommendations. They categorized expla-

nations into three types: 1) item-based, where explanations were created based on similar items, 2) user-based, where the system relied on similar users to explain its recommendations, and 3) feature-based, where various features, such as genre, were used to justify the output. The authors of this work used the KSE explanation style. An example of an explanation could look like "*This movie is being recommended to you because it is tagged with **mystery** , which exists in movies you've liked previously*". In a similar study that also used KSE as the explanation style [105], a CBF model was designed for recommending digital cameras. They used specific characteristics of cameras, such as memory size and resolution, and allowed each user to choose which set of features met his or her requirements. [106] built a CF recommender system that relies on the latent factor model technique to produce accurate recommendations, along with explanations that are generated using a sentiment analysis of users reviews.

[101] defined the three explanation types mentioned in Table 2.1. In their work, they produced a book recommender system, called LIBRA, which was an extension of [107]'s work that used a CBF technique. A hybrid model was created using user ratings and item content data. Explanation effectiveness was measured by promotion and satisfaction. Promotion refers to an explanation's ability to convince users to choose the recommended items, while satisfaction means that an explanation allows users to check an item's quality. In their study, they used all three explanation styles listed in the left-hand column of Table 2.1.

[108] evaluated the effectiveness of explanations in terms of how well they helped users make better decisions. They concluded that personalizing feature-based explanations is unfavorable from an effectiveness perspective, although it may increase satisfaction. In contrast to [98], [101] argued that KSE and ISE are better than NSE because the latter suffers from a bias toward top recommended items, which causes an overestimation of the recommended items. This issue does not exist in KSE and ISE.

In [109], the authors designed a recommender system that recommends places for tourists to visit. They grouped users demographically based on their age (e.g., children, adults, and the elderly) and suggested places to each group with justifications that suit their interests. [110] proposed a new recommender system that reads news articles to users using synthesized speech. The system receives voice feedback from users to improve its performance and generate explanations using the KSE explanation style. Symeonidis et al. [111] proposed a system that leverages explanations

aimed toward increasing transparency without sacrificing accuracy. They were motivated by the limitations of some e-commerce recommender systems that depend on ratings and user behavior to explain recommendations but ignore item features. To overcome this limitation, they constructed a feature-based user profile where each user's interests are represented by item content features. They also created biclusters, which are groups of users who exhibit similar ratings of groups of similarly rated items and used the clusters to find possible patterns of interests and preferences between test users and the group of users. A user study was conducted to show that their approach resulted in higher satisfaction compared to other approaches after showing the justification.

The approaches listed in Table 2.2 are reviewed in this section. First, eight studies are both explainable and semantic-aware, whereas the next seven research items are only explainable with no semantic-awareness (i.e., do not use semantic web technologies, such as RDF, OWL, or SPARQL). The next eight studies in Table 2.2 use semantics but omit explanations. Lastly, the four remaining works in this table are neither non-explainable nor semantic-aware.

The first work was conducted by [2], who used a linked data semantic distance (LDS) algorithm [69] to build a model that recommends movies. More details about this method were presented in the previous subsection 3. [68] used property values to explain why a certain artist was recommended. Following is an example of their explanation: *Johnny Cash and Elvis Presley share the same value for 'death place': Tennessee..* Also, Figure 2.4 shows a snapshot of an example.

The approach utilized by [112] exploited linked open data (LOD) to build a movie recommender system, and Section 3 presents additional details of this system. Similar to [2], the values of properties of recommended items are used for explanations. Di Noia et al. emphasized that CBF systems provide more transparency than CF systems. However, in the current study, we are proposing a Hybrid CF system using MF that is accurate and transparent enough for users to accept recommendations by generating explanations through semantic web technologies. TasteWeights [3] is an interactive hybrid recommender system designed for the music domain <sup>7</sup>. Several sources of information, such as Twitter, Facebook, and Wikipedia, are utilized as a data source for the recommendation process. In addition to generating a visual interactive interface that provides justifications to users, the explanation interface allows users to choose the source of the explanation. If the user chooses to see an explanation based on Facebook data, then users will see their friends who liked

---

<sup>7</sup>A demo video is available at <http://bit.ly/TasteWeights>

Approaches	Year	Domain	CF	CBF	HYB	WB	BB	Explanation	Semantic-aware	User Study	CSP
dbrec [2]	2009	Music	x	✓	x	✓	x	✓	✓	x	x
LOD-RS [112]	2012	Movies	x	✓	x	✓	x	✓	✓	x	x
TasteWeight [3]	2012	Music	x	x	✓	✓	x	✓	✓	✓	x
Entity-based-RS [4]	2017	Movies	x	✓	x	✓	x	✓	✓	x	x
SEP [5]	2018	Movies	x	x	✓	✓	x	✓	✓	x	x
RecExp [6]	2016	Movies, Social	x	x	✓	✓	x	✓	✓	x	x
SemRec [113]	2018	Movies, Yelp, Books	x	x	✓	✓	x	✓	✓	x	✓
LOD-TRS [7]	2019	Movies, Books, Music	x	✓	x	✓	x	✓	✓	✓	x
SemAuto [8]	2018	Movies	✓	x	x	x	✓	✓	✓	✓	✓
EMF [13] [14] [15]	2017	Movies	✓	x	x	x	✓	✓	x	✓	✓
MoviExplain [9]	2009	Movies	x	x	✓	✓	x	✓	x	✓	x
Tagsplanations [10]	2009	Movies	✓	x	x	✓	x	✓	x	✓	x
Hybrid-RS[11]	2013	Movies	x	x	✓	✓	x	✓	x	✓	✓
TEM [114]	2018	Tourism	x	x	✓	x	✓	✓	x	x	x
Embedding-RS [12]	2018	Amazon	✓	x	x	x	✓	✓	x	x	x
RippleNet [115]	2018	Movies, Books, News	✓	x	x	x	✓	✓	x	x	✓
Sem-JMF [95]	2016	Music	✓	x	x	x	✓	x	✓	x	x
MORE [116]	2012	Movies	x	✓	x	✓	x	x	✓	x	✓
HealthRecSys [117]	2017	Health	x	✓	x	✓	x	x	✓	✓	x
HeteRecom [118]	2012	Movies	x	✓	x	✓	x	x	✓	x	x
SemSim [119]	2018	Movies	✓	x	x	✓	x	x	✓	x	✓
HeteMF [120]	2013	Movies	x	x	✓	x	✓	x	✓	x	✓
PLDSD [121]	2017	Movies	x	✓	x	✓	x	x	✓	x	x
Social-RS [122]	2012	Movies	x	x	✓	✓	x	x	✓	x	✓
SocialMF [123]	2011	Social	✓	x	x	x	✓	x	x	x	x
AMF [96]	2010	Images	✓	x	x	x	✓	x	x	x	✓
ConvMF [124]	2016	Movies, Amazon	x	x	✓	x	✓	x	x	x	x
E-Learning-RS [125]	2015	E-Learning	x	✓	x	✓	x	x	x	x	✓
PMF [126]	2007	Movies	x	x	✓	x	✓	x	x	x	x
ERBM [127]	2017	Movies	✓	x	x	x	✓	✓	x	✓	✓
ASEMF [128]	2018	Movies	✓	x	✓	x	✓	✓	✓	✓	✓
SeER [129]	2019	e-learning	x	x	✓	x	✓	✓	✓	x	✓
<b>This work</b>	<b>2019</b>	<b>Movies, Books</b>	<b>x</b>	<b>x</b>	<b>✓</b>	<b>x</b>	<b>✓</b>	<b>✓</b>	<b>✓</b>	<b>✓</b>	<b>✓</b>

Table 2.2: A comparison of some recommender systems based on the (a) domain, (b) RS style, (c) availability of explanation, (d) system awareness of semantics, (e) user study, and (f) CSP solution to the cold start problem.

the recommended item as an explanation. The same output happens when Wikipedia or Twitter is chosen. The system consists of three layers. The first one contains users' liked music gathered from the user's Facebook page. The second layer is the content layer where items' features are listed from all three information sources (i.e., Wikipedia, Facebook, and Twitter). The third layer is the recommendation layer that shows the top recommended items. When retrieving information from Wikipedia, the semantic version of it, DBpedia, is used to perform the task using the query language SPARQL. The authors indicate that as Herlocker [98] and Middleton [130] emphasized previously, an explanation increases the acceptance of a recommendation, and an explanatory interface also helps users understand why certain recommendations are shown for them. It also encourages users to get educated and involved in the recommendation process. Thirty-two real users participated in a user study to evaluate the system's performance and how well the explanation interface helped them understand the recommendation process; see Figure 2.5. The authors concluded that although Wikipedia, when it was the source of the explanation, was more accurate than both Facebook and Twitter, explanations based on Facebook friends was favored by users due to trust in their friends' interests and tastes.

Catherine et al. [4] proposed a white box recommender system that explains its output by using predefined rules. For example, if user  $U$  likes movie  $M$ , and movie  $M$  is linked to entity  $E$  then, user  $U$  likes entity  $E$ . The ProPPR [131] technique is used to rank both items and their entities for recommendations and explanations. They argued that knowledge graphs, in addition to increasing accuracy, helped generate more convincing explanations. An example is shown in Figure 2.6.

Another approach that is explainable and semantic-aware is [5], where they proposed a post-hoc mechanism to generate explanations. After building the recommender system, a unified heterogeneous information network (HIN) is built to provide justification for the recommended items. Explanation paths between the target user and other system components are established and ranked then used to show the explanations. To rank the explanation paths candidates, three ranking metrics are used, Credibility, readability, and diversity. An example of the explanation is shown in Figure 2.7.

Hu et al. [6] emphasized the importance of explanations in recommender systems. In their approach, they relied on HIN [132] to generate semantic and justifiable recommendations. Figure 2.8 illustrates the explanation style. Another study that relied on the HIN technique to build a

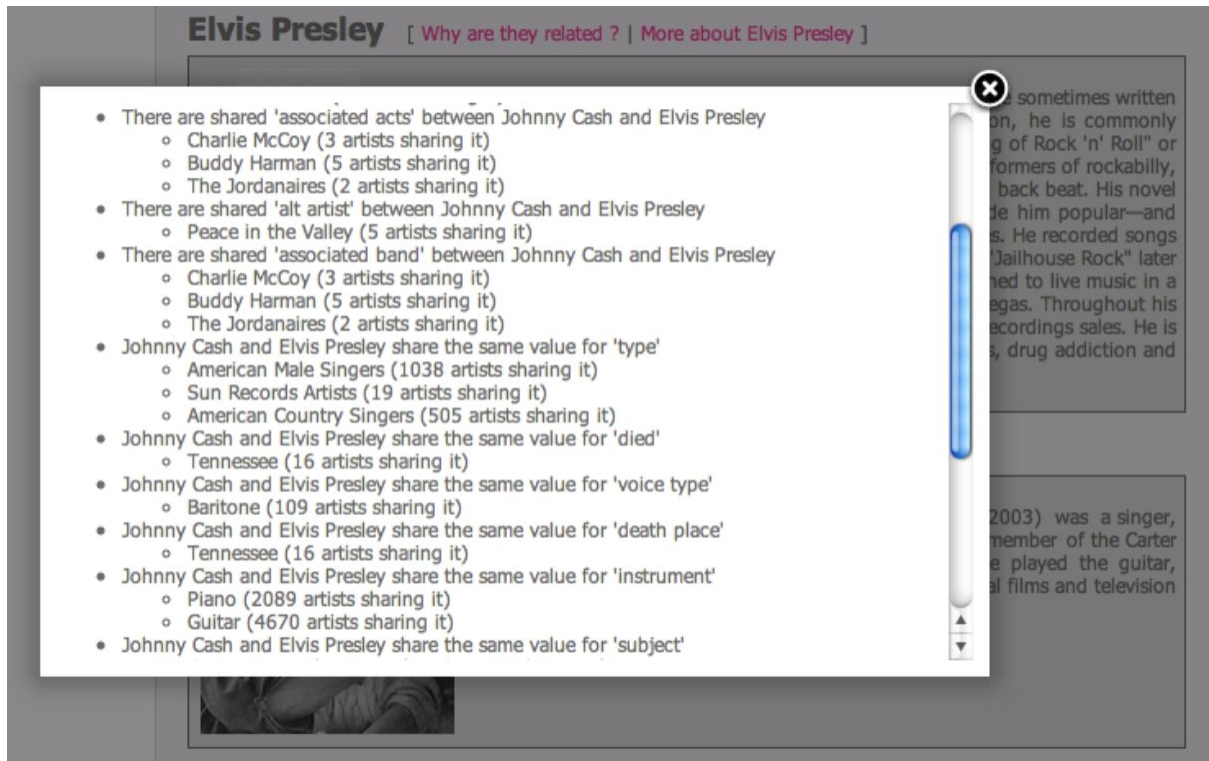


Figure 2.4: An Example of dbrec's Output, Source: [2].



Figure 2.5: An Example of TasteWeight's Output, Source: [3].

- (1) bridge\_of\_spies, score =  $0.4 + 0.3 = 0.7$ , reasons = { tom\_hanks, drama\_thriller }
- (2) snowden, score =  $0.3$ , reasons = { drama\_thriller }
- (3) inferno, score =  $0.4 - 0.2 = 0.2$ , reasons = { tom\_hanks, (-ve) crime }

Figure 2.6: An Example of Entity-based RS' Output, Source: [4].

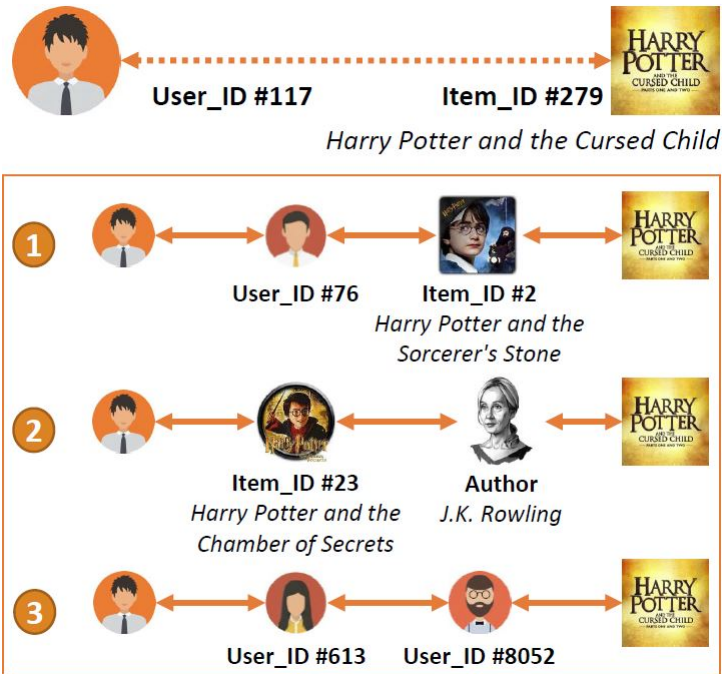


Figure 2.7: An Example of SEP's Output, Source: [5].

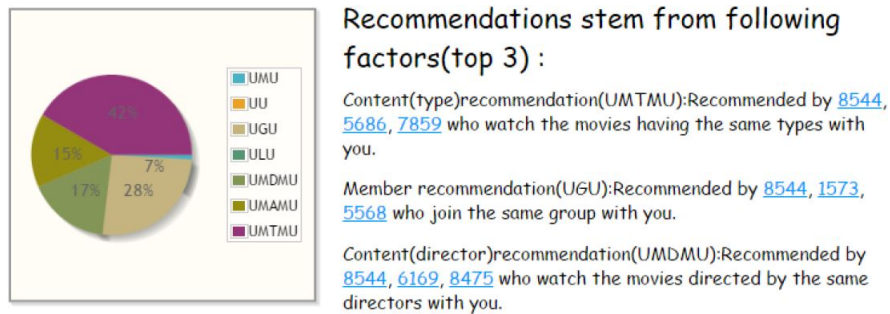


Figure 2.8: An Example of RecExp's Output, Source: [6].

recommender system is SemRec [113]. In this work, the meta-paths obtained from the HIN are personalized and prioritized to accommodate users' preferences. The cold-start problem is resolved in this work, and they stated that their model outperformed other baseline methods in terms of producing a lower error rate. A study conducted by [7] shed light on the significance of natural language explanations in recommender systems and how linked open data can empower them by linking the user's previously preferred items and items' attributes to the new recommendations. The explanation mechanism is based on the notion that descriptive properties that describe the items that the user liked in the past can serve as explanations for the outputs of the recommender system. A user study was conducted to evaluate the system, and the results show that the proposed system succeeded in



Explanation for each single item:	Explanation for the whole group:
I propose you <b>Drawing Restraint 9</b> because you sometimes like <i>movies produced by Barbara Gladstone</i> , as <b>Women Without Men</b> .	I recommend you <b>Hatchet III</b> and <b>The Gene Generation</b> because you sometimes like <i>movies whose music was composed by Scott Glasgow</i> , as <b>Poker Night</b> .
I suggest you <b>The Gene Generation</b> because you sometimes like <i>movies whose music was composed by Scott Glasgow</i> , as <b>Poker Night</b> .	Moreover, I propose you <b>Facing Windows</b> because you sometimes like <i>movies starred by Raoul Bova</i> , as <b>Avenging Angelo</b> .
I recommend you <b>Hatchet III</b> because you sometimes like <i>movies whose music was composed by Scott Glasgow</i> , as <b>Poker Night</b> .	Next, I suggest you <b>The Grudge 3</b> because you sometimes like <i>movies starred by Beau Mirchoff</i> , as <b>Poker Night</b> .
I propose you <b>The Grudge 3</b> because you sometimes like <i>movies starred by Beau Mirchoff</i> , as <b>Poker Night</b> .	Lastly, I recommend you <b>Drawing Restraint 9</b> because you sometimes like <i>movies produced by Barbara Gladstone</i> , as <b>Women Without Men</b> .
I suggest you <b>Facing Windows</b> because you sometimes like <i>movies starred by Raoul Bova</i> , as <b>Avenging Angelo</b> .	

Figure 2.9: An Example of LOD-TRS' Output, Source: [7].

producing transparent recommendations and explanations. An example of their explanation style is shown in Figure 2.9.

The study of [8] focuses on the issue of explaining the output of a black box recommender system. In this work, the recommender system is built using Autoencoder Neural Network technique that is also aware of the Knowledge Graphs retrieved from the Semantic Web. The KGs are also adopted for the explanation generation. The authors claim that explanations increase the users' satisfaction, loyalty, and trust in the system. In this study, three explanation styles are proposed, popularity-based, pointwise personalized, and pairwise personalized. Figure 2.10 depicts an example of the explanation styles. For evaluation, an A/B test was conducted to measure transparency, trust, satisfaction, persuasiveness, and effectiveness of the proposed explanations. The pairwise method was preferable by most users more than pointwise method.

All of the previous studies involved white box systems; however, [15], a Ph.D. thesis, involved a black box system. This doctoral thesis investigated whether it is possible to generate explanations for the output of a black box system using a neighborhood technique based on cosine similarity. An MF system recommends items without being able to explain why due to its complete dependency on explicit user preferences. However, the proposed method succeeded in generating explanations using the above-mentioned technique. This allows explanation generation using the three styles listed in the right-hand column of Table 2.1. Because this study was used for comparison to our proposed work, we elaborate on the technical issues of this study in Section 2.2.5.3. MoviExplain

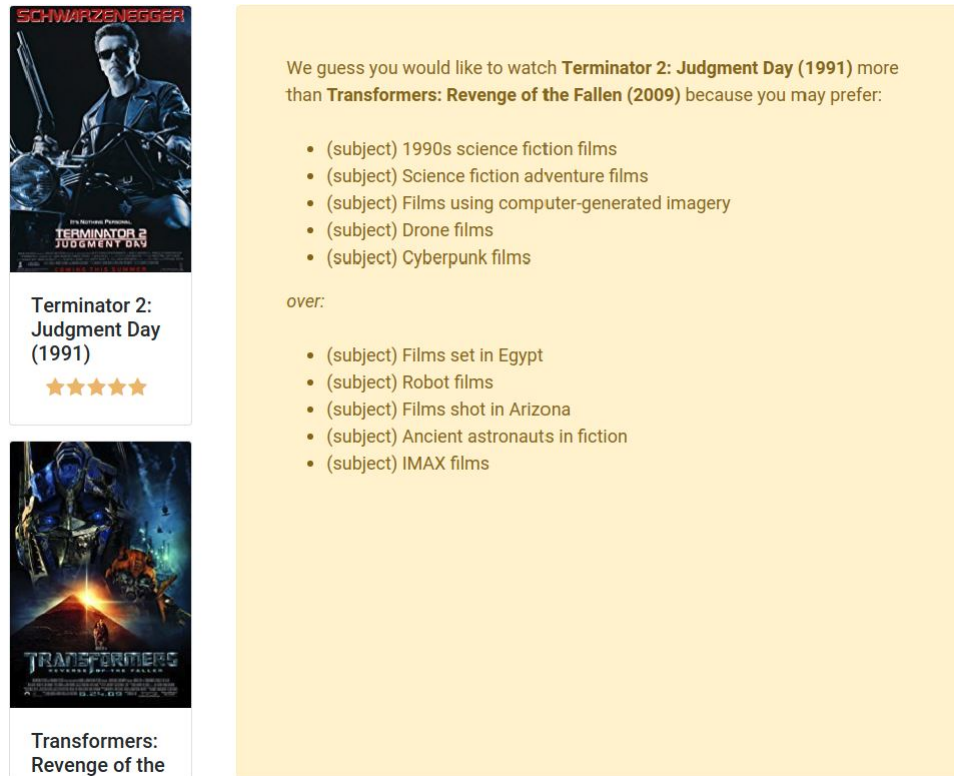


Figure 2.10: An Example of SemAuto's Output, Source: [8].

<sup>8</sup> is a project created by Symeonidis et al. [9]. It utilizes the idea of [111] where users are grouped into biclusters, which means each set of users are assigned to a set of movies. One benefit of this technique is that a feature, such as a genre, could be extracted from this assignment, leading to explaining the recommendation to users based on this feature. The styles of explanation used in this study were KSE and ISE in addition to a mixed style of the two previous styles, which they called KISE. An example of this system is shown in Figure 2.11. A user study was conducted in an attempt to justify their assumption that KISE's explanation style is better than the other two styles; they reached the conclusion that their proposed style was preferable by users more than the other two styles using various statistical metrics, such as mean, standard deviation, and Pearson correlation.

Tagsplanlation [10] is elaborated on in Section 2.2.5.2 and Figure 2.12, which presents an example. The next study is a Master's thesis written by Raza Ul Haq [11]. The author proposed a hybrid, white box, and explainable approach for recommending movies. In this study, both collaborative

<sup>8</sup><http://delab.csd.auth.gr/MoviExplain>

Our Justified Recommendations			
[Movie id]	[Movie title]	[The reason is]	[because you rated]
1526	Witness (1985)	Ford, Harrison (I)	21 movies with this feature
1273	Color of Night (1994)	Willis, Bruce	7 movies with this feature
1004	Geronimo: An American Legend (1993)	Hackman, Gene	7 movies with this feature
1442	Scarlet Letter, The (1995)	Oldman, Gary	7 movies with this feature
1044	Paper, The (1994)	Close, Glenn	7 movies with this feature
893	Casino (1995)	De Niro, Robert	6 movies with this feature
274	Sabrina (1995)	Pollack, Sydney	6 movies with this feature
1092	Dear God (1996)	Kinnear, Greg	5 movies with this feature

Figure 2.11: An Example of MovieExplain’s Output, Source: [9].



Figure 2.12: An example of Tagsplation’s Output, Source: [10].

and content-based filtering techniques were used and relied on additional information obtained from items and users. The author emphasized that interpretations are crucial in gaining customers’ trust and satisfaction. A user study was conducted that included fifty participants to test the system. The results show that most participants preferred to see justifications alongside the recommendations; see Figure 2.13 for an example recommendation. Wang et al. [114] discussed the necessity of side information in generating explanations. In this study, a combination of embedding-based and tree-based models was used to take advantage of good recommendations from the embedding-based model and transparency from the tree-based model. The evaluation showed that their system was superior to other baseline approaches.

The research of [12] focused on adding explanations to a black box recommender system by using structured knowledge bases. The system takes advantage of historical user preferences for producing accurate recommendations and structured knowledge bases about users and items for generating justifications. After the model recommends items, a soft matching algorithm is used, using the knowledge bases, to serve personalized explanations for the recommendations. They argue that their model outperforms other baseline methods. An example of the explanation is illustrated in Figure 2.14

**Recommended Movies For You**  
 Movie type: Romance Movies

1: Godfather, The (1972)



**Reasoning Behind To Recommend This Movie**  
**Popularity Among Other Users**  
 Average Rating of this Movie is: 4/5  
**Content Based Knowledgeable Justifications**

- Recommending you this movie because you like romnace movies
- Recommending you this movie because you like the actor(s) Marlon Brando
- Recommending you this movie because you like the director(s) Francis Ford Coppola

**Context Based Knowledgeable Justifications**

- Recommending you this movie because you have watched the movie on a weekend
- Recommending you this movie because you have watched romnace movies with a Significant other

**Interacting With the Recommender System - Improve Your Recommendations**  
 Give a Rating :

- I Like this Movie
- I want further recommendations on Romance movies

Figure 2.13: An Example of Hybrid-RS' Output, Source: [11].

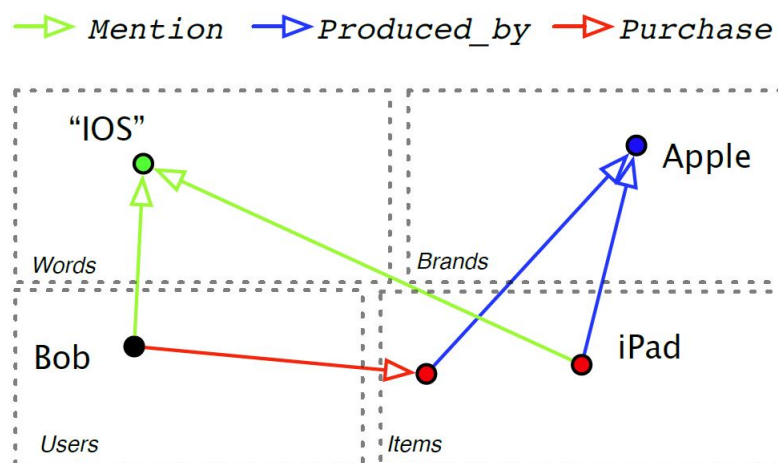


Figure 2.14: An Example of Embedding-RS' Output, Source: [12].

RippleNet [115] is an approach that used Knowledge Graphs in collaborative filtering to provide side information to the system in order to overcome the dilemma of sparsity and cold start. This black box system takes advantage of KGs, which are constructed using Microsoft Satori<sup>9</sup>, to better enhance the recommendation accuracy and transparency. They stimulate the idea of water ripple propagation on understanding the user preferences by iteratively considering more side information and propagating the user interests and showed that their model performed better than state-of-the-art models. In section 2, we discussed study number 15 [95] in Table 2.2 in detail. This study proposed a black box system that utilizes the semantic web to increase the system's accuracy. However, it lacks transparency due to the failure to generate explanations during the process of building the model.

MORE [116] is a Facebook application that recommends movies. The system leverages linked data from DBpedia to calculate movie similarities for use in the recommendation process, and by doing so, helping with the cold-start issue. They exploited the semantic version of the vector space model (sVSM) for similarity calculations, then used precision and recall to test the proposed model against baseline approaches. They claimed that the validity of their system is higher than other approaches.

Bocanegra [117] focused on building a semantic content-based recommender system for the health domain. The authors argued that the internet has an abundance of health-related videos that need to be managed correctly so customers can gain the most benefit from them. The proposed method exploits a couple of ontologies in the health domain to provide extra information for the recommendation process. The system was evaluated in a study of 26 health professionals, who agreed on the relevance of the recommended videos to their desires.

Zhou [118] extended the recommender system to explore not only historical user preferences but also the semantics of different items. They adopted the path technique of heterogeneous information network (HIN) to measure the relatedness of different objects. The system succeeded in providing semantic-aware recommendations in addition to recommending related items of same and different object types. Another study that used semantic web technologies in enriching the recommendation process is [119]. In this work, semantic information was used to determine the similarity of users and items as a step toward producing more accurate recommendations. The cold-start problem was

---

<sup>9</sup><https://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/>

resolved in this work, and the results show that their technique outperformed similar baseline techniques. A HIN is also used in [120] as a regularization term in matrix factorization. In addition to users' previous ratings, additional related information obtained from the HIN is effective in enhancing the recommendation process, leading to more accurate output. The evaluation process ensures the quality of the system in comparison to other approaches.

A doctoral study was conducted by Alfarhood [121] that investigated how the work of Passant [69] could be improved. Alfarhood stated that one drawback of an LDS algorithm is that it measures the similarity of two resources based only on their direct or indirect links. The author argued that engaging more indirect links could lead to a better result. Floyd-Warshall, an all pair shortest path algorithm [133], was employed to enrich the resources' similarity matrix by including more indirect links in the calculation of the semantic distance between resources. The system is called propagating linked data semantic distance (PLDSD). The final results confirmed that the proposed LOD-based recommender system produced more accurate recommendations than baseline methods.

Semantic web technologies increase the effectiveness of a recommender system [122]. Current syntactic-based recommender systems could improve their performances by considering semantics. The authors of [122] proposed a hybrid recommender system that leverages social networks in the recommendation process. The results reveal the proposed system performed more effectively in comparison to other baselines.

A black box approach was proposed by [123], who aimed to exploit social information and use it to improve the recommender system. The technique they used is the well-known matrix factorization algorithm [19], to which they added a new social regularization term. This new term introduces social information to MF as a social constraint. The authors stated that the system is domain free. The results of their experiments show that their system was superior to other baseline approaches. Another early black box approach was proposed by [96]. This approach was described previously in section 3.

Moreover, [124] emphasized that sparsity is a well-known problem in matrix factorization-based recommender systems, and existing approaches in the literature exploit external information, such as reviews and synopses to tackle this issue. However, this approach affects the understandability of the system. To overcome this issue, a convolutional neural network (CNN) and probabilistic matrix factorization were integrated, resulting in a context-aware recommender system (ConvMF).

This technique allows for capturing the side information that plays a major role in enhancing the accuracy of the system. With datasets from MovieLens and Amazon, the authors claimed that their system significantly outperformed other baseline systems.

The last study listed in Table 2.2 is [125]. This study focused on learning the semantic relatedness of concepts in the field of e-learning. The information was obtained from semantic versions of some encyclopedias and then the skip-gram model was applied to create concept vectors that allow the system to measure the semantic relatedness of concepts. Thus, recommendations could be offered to the user accordingly. The authors stated that the model is domain free and does not rely on historical data regarding users' preferences.

In the next subsection, we elaborate further on the details of EMF as described in [13] [14] [15], a doctoral study conducted by Behnoush Abdollahi of the University of Louisville.

### 2.2.5.3 Explainable Matrix Factorization (EMF)

In this section, EMF [13] [14] [15] is reviewed. This study introduced a solution to a known problem in matrix factorization, the black box, which means recommending items to users without the ability to explain the recommendations. Explaining recommendations encourages users to accept the recommendation. Therefore, the authors proposed a solution to this problem in the form of an explainable matrix factorization approach. They managed to explain the recommendations that matrix factorization produces while simultaneously calculating the predictions of missing values. They did this by constructing a new graph that represents the explainability score for each user for each item using the neighborhood technique, which bases explanations on the fact that items liked by close neighbors are more likely to be also liked by the target user. Hence, the recommendation comes with an explanation. They explained recommendations using two styles: user-based and item-based neighborhood style explanations (NSEs) as illustrated in Figure 2.15. [13] calculated an explainability score as follows

$$W_{ij} = \begin{cases} \frac{|N'(i)|}{|N_k(i)|} & \text{if } \frac{|N'(i)|}{|N_k(i)|} > \theta^n \\ 0 & \text{otherwise} \end{cases} \quad (2.15)$$

Movie	Your rating (1-5)
Batman	5
Twilight	5
Scream 2	4
Space Jam	3
Dead Man	5

Rating	Number of Neighbors
★	0
★★	0
★★★	6
★★★★	8
★★★★★	7

Figure 2.15: EMF, Source: [13] [14] [15] neighborhood style explanation examples; the example on the left is user-based, and the example on the right is item-based.

where  $N'(i)$  is the set of neighbors of user  $i$  who rated item  $j$ , and  $N_k(i)$  represents the list of  $k$  nearest neighbors of  $i$ .  $\theta^n$  is a threshold that determines whether item  $j$  is an explainable item for user  $i$ . The authors used the cosine similarity to calculate distances between users.

The objective function is as follows:

$$J = \sum_{i,j \in R} (r_{ij} - p_i q_j^T)^2 + \frac{\beta}{2} (\|p_i\|^2 + \|q_j\|^2) + \frac{\lambda}{2} (p_i - q_j)^2 W_{ij} \quad (2.16)$$

Here,  $r_{ij}$  is the rating of item  $j$  by user  $i$  and  $R$  is the matrix of user-item ratings,  $p_i$  is the user latent space,  $q_j$  is the item latent space, and  $\frac{\beta}{2} (\|p_i\|^2 + \|q_j\|^2)$  is a regularization term with coefficient  $\beta$  for weighting, and  $\lambda$  denotes the explainability regularization coefficient that insures a smooth inclusion of the new term to prevent over-fitting.  $W_{ij}$  is the explainability matrix.

For optimization, the stochastic gradient descent [93] technique is used to minimize the objective function. Hence, the update rules for  $p$  and  $q$  are as follows:

$$p_i \leftarrow p_i + \alpha(2(r_{ij} - p_i q_j^T) q_j - \beta p_i - \lambda(p_i - q_j) W_{ij}) \quad (2.17)$$

$$q_j \leftarrow q_j + \alpha(2(r_{ij} - p_i q_j^T) p_i - \beta q_j - \lambda(p_i - q_j) W_{ij}) \quad (2.18)$$

They conducted the experiment using the MovieLens 100k dataset, which consists of 943 users and 1,682 movies. Two evaluation metrics were used, RMSE and nDCG@10. Moreover, they compared the result with four collaborative filtering methodologies: standard NMF [19], probabilistic MF [126], user-based top-n CF [134], and item-based top-n CF [135]. The outcomes of EMF [13] [14] [15] outperformed most of the techniques. To evaluate explainability, they used mean explainability precision (MEP) and mean explainability recall (MER) metrics. Comparisons to the five baseline



techniques showed that EMF outperformed the other techniques. The explanation style was in the form of a bar chart showing the number of neighbors who rated the recommended item with a high score.

#### 2.2.5.4 Goals and Metrics

Table 2.3 contains the explanatory goals and their definition [99]. In our study, we focus on increasing the effectiveness of the explanation, which helps users make better decisions.

Table 2.3: Explanation objectives.

Aim	Definition
Transparency	Explain how the system works
Scrutability	Allow users to tell the system is wrong
Trust	Increase users' confidence in the system
Effectiveness	Help users make good decisions
Persuasiveness	Convince users to try or buy
Efficiency	Help users make decisions faster
Satisfaction	Increase the ease of use or enjoyment

#### 2.2.5.5 Explanation Evaluation

Three types of metrics are used to evaluate recommendation explanations [13, 15]: MEP, MER, and EF1 scores:

$$MEP = \frac{1}{|U|} \sum_{u \in U} \frac{|R \cap W|}{|R|} \quad (2.19)$$

$$MER = \frac{1}{|U|} \sum_{u \in U} \frac{|R \cap W|}{|W|} \quad (2.20)$$

$$xF - score = 2 * \frac{MEP * MER}{MEP + MER} \quad (2.21)$$

$U$  represents the total number of users, while  $R$  is the set of recommended items, and  $W$  denotes the set of explainable items. MEP computes the ratio of the number of simultaneously recommended and explainable items to the total number of recommended items across all users. Similarly, MER calculates the ratio of the number of simultaneously recommended and explainable items across the total number of explainable items and, again, across all users.

## 2.2.6 Inference of Facts

In this section, we review the literature of inference of facts and rules that have been applied in the field. In artificial intelligence, inference means deriving new facts from historical ones [136]. When the system asks the knowledge base a question or sends a query, the knowledge base of the system that is built on historical knowledge should answer based on what it already knows. Inferring the right answer is like finding a needle in a haystack [136]. Uncertainty in AI means acting while having an incomplete observation or less determinism, which will cause the system to be in an uncertainty state. If a student is planning to graduate in four years  $G_4$ , this does not imply that he or she will certainly graduate as planned because other factors, such as financial difficulties, physical and mental health issues, and other factors, may cause a delay [136].

In the semantic web, inference rules are used to derive new knowledge from known facts [137]. For example, if  $A$  is of type  $B$ , while  $B$  is of type  $C$ , then consequently,  $A$  is of type  $C$ . This example shows complete certainty; however, uncertainty is impossible in some cases. To handle the uncertainty issue, Bayesian inference is applied to compute  $P(A \cap B) = P(A/B) * P(B)$ . For example, if a *person* has the *flu*, and the *flu* is a *disease*, then the *person* may have a *disease*.

The inference concept has been adopted in various studies involving recommender systems. Users' preferences are captured either explicitly via ratings or implicitly by inferring them [138]. In this study, the inference technique is applied using a weighted ordered weighted average (WOWA) operator [139] where concepts are given weights according to their distances from each other. [140] stated that collaborative filtering-based recommender systems produce subjective predictions because they rely solely on ratings. Therefore, the use of inference rules that take advantage of objective metadata leads to more personalized and context-aware recommendations. One of the objectives of this study is to increase the accuracy of the recommendations using inference rules. The following is an example of an inference rule: *If the rating of aspect A of the rated item i is 5 and the aspect of item j is the same as that of the rated item i then increase the predicted rating of item j.* This example illustrates that if the user likes an item, then he or she may also like specific aspects of that item, resulting in an increase in the predicted rating of items that share these aspects. [16] adapted the notion of inferred rules by designing a recommender system that infers the desire of the user using input data from the user and then recommending a possible match. The system was

The screenshot displays a web interface for a camera recommender system. At the top, the selected category is 'Photo Cameras'. A section titled 'Study searching attributes' allows filtering by Price (checked), Weight, and Length. A 'Study' button is present. Below this, an 'Inferred Association Rules' box contains the text: 'At MUCH LESS Price, there are products at [MUCH LESS] Weight AND [EQUAL or MORE or MUCH MORE] Length (49 times, 83.0% of the rules); There are 49 products at MUCH LESS Price that you search, at a much lower value of Weight specified, AND at the same or a higher or a much higher value of Length specified'. The main content area is titled 'Photo Cameras' and shows a grid of product recommendations. A 'Filter by:' bar categorizes products into 'Very Recommendable', 'Recommendable', and 'Not Much Interesting'. The products listed include Sony Cyber-shot DSC-T200, Canon Ixus 970 IS, Olympus MJU 770 SW, Panasonic Lumix DMC TZ5, Canon PowerShot A650 IS, and Panasonic Lumix DMC-FX100EG-S. Navigation links '1 2 3 4 5 6 7 8 9 10 Next' are at the bottom.

Figure 2.16: Snapshot of the camera recommender system, Source: [16].

designed to recommend the best camera for a user. The system asks the user for the desired features, such as the price, weight, and length and then calculates the distance between each product in the dataset and the specifications given by the user. Based on this calculation, a list of recommendations is shown to the user and divided into three categories, very recommendable, recommendable, and less recommendable. Figure 2.16 depicts a snapshot of the system.

## 2.3 Summary

To summarize, in this chapter, we summarized the techniques commonly used in recommender systems, such as collaborative, content-based, and others, as well as the role of the semantic web in the field of recommender systems. Lastly, a review of explanations' impact on recommender systems was presented.

## CHAPTER 3

### PROPOSED WORK

#### 3.1 Introduction

Our goal is to integrate semantic web technologies into the process of building a recommender system model that uses a matrix factorization technique to produce more meaningful explanations. Thus, we propose four approaches that aim to enrich user- and item-side information by leveraging data from the semantic web: asymmetric semantic explainable matrix factorization (ASEMF) [128] (sec. 3.2), semantic explainable matrix factorization (SemEMF)(sec.3.3), merged semantic explainable matrix factorization (MergedSemEMF)(sec. 3.4), and linked data semantic distance matrix factorization (LDSDMF) (sec. 3.5). See Figure 3.1 for an illustration and a comparison of the proposed methods to other approaches. We published some of this research in KDIR 2018 [128] and KDIR 2019 (to appear in Proceedings of KDIR 2019).

##### 3.1.1 Semantic Knowledge Graphs (KGs)

The web is abundant with information that is being harvested and structured into (Knowledge Graphs). KGs are extensive networks of objects, along with their properties, their semantic types, and the relationships between objects representing factual information in a specific domain [141]. Examples of KGs are DBpedia [142], Freebase [72], Wikidata [143], YAGO [144], NELL [145], and the Google Knowledge Graph [146]. In this work, DBpedia is used to build the desired KGs about users, items, and semantic attributes. In order to study the effect of one item semantic attribute or semantic feature (e.g. actor, author, etc) when building the semantic KG in increasing the transparency of the system, one semantic feature information is used in the three proposed

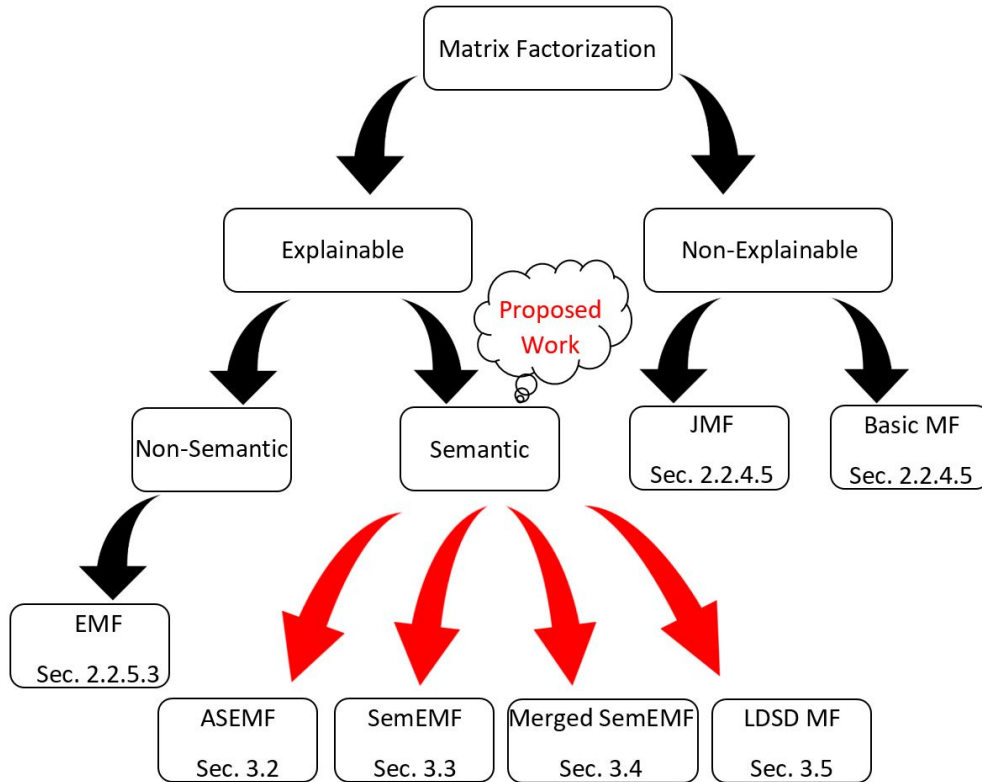


Figure 3.1: Proposed methods and other approaches.

models ASEMF [128] (3.2), SemEMF (3.3), and MergedSemEMF (3.4). However, in the last proposed model, LDSDMF (3.5), more influential semantic attributes (subject(s), actor(s), director(s), writer(s), author(s), publisher(s), etc) are added in the process of building the semantic KGs to better capture the semantic similarity between items. The LDSM algorithm [69] is used to weigh the similarity between items. Then, Matrix Factorization (MF), [19] with the added regularization term in Joint MF (JMF) [94], is used for building the explainable model.

### 3.1.2 The Semantic Web’s Effect in Latent Space

The semantic web or, as it is sometimes referred to, linked open data (LOD), is an online platform for data that is linked using semantics. A prime example of LOD is DBpedia, which is the semantic web version of Wikipedia. The semantic web allows us to infer closeness or similarities between entities based on their shared semantic properties even when these entities are not directly comparable (e.g. different types of entities, different format, etc). This great advantage drives us to build semantically-rich latent spaces for both users and items in the matrix factorization process. We

will initially consider only one item semantic attribute (e.g., the starring actor or actress property in the movie domain or the author property in the book domain) to design the first three models. Then we will use multiple semantic attributes or semantic features to design the fourth model. Matrix factorization allows us to predict how likely each actor is to star in a movie or an author to write a book and the probability of each user liking or disliking a certain semantic attribute. Our approach will create semantic knowledge graphs about each item (e.g., a movie or book) and each user before using the users' prior explicit preferences/ratings to predict unseen items' ratings. In addition to providing explainable recommendations, our approach overcomes the cold-start problem because the semantic properties help recommend new items that have not been rated or seen yet by any user.

## 3.2 Asymmetric Semantic Explainable Matrix Factorization (ASEMF)

Asymmetric factorization [96] handles more than one domain in the process of building a heterogeneous matrix factorization model. In this work, the items' semantic data are the first domain, whereas users' known ratings are the second domain. A flow chart of this method is shown in Figure 3.2. There are two sources of knowledge, the users' explicit preferences and items' semantic data retrieved via SPARQL<sup>1</sup> queries.

### 3.2.1 ASEMF Semantic KGs

Figure 3.2 shows a flowchart of our proposed methodology. First, knowledge sources are pre-processed and prepared to be used as inputs to our algorithm. Explicit preferences are the users' known ratings, whereas semantic data consists of the items' additional information, retrieved from DBpedia using a SPARQL endpoint. To calculate a semantic explainability score for each user with respect to each item, we need to transform the semantic data from its categorical format into a quantitative format by, for example, giving identification numbers to semantic features. We used three types of relationships to construct three explainability graphs. The first graph is item-based, namely the relationship between each item (e.g., a movie or book) and each semantic feature (e.g., an actor or author). The second graph is user-based, namely the relationship between each user and each semantic feature. The last graph is computed using the dot product of the two previous graphs

---

<sup>1</sup><https://www.w3.org/TR/rdf-sparql-query/>

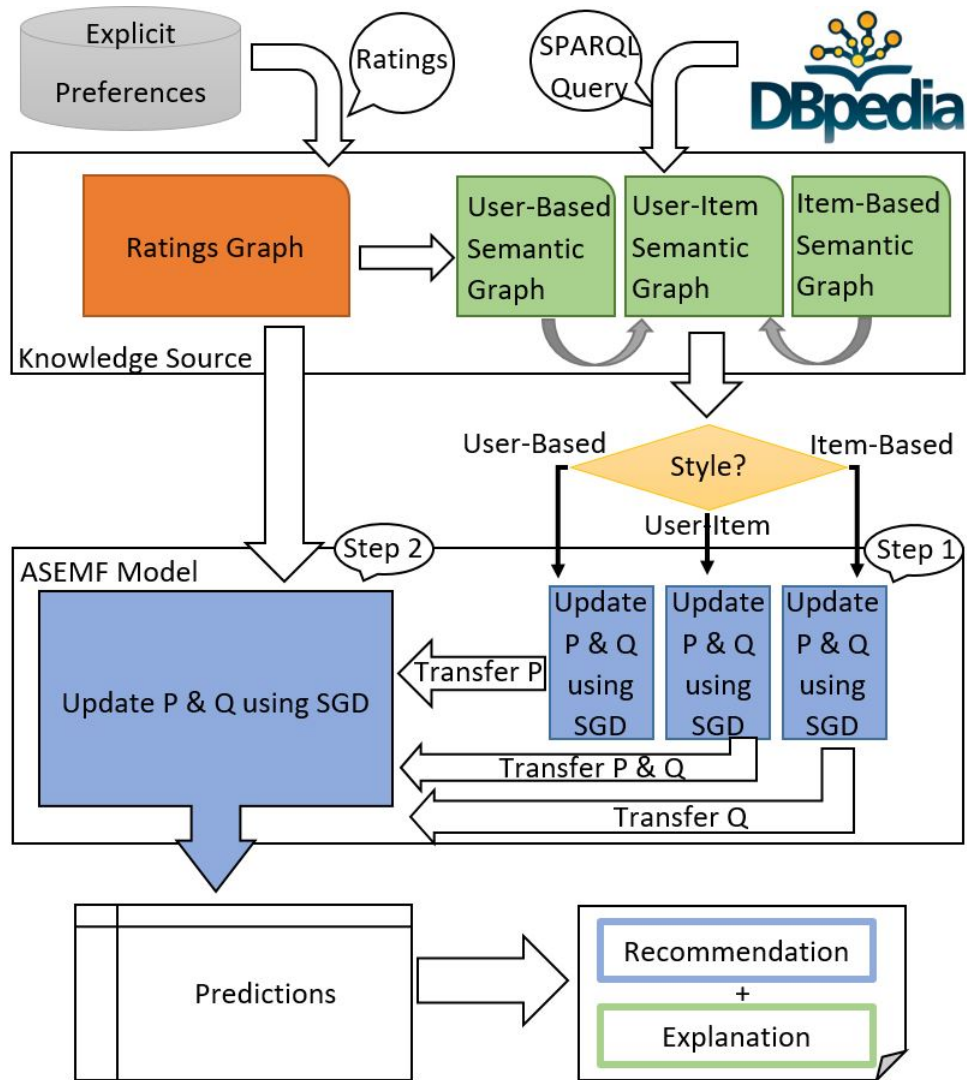


Figure 3.2: Asymmetric Semantic Explainable Matrix Factorization (ASEMF).

and results in a semantic KG for each user and item based on the semantic feature that was chosen at the beginning of the process. As shown in Figure 3.2, three different semantic graphs are thus constructed during the knowledge source stage:

1. An item-based semantic KG ( $S^I$ ).
2. A user-based semantic KG ( $S^U$ ).
3. A user-item-based semantic KG ( $S^{UI}$ ).

The first graph is a binary graph that consists of items (e.g., movies <sup>2</sup> or books <sup>3</sup> and semantic features retrieved via SPARQL endnote <sup>4</sup>). This graph,  $S^I$ , is defined as follows:

$$S_{f,i}^I = \begin{cases} 1 & \text{if } f \text{ possessed by } i \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

where  $f$  represents a semantic feature, and  $i$  denotes a movie.

The user-based graph  $S^U$  is a user-by-semantic-feature graph. This graph is constructed based on the following formula:

$$S_{f,u}^U = \begin{cases} N & \text{if } f \text{ possessed by items rated by } u \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

$f$  is a semantic feature, and  $u$  represents a user.  $N$  is, for example, the number of times each actor  $f$  had starred in movies that user  $u$  has rated in the past.

The final graph  $S^{UI}$ , the user-item graph, is the product of the two previous graphs:

$$S_{u,i}^{UI} = \begin{cases} S_{f,u}^U \cdot S_{f,i}^I & \text{if } S_{f,u}^U \cdot S_{f,i}^I > \theta^s \\ 0 & \text{otherwise} \end{cases} \quad (3.3)$$

<sup>2</sup><https://grouplens.org/datasets/movielens/>

<sup>3</sup><http://www2.informatik.uni-freiburg.de/cziegler/BX/>

<sup>4</sup><http://dbpedia.org/sparql>



### 3.2.2 Model Building

Figure 3.2 shows that after all graphs are constructed, the algorithm handles the rest of the process to compute a model using an asymmetric matrix factorization framework. The model is built in two steps:

1. The first step is the factorization of the semantic graph using a number of latent features  $k$  to learn the semantic latent spaces. The following is the matrix factorization for the semantic graph:

(a) Item-based semantic KG:

$$S_{f,xi}^I \simeq P_{f,xk}^F X Q_{ixk}^T \quad (3.4)$$

(b) User-based semantic KG:

$$S_{uxf}^U \simeq P_{uxk} X Q_{f,xk}^{F,T} \quad (3.5)$$

(c) User-Item-based semantic KG:

$$S_{uxi}^{UI} \simeq P_{uxk} X Q_{ixk}^T \quad (3.6)$$

$S^I$ ,  $S^U$ , and  $S^{UI}$  are the semantic KGs, either item-based, user-based, or user-item based, respectively.  $f$  is the semantic feature (e.g., the starring actor or author), while  $i$  denotes the item (e.g., a movie or book), and  $u$  is the user.  $P_{f,xk}^A$  represents the semantic feature lower-rank dimensional space, where  $Q_{ixk}^I$  denotes the item's lower-rank dimensional space.  $F$  denotes the semantic feature name (e.g. actor), and  $T$  denotes the matrix transpose. The objective functions to be minimized over either of the constructed semantic KGs are as follows:

$$J_1 = \sum_{f,i \in S} (S_{f,i}^I - p_f^F q_i^T)^2 + \frac{\beta}{2} (\|p_f^F\|^2 + \|q_i\|^2) \quad (3.7)$$

$$J_2 = \sum_{u,f \in S} (S_{u,f}^U - p_u q_f^{F,T})^2 + \frac{\beta}{2} (\|p_u\|^2 + \|q_f^F\|^2) \quad (3.8)$$

$$J_3 = \sum_{u,i \in \mathcal{S}} (S_{u,i}^{UI} - p_u q_i^T)^2 + \frac{\beta}{2} (\|p_u\|^2 + \|q_i\|^2) \quad (3.9)$$

The  $L_2$  regularization term  $\frac{\beta}{2} (\|p_f^F\|^2 + \|q_i\|^2)$  in Equation 3.7 is added to prevent overfitting and weighted using coefficient  $\beta$ , which is responsible for the smoothness of the new term. While  $J_1$  is a non-convex function, it is convex with respect to either the  $p$  or the  $q$  vectors individually. Therefore, we can optimize this function by updating  $p$  and  $q$  alternatively using stochastic gradient descent [93], which will allow updating  $p$  and  $q$  iteratively until the objective function  $J_1$  converges. Considering Equation 3.7, the gradient of  $J_1$  with respect to  $p_f^A$  is

$$\frac{\partial J_1}{\partial p_f^F} = -2(S_{f,i}^I - p_f^F q_i^T) q_i + \beta p_f^F \quad (3.10)$$

The gradient of  $J_1$  with respect to  $q_i$  is

$$\frac{\partial J_1}{\partial q_i} = -2(S_{f,i}^I - p_f^F q_i^T) p_f^F + \beta q_i \quad (3.11)$$

The gradients of  $J_2$  and  $J_3$  are similar to  $J_1$ 's gradient. The gradient-based update rules when considering the item-based semantic KG in  $J_1$  are given by

$$p_f^{F,(t+1)} \leftarrow p_f^{F,(t)} + \alpha (2(S_{f,i}^I - p_f^{F,(t)} (q_i^{(t)})^T) q_i^{(t)} - \beta p_f^{F,(t)}) \quad (3.12)$$

and

$$q_i^{(t+1)} \leftarrow q_i^{(t)} + \alpha (2(S_{f,i}^I - p_f^{F,(t)} (q_i^{(t)})^T) p_f^{F,(t)} - \beta q_i^{(t)}), \quad (3.13)$$

where  $\alpha$  is the learning rate.

When considering the user-based semantic KG in  $J_2$ , the update rules are given by

$$p_u^{(t+1)} \leftarrow p_u^{(t)} + \alpha (2(S_{u,f}^U - p_u^{(t)} (q_f^{F,(t)})^T) q_f^{F,(t)} - \beta p_u^{(t)}) \quad (3.14)$$

and

$$q_f^{F,(t+1)} \leftarrow q_f^{F,(t)} + \alpha(2(S_{u,f}^U - p_u^{(t)}(q_f^{F,(t)})^T)p_f^{(t)} - \beta q_u^{F,(t)}), \quad (3.15)$$

When considering the user-item-based semantic KG in  $J_3$ , the update rules are

$$p_u^{(t+1)} \leftarrow p_u^{(t)} + \alpha(2(S_{u,i}^{UI} - p_u^{(t)}(q_i^{(t)})^T)q_i^{(t)} - \beta p_u^{(t)}) \quad (3.16)$$

and

$$q_i^{(t+1)} \leftarrow q_i^{(t)} + \alpha(2(S_{u,i}^{UI} - p_u^{(t)}(q_i^{(t)})^T)p_u^{(t)} - \beta q_i^{(t)}) \quad (3.17)$$

2. During the second step, the algorithm performs factorization depending on which semantic KG is chosen. First, we perform the matrix factorization on known ratings:

$$R_{uxi} \simeq P_{uxk}XQ_{ixk}^T \quad (3.18)$$

$R_{uxi}$  is the matrix of known ratings, while  $P_u$  represents the user's lower-rank latent space, and  $Q_i$  is the item's lower-rank latent space. The idea here is that, instead of using a randomized  $p$  and  $q$  for the factorization process, we will initialize them using semantic data during the first step, which will give meaningful values for  $p$  and  $q$ . Then, we transfer them to the second step and update them again using the known ratings. The objective function to be minimized over the given ratings is:

$$J_4 = \sum_{u,i \in R} (R_{u,i} - p_u q_i^T)^2 + \frac{\beta}{2} (\|p_u\|^2 + \|q_i\|^2) \quad (3.19)$$

During the updating stage, depending on which KG is used in the first step, there will be three cases:

- (a) Item-based semantic KG (in  $J_1$ ): From the first step, the user's latent space  $p$  will be discarded and randomized, while the item's latent space  $q$  will be transferred to the second step and updated using the known ratings as follows:

$$p_u^{(t+1)} \leftarrow p_u^{(t)} + \alpha(2(R_{u,i} - p_u^{(t)}(q_i^{(t)})^T)q_i^{(t)} - \beta p_u^{(t)}) \quad (3.20)$$

$$q_i^{(t+1)} \leftarrow q_i^{(t)} + \alpha(2(R_{u,i} - p_u^{(t)}(q_i^{(t)})^T)p_u^{(t)} - \beta q_i^{(t)}) \quad (3.21)$$

- (b) User-based semantic KG (in  $J_2$ ): From the first step, the user latent space  $p$  will be transferred to the second step and updated using the known ratings while the Item latent space  $q$  will be discarded and randomized. The update rules are the same as in 3.20 and 3.21.
- (c) User-item semantic KG (in  $J_3$ ): From the first step, both the user's latent space  $p$  and the item's latent space  $q$  are transferred to the second step and updated using known ratings. Again, the update rules are the same as in 3.20 and 3.21.

### 3.2.3 Explanation Style

When the model recommends a movie, it attaches an explanation. As [101] stated, three explanation styles are used in recommender systems: NSE, ISE, and KSE. In this method, we use ISE style explanation to show the influence of users' semantic property preferences on making recommendations. Since [98] found that a histogram-like explanation interface was the most preferred by users, we adopt this format for our explanation. Another possible style is to show a plain text interface on which the most influential semantic property in the recommendation is mentioned. Figures 3.3 and 3.4 show examples of semantic ISE explanations for two sample users.



Figure 3.3: A histogram-like semantic ISE (left) and a plain text semantic ISE (right) recommendation for sample user 1.

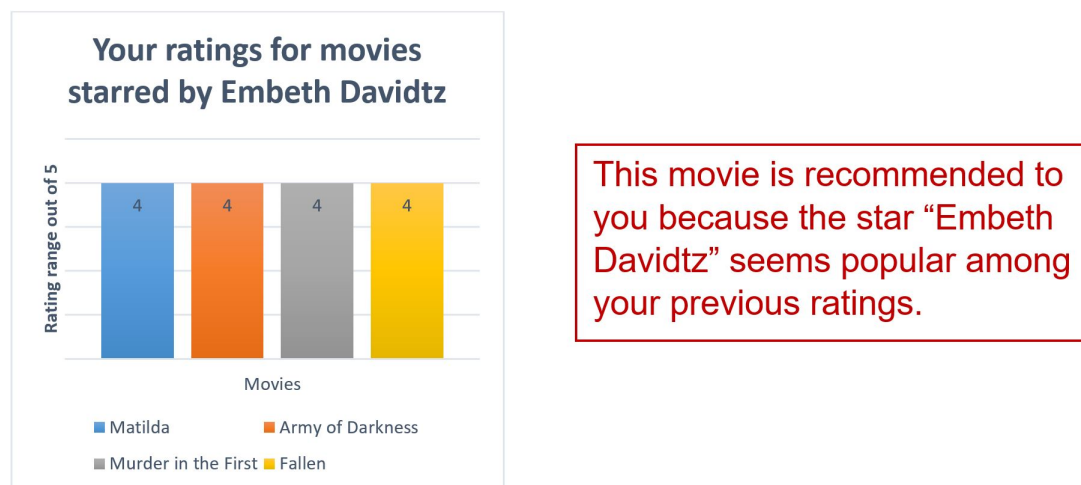


Figure 3.4: A histogram-like semantic ISE (left) and a plain text semantic ISE (right) recommendation for sample user 2.

### 3.3 Semantic Explainable Matrix Factorization (SemEMF)

In Section 2.2.5.3, we reviewed explainable matrix factorization, which was proposed by [13] [14] [15] to produce explanations for recommendations using the neighborhood technique. Items that are liked by neighbors are more likely to be liked by the target user. Hence, the explanation, in the latent space, is in the form of a matrix of explainability scores for each user and each item. This matrix contributes to the process of building the latent space for both vectors, users, and items.

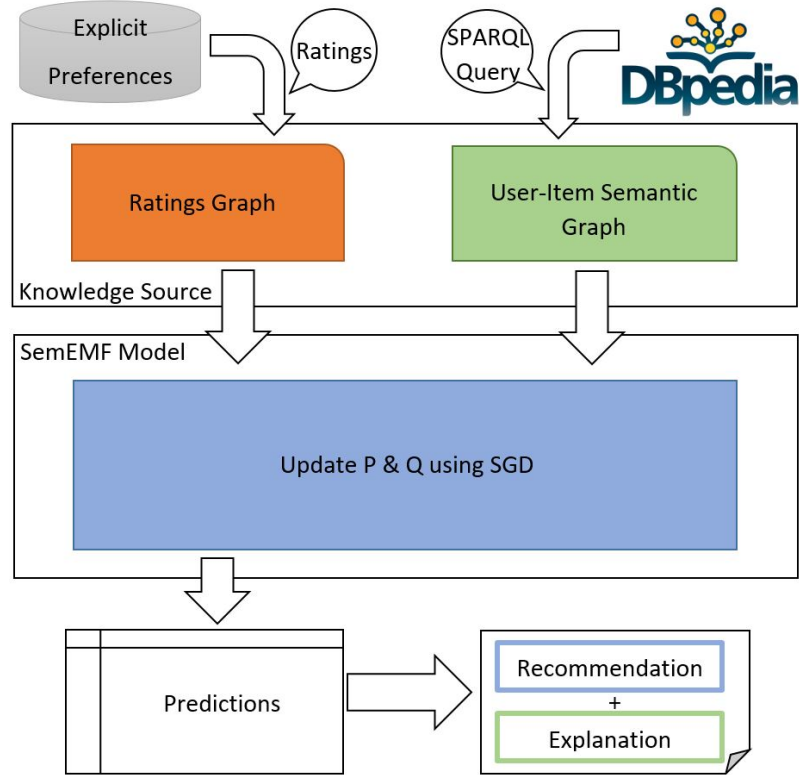


Figure 3.5: Semantic Explainable Matrix Factorization.

The data used in constructing the explainability score are merely the known ratings. For this reason, we propose a new method that calculates the explainability score using semantic data, which will allow a greater variety in the explanations. Figure 3.5 shows a flow chart of our proposed method.

### 3.3.1 SemEMF Semantic KGs

Figure 3.5 shows that preparing the knowledge source is the first step in building the model. The user-item-based semantic KG ( $S^{UI}$ ) that we discussed in sec. 3.2.1 is used to compute the explainability scores based on the semantics to train our model. This process is discussed in detail in the next section.

### 3.3.2 Model Building

Unlike [13] [14] [15] which used the neighborhood technique in producing the explainability scores, we use semantic data. Therefore, after computing the ratings and semantic explainability

graphs, we introduce the following objective function to be minimized:

$$\min J = \sum_{u,i \in R} (R_{u,i} - p_u q_i^T)^2 + \frac{\beta}{2} (\|p_u\|^2 + \|q_i\|^2) + \frac{\gamma}{2} \|p_u - q_i\|^2 S_{u,i}^{UI} \quad (3.22)$$

The first two terms of the formula were used in [19], and include a regularization term for each unknown parameter (latent vector) to avoid over-fitting, with  $\beta$  being a regularization coefficient that controls the smoothness of this newly added term. The third term incorporates the new term that handles the semantic explainability graph.  $\gamma$  is a coefficient that ensures the smoothness of the newly added term, while  $S_{u,i}^{UI}$  is the semantic explainability score of user  $u$  for item  $i$ . The construction of  $S^{UI}$  is detailed in sec. 3.2.1

We can utilize stochastic gradient descent to update  $p$  and  $q$  iteratively until the convergence of  $J$ .

The gradient of  $J$  with respect to  $p_u$  is calculated as follows:

$$\frac{\partial J}{\partial p_u} = -2(R_{u,i} - p_u q_i^T)q_i + \beta p_u - \gamma(p_u - q_i)S_{u,i}^{UI} \quad (3.23)$$

The gradient of  $J$  with respect to  $q_i$  is

$$\frac{\partial J}{\partial q_i} = -2(R_{u,i} - p_u q_i^T)p_u + \beta q_i + \gamma(p_u - q_i)S_{u,i}^{UI} \quad (3.24)$$

Using the gradients, the formulation of the updating rules will be

$$p_u^{(t+1)} \leftarrow p_u^{(t)} + \alpha(2(R_{u,i} - p_u^{(t)}(q_i^{(t)})^T)q_i^{(t)} - \beta p_u^{(t)} - \gamma(p_u^{(t)} - q_i^{(t)})S_{u,i}^{UI}) \quad (3.25)$$

$$q_i^{(t+1)} \leftarrow q_i^{(t)} + \alpha(2(R_{u,i} - p_u^{(t)}(q_i^{(t)})^T)p_u^{(t)} - \beta q_i^{(t)} + \gamma(p_u^{(t)} - q_i^{(t)})S_{u,i}^{UI}) \quad (3.26)$$

The main difference between our method and the basic matrix factorization method [19] is that our method builds a model that can generate recommendations and explanations simultaneously. Our model also differs from [13] [14] [15] because we exploit the semantic data in the process of generating explanations. Furthermore, this method differs from the ASEMFM method, which was presented in sec. 3.2, by virtue of requiring only one step to build the model.

### 3.3.3 Explanation Style

The explanation style in this method is similar to the first explanation style in the previous method, which was semantic ISE (see sec. 3.2.3). Examples of plain text and histogram-like explanations can be seen in Figures 3.3 and 3.4.

## 3.4 Merged Semantic Explainable Matrix Factorization (MergedSemEMF)

In this section, we propose a new method that incorporates EMF and SemEMF methods in one approach. The intuition here is that since EMF and SemEMF are both designed to generate explanations by adding a new regularization term to the MF, combining both techniques may result in better performance. Once the model is built, and to generate explanations, there will be three available options. One is to allow the target user to choose between the two explainability graphs as the explanation generation base. The second option is to display both styles, and the third option is to automatically choose the better style based on the explainability score. A flow chart of this method is shown in Figure 3.6.

As seen in Figure 3.6, the first step is to have the input data, which are the known ratings, neighborhood explainability score graph, and semantic explainability score graph. Then,  $p$  and  $q$  will be iteratively updated using stochastic gradient descent. Finally, the recommendation and explanation will be provided.

### 3.4.1 MergedSemEMF Semantic KGs

The semantic KG follows the user-item-based style, which we explained in Section 3.2.1.

### 3.4.2 Model Building

In the model building stage, since we are including both explainability score graphs in the process of building the user and item latent space vectors, the following objective function is formulated:



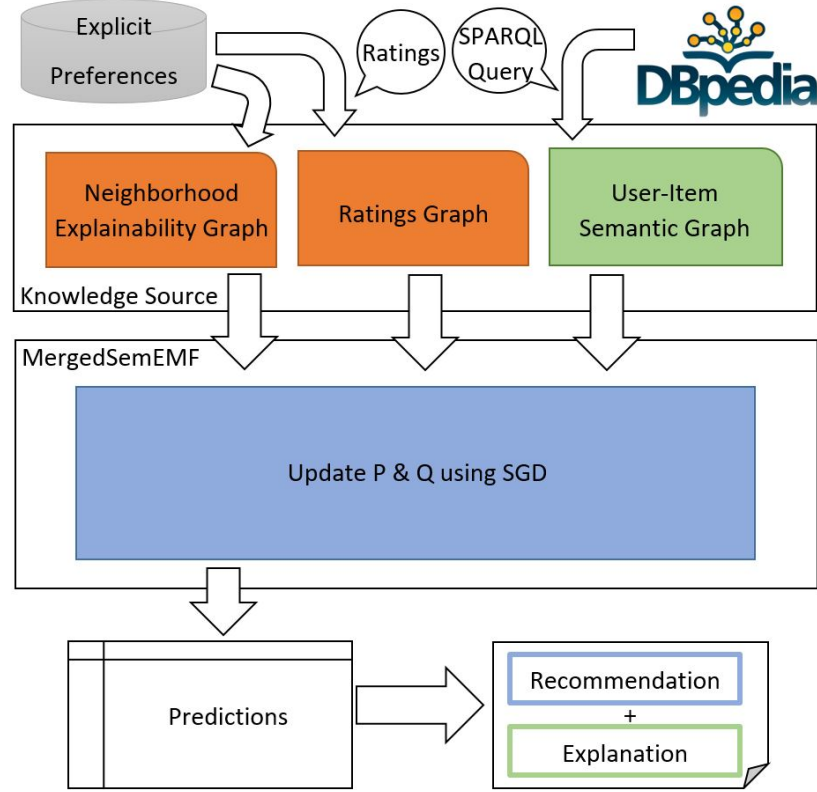


Figure 3.6: Merged Semantic Explainable Matrix Factorization.

$$J = \sum_{u,i \in R} (R_{u,i} - p_u q_i^T)^2 + \frac{\beta}{2} (\|p_u\|^2 + \|q_i\|^2) + \frac{\lambda}{2} \|p_u - q_i\|^2 W_{u,i} + \frac{\gamma}{2} \|p_u - q_i\|^2 S_{u,i}^{UI} \quad (3.27)$$

The first three terms are the explainable matrix factorization (EMF) objective [13] [14] [15], which was described in Section 2.2.5.3. The last term is where the semantic explainability scores graph,  $S^{UI}$ , is handled.  $S^{UI}$  is the semantic explainability score graph, which was detailed in Section 3.2.1.  $\gamma$  is a coefficient used to control the smoothness of the new term.

We can apply stochastic gradient descent to update  $p$  and  $q$  iteratively until the convergence.

The derivative of  $J$  with respect to  $p_u$  is

$$\frac{\partial J}{\partial p_u} = -2(R_{u,i} - p_u q_i^T)q_i + \beta p_u - \lambda(p_u - q_i)W_{u,i} - \gamma(p_u - q_i)S_{u,i}^{UI} \quad (3.28)$$

The derivative of  $J$  with respect to  $q_i$  is

$$\frac{\partial J}{\partial q_i} = -2(R_{u,i} - p_u q_i^T) p_u + \beta q_i + \lambda(p_u - q_i) W_{u,i} + \gamma(p_u - q_i) S_{u,i}^{UI} \quad (3.29)$$

Using the gradient, the formulation of the updating rules is given by

$$p_u^{(t+1)} \leftarrow p_u^{(t)} + \alpha(2(R_{u,i} - p_u^{(t)}(q_i^{(t)})^T) p_u^{(t)} - \beta p_u^{(t)} - \lambda(p_u^{(t)} - q_i^{(t)}) W_{u,i} - \gamma(p_u^{(t)} - q_i^{(t)}) S_{u,i}^{UI}) \quad (3.30)$$

$$q_i^{(t+1)} \leftarrow q_i^{(t)} + \alpha(2(R_{u,i} - p_u^{(t)}(q_i^{(t)})^T) p_u^{(t)} - \beta q_i^{(t)} + \lambda(p_u^{(t)} - q_i^{(t)}) W_{u,i} + \gamma(p_u^{(t)} - q_i^{(t)}) S_{u,i}^{UI}) \quad (3.31)$$

### 3.4.3 Explanation Style

As discussed previously at the beginning of this section, there will be three options for showing explanations. The first option is to allow the target user to choose which explainability graph to rely on to generate the explanation. The second option is to allow the algorithm to decide, and the third option is to display both explanations. In this work, we use the simple option and show the semantic ISE as done in sections 3.2 and 3.3. An example of a neighborhood-based style explanation can be seen in Figure 2.15. For an example of a semantic explanation, please refer to Figures 3.3 and 3.4.

## 3.5 Linked Data Semantic Distance Matrix Factorization (LSDMF)

In the previous three methods, we relied on one property to construct the semantic explainability graph. However, in this section, we utilize LSD [69] to build a multi-property semantic explainable graph to be used within our explainability regularization term. Stochastic gradient descent is used, as in the previous methods, to update the user and item latent space vectors. Figure 3.7 shows a flowchart of this new method.

The flowchart starts with data preprocessing, where two input items, the known ratings and the new constructed semantic explainability graph, are prepared. Then, the model is learned based on these graphs using SGD. Predictions are then computed, and hence, recommendations with explanations are presented to the target user. In the following sections, the semantic graph and model building are discussed in greater detail.

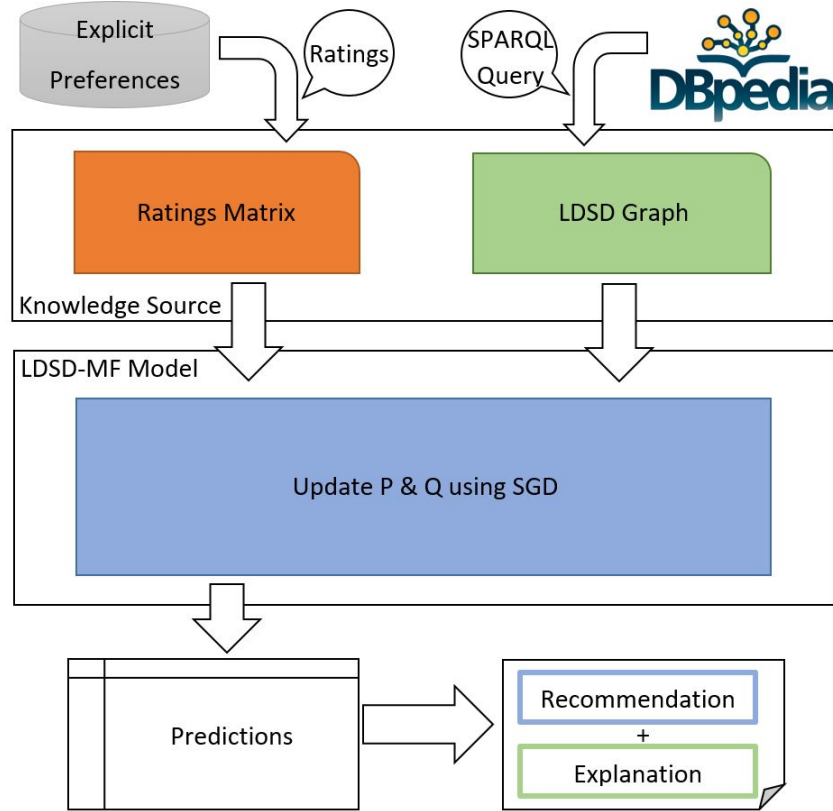


Figure 3.7: LSDDMF flowchart.

### 3.5.1 Linked Data Semantic Distance (LSD)

Passant [2] proposed a method to build a music recommender system using Semantic Web resources. The proposed algorithm captures both in-going and out-going as well as direct and indirect links between entities. Figure 3.8 shows a generic example of a semantic KG containing entities and links.  $r_i$  in Figure 3.8, represents a resource (e.g. movie, actor, etc).  $l_j$  is a link or property (e.g. starring, directedBy, etc). Out of  $r_i$  and  $l_j$  we can extract six RDF triples that exist in this graph and they are:

$$E = \{(l1 : r1 : r2), (l1 : r2 : r1), (l2 : r1 : r2), (l2 : r1 : r3), (l3 : r1 : r4), (l3 : r2 : r4)\}.$$

As mention earlier, there are in-going and out-going, as well as direct and indirect relationships between resources, which in total, represent the Linked Open Data (LOD). [2] proposed three semantic similarities in such data:

1. Direct similarity:

If there exists a property ( $l_x$ ) that directly links two resources ( $r_y$  and  $r_z$ ), then the value  $C_{l_x, r_y, r_z}^{(d)}$

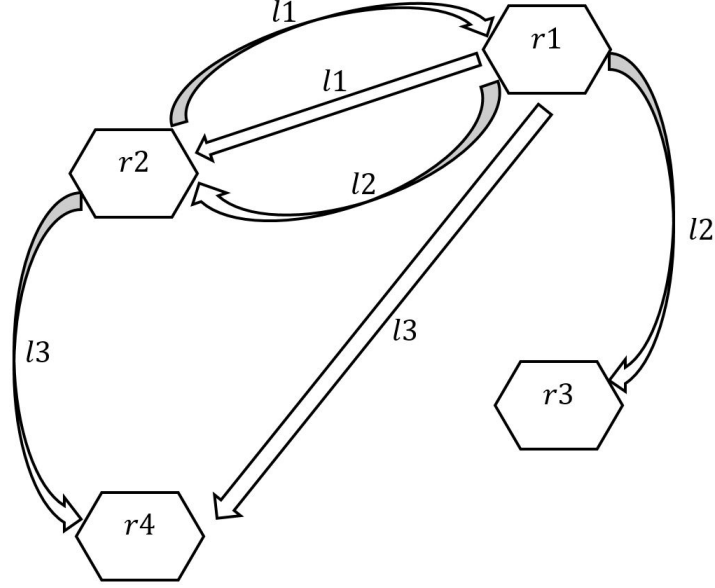


Figure 3.8: A Generic Semantic Knowledge Graph.

is 1, otherwise 0:

$$C_{l_x, r_y, r_z}^{(d)} = \begin{cases} 1 & \text{if exists } (l_x : r_y : r_z) \\ 0 & \text{otherwise} \end{cases} \quad (3.32)$$

$C^{(d)}$  denotes a triple of semantic data, where superscript  $(d)$  means direct. Looking back to Figure 3.8, there exist six direct relationships between the four resources. Therefore, using equation 3.32, we have the following  $C^{(d)}$  values:

$$C_{l1:r1:r2}^{(d)} = 1$$

$$C_{l1:r2:r1}^{(d)} = 1$$

$$C_{l2:r1:r2}^{(d)} = 1$$

Similarly, we can aggregate similarities over many properties as in equation 3.33,

$$C_{n, r_y, r_z}^{(d)} = \begin{cases} \sum_{l_x} C_{l_x, r_y, r_z} & \text{if exists } (l_x : r_y : r_z) \\ 0 & \text{otherwise} \end{cases} \quad (3.33)$$

$$C_{n:r1:r2}^{(d)} = 2$$

also we can aggregate similarities over many target resources as in equation 3.34

$$C_{l_x, r_y, n}^{(d)} = \begin{cases} \sum_{r_z} C_{l_x, r_y, r_z} & \text{if exists } (l_x : r_y : r_z) \\ 0 & \text{otherwise} \end{cases} \quad (3.34)$$

$$C_{l_2: r_1: n}^{(d)} = 2$$

The first similarity function is obtained:

$$LDS D^{(d)}(r_y, r_z) = \frac{1}{1 + C^{(d)}(n, r_y, r_z) + C^{(d)}(n, r_z, r_y)} \quad (3.35)$$

A weighted version of this function is introduced using weighting methodology in [147]:

$$LDS D^{(wd)}(r_y, r_z) = \frac{1}{1 + \sum_x \frac{C^{(d)}(l_x, r_y, r_z)}{1 + \log(C^{(d)}(l_x, r_y, n))} + \sum_x \frac{C^{(d)}(l_x, r_z, r_y)}{1 + \log(C^{(d)}(l_x, r_z, n))}} \quad (3.36)$$

## 2. Indirect in and out similarity:

Another LDS D algorithm is designed to handle the indirect, in and out, RDF triples. Looking at the following formula:

$$C_{l_x, r_y, r_z}^{(ii)} = \begin{cases} 1 & \text{if exists } n \text{ in } (l_x : n : r_y) \text{ and } (l_x : n : r_z) \\ 0 & \text{otherwise} \end{cases} \quad (3.37)$$

$$C_{l_x, r_y, r_z}^{(io)} = \begin{cases} 1 & \text{if exists } n \text{ in } (l_x : r_y : n) \text{ and } (l_x : r_z : n) \\ 0 & \text{otherwise} \end{cases} \quad (3.38)$$

We can compute the following indirect in and out similarities values respectively:

$$C_{l_2, r_2, r_3}^{(ii)} = 1 \quad (3.39)$$

$$C_{l_3, r_1, r_2}^{(io)} = 1 \quad (3.40)$$

The idea basically is that if there exists a resource that is in both triples with the same property, then the value is 1, otherwise 0. Using this assumption, we can infer the following relationships from Figure 3.8: (note that Superscript <sup>(ii)</sup> indicates indirect in and <sup>(io)</sup> indicates indirect out).

$C_{l_3, r_1, r_2}^{(io)} = 1$  means that both  $r_1$  and  $r_2$  are indirectly linked by outgoing link  $l_3$  from both resources to  $r_4$ . For indirect incoming relationship, here is an example from the same Figure:  $C_{l_2, r_2, r_3}^{(ii)} = 1$  where we can see that the link  $l_2$  is ingoing into both resources  $r_2$  and  $r_3$  from one resource  $r_1$ .

Finally the equation for the LDSD similarity is given by combining both ingoing and outgoing similarities:

$$LDSD^{(i)}(r_y, r_z) = \frac{1}{1 + C^{(ii)}(n, r_y, r_z) + C^{(io)}(n, r_y, r_z)} \quad (3.41)$$

A weighted version is given by:

$$LDSD^{(wi)}(r_y, r_z) = \frac{1}{1 + \sum_x \frac{C^{(ii)}(l_x, r_y, r_z)}{1 + \log(C^{(ii)}(l_x, r_y, n))} + \sum_x \frac{C^{(io)}(l_x, r_y, r_z)}{1 + \log(C^{(io)}(l_x, n, r_z))}} \quad (3.42)$$

### 3. Combined similarity:

Lastly, a final combined and weighted version of LDSD is formulated as follows:

$$LDSD^{(wc)}(r_y, r_z) = \frac{1}{1 + \sum_x \frac{C^{(d)}(l_x, r_y, r_z)}{1 + \log(C^{(d)}(l_x, r_y, n))} + \sum_x \frac{C^{(d)}(l_x, r_z, r_y)}{1 + \log(C^{(d)}(l_x, r_z, n))} + \sum_x \frac{C^{(ii)}(l_x, r_y, r_z)}{1 + \log(C^{(ii)}(l_x, r_y, n))} + \sum_x \frac{C^{(io)}(l_x, r_y, r_z)}{1 + \log(C^{(io)}(l_x, n, r_z))}} \quad (3.43)$$

It combines both weighted, direct and indirect, LDSD equations mentioned earlier.

To sum up, the similarity measures allow us to construct an item by item semantic similarity graph using semantic data in order to use it as an explanation regularization term in our proposed method. In our research, since we work on movie and book item domains, we focused on the indirect, ingoing and out-going, relationships. The reason is that there are almost no direct links between items.

However, actors, as an example in the movie domain, can indirectly, both in and out, link different items to each other. Equation 3.43 allows us to construct a semantic KG that captures direct and indirect semantic relationships between items.

### 3.5.2 Model Building

The proposed loss function is inspired by the work of [19] and [94]. It is defined as follows:

$$J = \sum_{u,i \in R} (R_{u,i} - p_u q_i^T)^2 + \frac{\gamma}{2} \sum_{i,j \in S^{l_{dsd}}} (S_{i,j}^{l_{dsd}} - q_i q_j^T)^2 + \frac{\beta}{2} (\|p_u\|^2 + \|q_i\|^2) \quad (3.44)$$

$R_{u,i}$  represents the rating for item  $i$  by user  $u$ .  $p_u$  and  $q_i$  represent the low dimensional latent factor vectors of users and items, respectively.  $S^{l_{dsd}}$  is the semantic KG constructed using equation 3.42.  $q_i$  and  $q_j$  indicate two items in the KG,  $S^{l_{dsd}}$ , and  $\gamma$  is a coefficient that weighs the contribution of the new term,  $S^{l_{dsd}}$ . Stochastic gradient descent [93] is employed to update  $p$  and  $q$  iteratively until  $J$  converges.

The gradient of  $J$  with respect to  $p_u$  is given by:

$$\frac{\partial J}{\partial p_u} = -2(R_{u,i} - p_u q_i^T)q_i + \beta p_u. \quad (3.45)$$

The gradient of  $J$  with respect to  $q_i$  is given by

$$\frac{\partial J}{\partial q_i} = -2(R_{u,i} - p_u q_i^T)p_u + 2\gamma(S_{i,j} - q_i q_j^T)q_j + \beta q_i. \quad (3.46)$$

The updating rules are given by:

$$p_u^{(t+1)} \leftarrow p_u^{(t)} + \alpha(2(R_{u,i} - p_u^{(t)}(q_i^{(t)})^T)q_i^{(t)} - \beta p_u^{(t)}) \quad (3.47)$$

$$q_i^{(t+1)} \leftarrow q_i^{(t)} + \alpha(2(R_{u,i} - p_u^{(t)}(q_i^{(t)})^T)p_u^{(t)} + 2\gamma(S_{i,j}^{l_{dsd}} - q_i^{(t)}(q_j^{(t)})^T)q_j^{(t)} - \beta q_i^{(t)}). \quad (3.48)$$

### 3.5.3 Explanation Style

The semantic explanation KGs are constructed using the approach described in section 3.2.1 for all semantic attributes, and hence explanations. In addition to the known ratings used to update  $q_i$ , the semantic explanation KGs also contribute to the final predicted rating of item  $i$  by user  $u$ .

## 3.6 Inferred Fact Style Explanation (IFSE)

In this work, a new explanation style we propose that utilizes the previously constructed KGs on users and semantic attributes. In this style, the uncertainty degree of the users' preferences for semantic attributes is employed to justify a recommendation. Inference rules are used to derive new knowledge from known facts [137]. For example, if  $A$  is of type  $B$  while  $B$  is of type  $C$ , then  $A$  must be of type  $C$ . This example indicates complete certainty; however, in our work, we obtained the uncertainty degree from the constructed user by semantic attribute matrix based on the work of [128] which was presented in section 3.2. For example, if a user,  $u$  watched, interacted with, or rated a certain item,  $i$ , and this item is linked to a certain semantic attribute,  $a$ , a new inferred fact is derived: user  $u$  likes semantic attribute  $a$  to a certain degree (see Equations 3.1 and 3.2).

The likability degree depends on the number of times the user interacts with items that are linked to that specific semantic attribute (see Equation 3.2). The item's rating itself does not necessarily reflect the user opinion about the underlying semantic attributes, he or she may dislike the item but may still like some of the semantic attributes. For example, in the movie domain, if the user rates a movie with 1 out of 5, this may happen because of bad directing, poor picture quality, etc, not necessarily because of bad acting or a bad story. Thus, once the user rates a movie, this tells us that he or she is showing an interest somehow towards the semantic attributes. Only through accumulating the likes and dislikes over many items, we can infer a likeability degree towards a particular semantic attribute. Figure 3.9 illustrates an example of an inferred fact in the movie domain. In Chapter 4, we will show an example of this explanation style (see Figure 4.11) when we describe the user study that we conducted in this study. Algorithm 1 below shows the steps of this technique with referral to the corresponding equations.



---

**Algorithm 1** IFSE-based Explanation Generation Algorithm

---

**Input:** Active user ( $u$ ), semantic Knowledge Graphs constructed using equations 3.1 and 3.2, knowledge graphs originally retrieved from DBpedia using SPARQL

**Output:** IFSE-based histogram explanation

1. Use one of the proposed models in sections 3.2, 3.3, 3.4, or 3.5) to recommend the top  $n$  items ( $i$ ).
  2. Find the similarities between user ( $u$ ) and each selected semantic attribute ( $f$ ) that exists in the recommended item ( $i$ ) based on the semantic KGs defined in the input section.
  3. Generate the IFSE histogram using the similarities between the user ( $u$ ) and each semantic attribute ( $f$ ) found in step 2.
- 

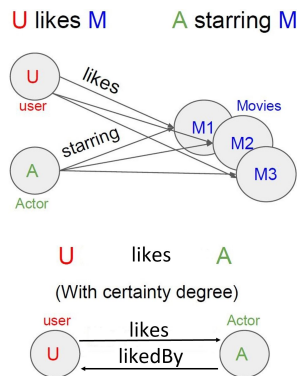


Figure 3.9: Inferred Facts: Movie Example.

## CHAPTER 4

# EXPERIMENTAL EVALUATION

In this chapter, we report the experimental results of our proposed methods. The evaluation process focuses on evaluating our proposed methods against baseline approaches in terms of prediction error rates, top-n recommendations, and the explainability of the recommended items using appropriate measurements. The experimental setting and the three evaluation mechanisms, as well as the results of a user study, are also presented.

### 4.1 Experimental Setting

Two different domains are used in this study to test the proposed methods: movies using the 100K MovieLens benchmark dataset <sup>1</sup>, which includes 943 users, 1,682 movies, and 100K ratings, and books using the book-crossing dataset <sup>2</sup>, which includes 278,858 users, 271,379 books, and 1M ratings. We mapped the movies and books to the DBpedia KG, which resulted in a reduction in the total number of movies from 1,682 to 1,012 and books from 271,379 to 2,217. Furthermore, in the book domain, only users with at least 6 ratings and books that have been rated at least 6 times were kept to reduce the sparsity and computational complexity. This is due to either the absence of some movies or books in DBpedia or different spellings of some movie or book titles in the datasets. Movie and book titles are used in the mapping process due to differences in IDs in all datasets. The total number of ratings were reduced to 60K in the movie domain and 25K in the book domain after mapping. Ratings in both datasets were normalized to the highest rating in the dataset. See Figures 4.1 and 4.2 for more details.

---

<sup>1</sup><https://grouplens.org/datasets/movielens/>

<sup>2</sup><http://www2.informatik.uni-freiburg.de/cziegler/BX/>

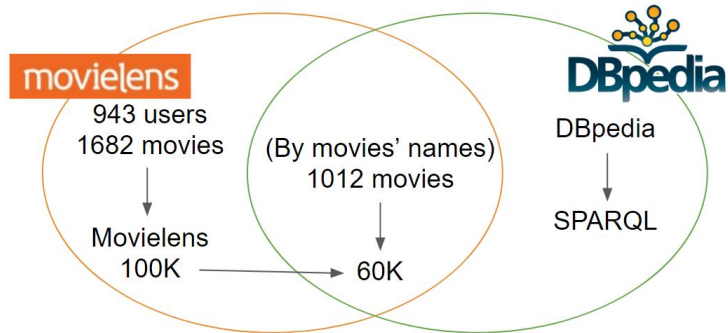


Figure 4.1: The mapping process for the movie domain.

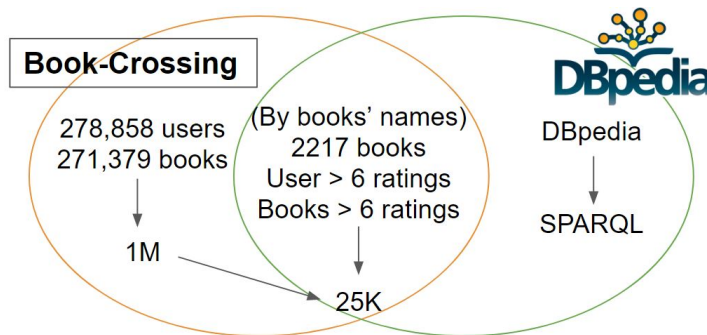


Figure 4.2: The mapping process for the book domain.

The semantic KGs' sizes are shown in Tables 4.1 and 4.2. The first row shows the size of the graphs of the already existed information about both movies and books before applying the inferred fact mechanism, meaning that the relationship is direct between the items and the semantic attributes. The second row shows the semantic KGs sizes after applying the IF mechanism, meaning that new relationships now have been inferred between the user and the semantic attributes. Here we used the new information to feed the semantic KGs, and hence build the models.

Only five semantic attributes were used in the movie domain and four in the book domain that we believe are the most important and influential for the users. In other specialized domains, such as medicine and education, a domain expert would be required in the process of choosing the proper semantic properties. Other semantic attributes, such as music composer and budget in the movie domain, and the type in the book domain, were judged to be less important factors from the user prospective when justifying recommended movie to watch or book to read.

The data were split into two sets, and 90% of each user's ratings were designated to be the training set, while the remaining 10% was designated as the testing set. Parameters  $\alpha$ ,  $\beta$ ,  $\lambda$ , and  $\gamma$

Table 4.1: The effect of applying the inferred fact mechanism on the semantic KGs’ sizes in the movie domain; the first and second rows show the sizes before and after the application, respectively.

Preferences	Ratings	Subjects	Actors	Directors	Producers	Writers
Movies (Before IF)	60K	19983	6770	1577	1868	1944
Movies (After IF)	60K	818784	332484	92008	103943	110692

Table 4.2: The effect of applying the inferred fact mechanism to the semantic KGs’ sizes in the book domain; the first and second rows show the sizes before and after the application, respectively.

Preferences	Ratings	Subjects	Authors	Publisher	Literary Genre
Books (Before IF)	25K	3375	968	408	274
Books (After IF)	25K	2007447	213535	629067	888011

were tuned to their best values using cross-validation. The optimal values are  $\alpha = 0.01$ ,  $\beta = 0.1$ ,  $\lambda = 0.005$ , and  $\gamma = 0.1$ . Recall that  $\alpha$  is the learning rate,  $\beta$  is the coefficient for the regularization term,  $\lambda$  and  $\gamma$  are the controlling coefficient used to weight the contribution of the new terms, the explainability graphs. Stochastic Gradient Descent was used to find the optimal solution.

We compared our methods to the following baseline methods: basic matrix factorization (MF) [19], explainable matrix factorization (EMF) [13] [14] [15], basic asymmetric matrix factorization (AMF) [96], and probabilistic matrix factorization (PMF) [126].

The hypothesis is that the mean of all metrics for all models are equal. We are trying to prove that this is false by conducting a t-test experiments for all methods using all metrics. The models were each executed 10 times while randomly initializing the user and item latent factors. Then we calculated all metrics and performed the significance tests which are reported in this chapter.

## 4.2 Accuracy Evaluation

In order to evaluate our proposed methods in terms of accuracy, we used the root mean square error equation as follows:

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} (r'_{ui} - r_{ui})^2}. \quad (4.1)$$

$T$  represents the total number of predictions, while  $r'_{ui}$  is the predicted rating for item  $i$  by user  $u$ , and  $r_{ui}$  is the actual rating of item  $i$  by user  $u$ . Tables 4.3 and 4.4 show the error rate against the number of latent features  $K$  in movie and book domains.

Table 4.3: RMSE versus the number of latent factors  $K$  in the movie domain. SemEMF, MergedSemEMF, ASEMF\_UIB, and LDSDMF denote our proposed methods.

RMSE								
K	MF	PMF	AMF	EMF	SemEMF	MergedSemEMF	ASEM <u>F</u> <sub>UIB</sub>	LDSDMF
10	0.205	0.698	0.236	0.205	0.207	0.207	0.205	<b>0.204</b>
20	0.212	0.698	0.27	0.211	0.213	0.213	0.204	0.204
30	0.214	0.698	0.309	0.215	0.216	0.216	0.204	0.204
40	0.216	0.7	0.344	0.217	0.217	0.218	0.203	0.205
50	0.217	0.7	0.374	0.217	0.218	0.219	0.203	0.206

Table 4.4: RMSE versus the number of latent factors  $K$  in the book domain. SemEMF, MergedSemEMF, ASEMF\_UIB, and LDSDMF denote our proposed methods.

RMSE								
K	MF	PMF	AMF	EMF	SemEMF	MergedSemEMF	ASEM <u>F</u> <sub>UIB</sub>	LDSDMF
10	0.345	0.803	0.187	0.346	0.338	0.342	0.333	0.410
20	0.598	0.804	0.239	0.603	0.574	0.576	0.333	0.576
30	0.833	0.805	0.315	0.841	0.771	0.780	0.334	0.734
40	1.062	0.805	0.405	1.054	0.952	0.951	0.333	0.883
50	1.273	0.806	0.496	1.264	1.102	1.099	<b>0.332</b>	1.033

Table 4.3 shows a comparison of the proposed models and the baseline approaches in the movie domain based on the prediction error rate for 10 runs for each method against each number of features  $K$ . We conducted a significance test on LDSDMF in comparison to all the baseline approaches at  $K = 10$ . We ran each method, as mentioned previously, 10 times while randomly initializing the user latent factors  $p$  and item latent factors  $q$  each time. The results in Table 4.5 show that LDSDMF’s p-value were significantly lower than all the baseline models indicating that there is a significant difference between the compared models, and our proposed model significantly outperforms the baseline approaches. Overall, ASEMF\_UIB and LDSDMF produced lower error rates as  $K$  increases. However, SemEMF and MergedSemEMF were competing with MF and EMF with an increasing error rate as  $K$  increases. The AMF and PMF models produced higher error rates than all other approaches.

Table 4.4 shows the error rate of all approaches in the book domain. While  $K$  is small, the baseline AMF is the winner. However, by including more hidden features, our proposed model ASEMF\_UIB tended to give a lower error rate than all the other approaches. A significance test was conducted, and at  $K = 50$  between ASEMF\_UIB against all baseline methods, the p-value, as

shown in Table 4.6, is small, meaning that our model significantly outperformed all baseline models. LDSDMF, SemEMF, and MergedSemEMF, alongside the baseline MF and EMF, were competing at an increasing rate as  $K$  increased. PMF was stable at a high error rate.

Table 4.5: RMSE significance test results in the movie domain ( $K = 10$ ).

Model 1	Model 2	p-value
MF	<b>LDSDMF</b>	2.3e-07
EMF	<b>LDSDMF</b>	4.8e-08
PMF	<b>LDSDMF</b>	4.04e-54
AMF	<b>LDSDMF</b>	6.6e-22
ASEMF_UIB	<b>LDSDMF</b>	1.3e-07

Table 4.6: RMSE significance test results in the book domain ( $K = 50$ ).

Model 1	Model 2	p-value
MF	<b>ASEMF_UIB</b>	8.02e-32
EMF	<b>ASEMF_UIB</b>	1.4e-30
PMF	<b>ASEMF_UIB</b>	5.2e-43
AMF	<b>ASEMF_UIB</b>	6.8e-23
LDSDMF	<b>ASEMF_UIB</b>	1.8e-28

### 4.3 Recommendation Evaluation

We will now explore how our proposed methods performed against the baseline methods using mean average precision (MAP), defined as follows:

$$MAP@N = \frac{1}{|U|} \sum_{u=1}^U \frac{1}{m} \sum_{n=1}^N P_u(n) \cdot rel_u(n). \quad (4.2)$$

$U$  denotes the total number of users, while  $m$  is the number of relevant items, and  $N$  is the number of desired recommendations.  $P$  is the precision, which is the ratio of simultaneously recommended and relevant items to the total number of recommended items ( $n$ ).  $rel(n)$  is either 0 or 1, indicating whether the  $n^{th}$  item is relevant.

Figure 4.3 shows the MAP at top 10 performances of the proposed and baseline approaches in the movie domain. The results indicate that LDSDMF significantly outperformed all other models with a decreasing rate as  $K$  increased (see Table 4.7 for significance values), followed by PMF and then ASEMF\_UIB and AMF interchangeably who performed better when  $K$  is high. SemEMF,

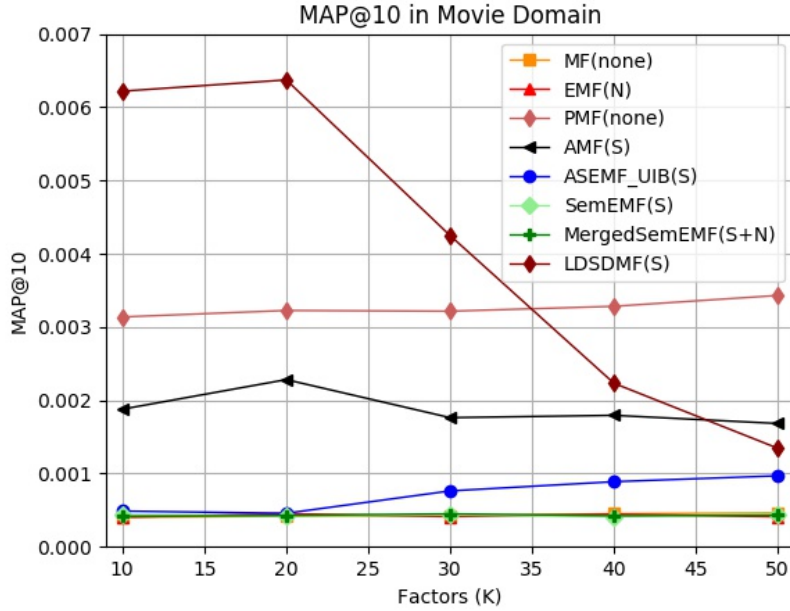


Figure 4.3: The graph shows the MAP@10 results of all methods while varying  $K$  in the movie domain.

MergedSemEMF, MF, and EMF performed similarly at a low rate, just above zero.

Table 4.7: MAP significance test results in the movie domain ( $K = 10$ ).

Model 1	Model 2	p-value
MF	<b>LDSDMF</b>	1.6e-15
EMF	<b>LDSDMF</b>	1.6e-15
PMF	<b>LDSDMF</b>	7.3e-09
AMF	<b>LDSDMF</b>	6.5e-11
ASEMF_UIB	<b>LDSDMF</b>	1.3e-12

Figure 4.4 depicts the MAP at top 10 results of the proposed methods and the baseline approaches. While varying  $K$ , our proposed methods ASEMUF\_UIB, SemEMF, MergedSemEMF, and LDSDMF, alongside the baseline methods MF and EMF, mutually compete in being the best model, followed by AMF and PMF, respectively.

## 4.4 Explainability Evaluation

Next, we measured our models using explainability metrics, MEP, MER, and xF-score, as follows [13] [14] [15]:

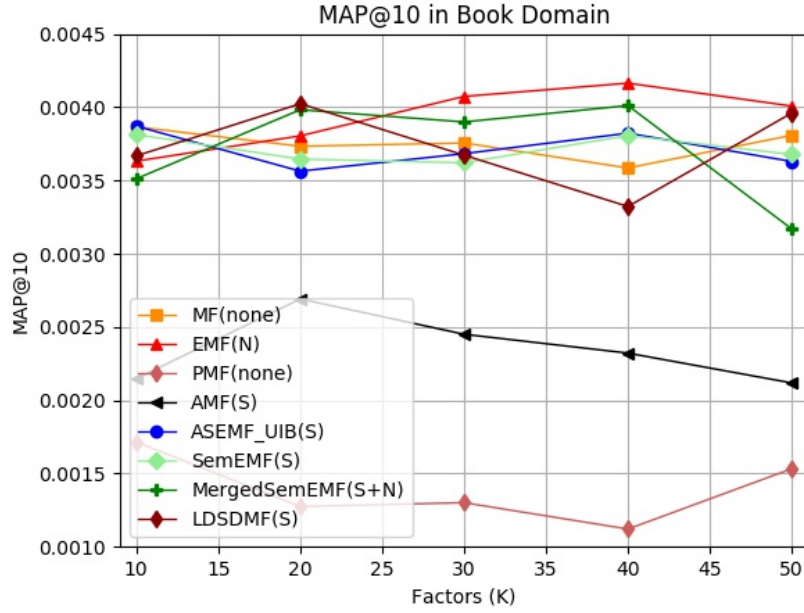


Figure 4.4: This graph shows the MAP@10 results of all methods while varying  $K$  in the book domain.

$$MEP = \frac{1}{|U|} \sum_{u \in U} \frac{|R \cap W|}{|R|} \quad (4.3)$$

$$MER = \frac{1}{|U|} \sum_{u \in U} \frac{|R \cap W|}{|W|} \quad (4.4)$$

$$xF - score = 2 * \frac{MEP * MER}{MEP + MER} \quad (4.5)$$

$U$  represents the set of users, while  $R$  is the set of recommended items, and  $W$  denotes the set of explainable items. MEP computes the proportion of simultaneously recommended and explainable items to the total number of recommended items across all users. Similarly, MER calculates the proportion of simultaneously recommended and explainable items to the total number of explainable items, averaged again, across all the users. xF-score is the harmonic mean of MEP and MER. We computed the above explainability metrics using the semantic and neighborhood explainability graphs as explained next, which capture what items are considered to be "explainable" in order to compute the explainability metrics.



#### 4.4.1 Semantic Explainability Metrics

We calculated the MEP, MER, and xF-score performance of the baseline methods and our proposed methods while varying  $\theta$ , which is a threshold for items to be considered semantically explainable when constructing the explainability graph (see equation 3.3). We varied  $\theta$  using the following values: 0, 0.1, 0.25, 0.5, 0.75, and 0.9. Figures 4.5 and 4.7 illustrate the results.

In Figure 4.5, three graphs show the performance of all models while varying  $\theta^s$  using MEP, MER, and xF-score metrics.  $\theta^s$  is a threshold for items to be considered semantically explainable in equation 3.3. The results illustrate that when  $\theta^s$  is set to 0, which means that all items (even those with a small explainability value) are considered explainable, the baseline PMF is the winner. However, when adding more restrictions to items to be considered semantically explainable, the proposed method, LDSDMF, significantly outperformed the other methods on all metrics (i.e., *MEP*, *MER*, and *xF-score*). Tables 4.8, 4.9, and 4.10 present the significance test results.

Table 4.8: MEP@10 significance test results in the movie domain ( $K = 10$  and  $\theta^s = 0.25$ ) using semantic KGs to calculate explainability metrics.

Model 1	Model 2	p-value
MF	<b>LDSDMF</b>	8.06e-23
EMF	<b>LDSDMF</b>	8.1e-23
PMF	<b>LDSDMF</b>	3.05e-17
AMF	<b>LDSDMF</b>	8.06e-23
ASEMF_UIB	<b>LDSDMF</b>	2.6e-20

Table 4.9: MER@10 significance test results in the movie domain ( $K = 10$  and  $\theta^s = 0.25$ ) using semantic KGs to calculate explainability metrics.

Model 1	Model 2	p-value
MF	<b>LDSDMF</b>	6.2e-21
EMF	<b>LDSDMF</b>	6.3e-21
PMF	<b>LDSDMF</b>	2.1e-15
AMF	<b>LDSDMF</b>	6.2e-21
ASEMF_UIB	<b>LDSDMF</b>	1.3e-19

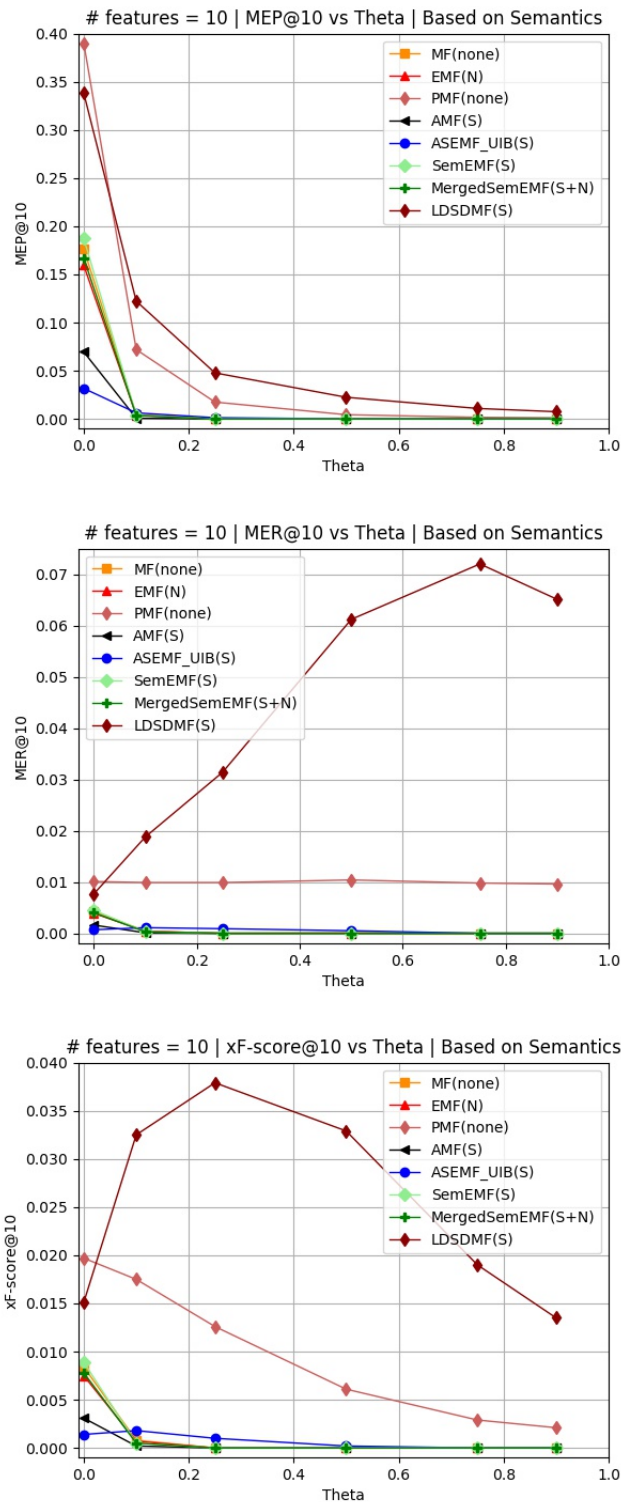


Figure 4.5: The upper graph shows the results of MEP@10 for all methods, while the middle one shows MER@10 for all methods, and the lower graph illustrates the results of all methods using the xF-score metric. All explainability metrics utilize semantic KGs. All the results are in the movie domain.

Table 4.10: xF-score@10 significance test results in the movie domain ( $K = 10$  and  $\theta^s = 0.25$ ) using semantic KGs to calculate explainability metrics.

Model 1	Model 2	p-value
MF	<b>LDSDMF</b>	1.1e-21
EMF	<b>LDSDMF</b>	1.1e-21
PMF	<b>LDSDMF</b>	5.1e-16
AMF	<b>LDSDMF</b>	1.1e-21
ASEMF_UIB	<b>LDSDMF</b>	5.6e-20

Table 4.11: MEP@10 significance test results in the book domain ( $K = 50$  and  $\theta^s = 0.25$ ) using semantic KGs to calculate explainability metrics.

Model 1	Model 2	p-value
MF	<b>ASEMF_UIB</b>	5.9e-08
EMF	<b>ASEMF_UIB</b>	1.5e-07
PMF	<b>ASEMF_UIB</b>	1.1e-13
AMF	<b>ASEMF_UIB</b>	6.9e-21
LDSDMF	<b>ASEMF_UIB</b>	4.1e-07

Table 4.12: MER@10 significance test results in the book domain ( $K = 50$  and  $\theta^s = 0.25$ ) using semantic KGs to calculate explainability metrics.

Model 1	Model 2	p-value
MF	<b>ASEMF_UIB</b>	1.3e-07
EMF	<b>ASEMF_UIB</b>	2.9e-08
PMF	<b>ASEMF_UIB</b>	1.2e-12
AMF	<b>ASEMF_UIB</b>	5.5e-21
LDSDMF	<b>ASEMF_UIB</b>	4.8e-06

Table 4.13: xF-score@10 significance test results in the book domain ( $K = 50$  and  $\theta^s = 0.25$ ) using semantic KGs to calculate explainability metrics.

Model 1	Model 2	p-value
MF	<b>ASEMF_UIB</b>	1.05e-07
EMF	<b>ASEMF_UIB</b>	3.07e-08
PMF	<b>ASEMF_UIB</b>	1.01e-12
AMF	<b>ASEMF_UIB</b>	5.5e-21
LDSDMF	<b>ASEMF_UIB</b>	3.9e-06

Figure 4.6 illustrates the MEP, MER, and xF-score performance of all models against  $K$ . As shown in the three graphs, the blue line, which indicates the model ASEMFIUIB, is the winner with a significance values as shown in Tables 4.11, 4.12, and 4.13, meaning that it recommends more semantically explainable items in the top 10 list than all models. ASEMFIUIB learns about

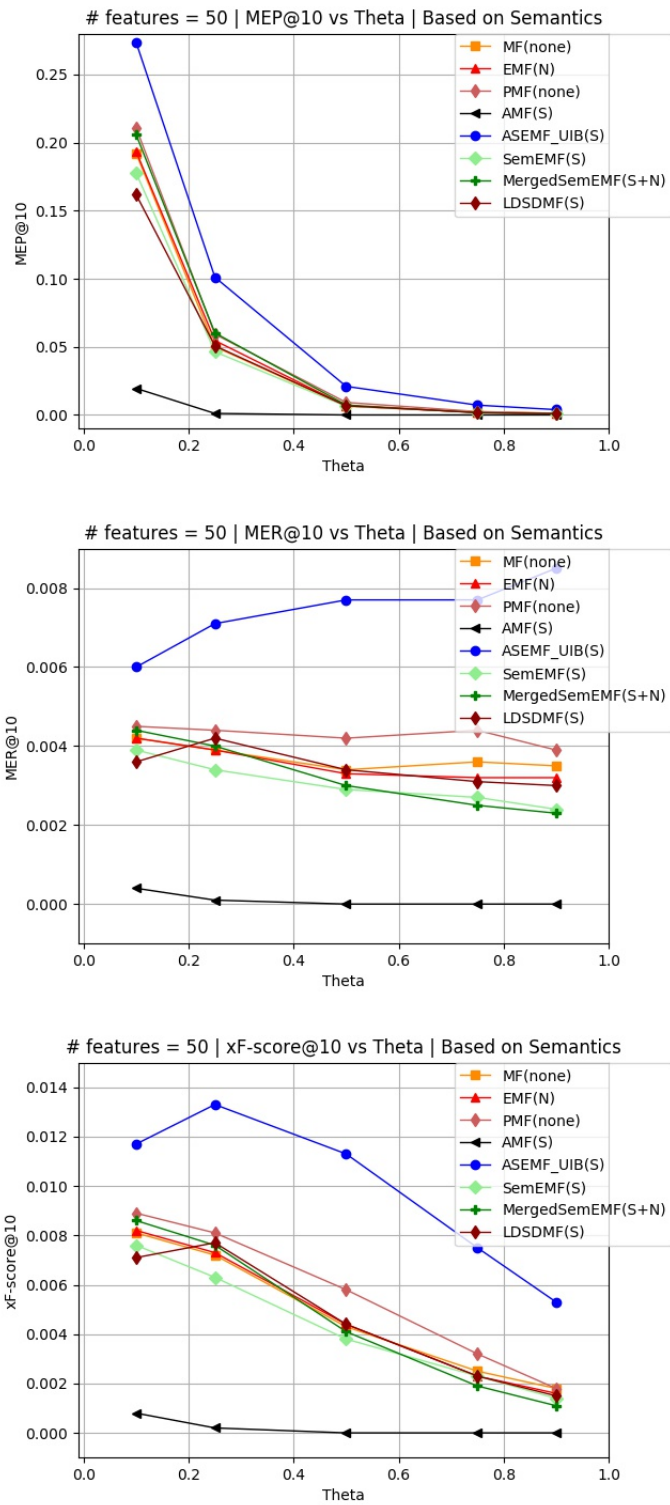


Figure 4.6: The upper graph shows the results of MEP@10 for all methods, while the middle one shows MER@10 for all methods, and the lower graph illustrates the results of all methods using the xF-score metric. All explainability metrics utilize semantic KGs. All the results are in the book domain.

one property in the latent space making the model more focused on it especially with sparse data as the case with book domain. Followed by the rest of the baseline models that performed similarly. AMF’s performance was the lowest in all metrics.

#### 4.4.2 Neighborhood Explainability Metrics

The previous results show the performance when explainability metrics were computed based on the semantic graph. In this section, we examine how they perform when using the neighborhood explainability graph for the explainability metrics. In Figure 4.7, three graphs show the performance of all models while varying  $\theta^n$ .  $\theta^n$  is a threshold for items to be explainable based on the neighborhood technique used in the baseline EMF [13] [14] [15]. Recall that formula for generating the neighborhood-based explainability matrix (Equation 2.15) is

$$W_{ui} = \begin{cases} \frac{|N'(u)|}{|N_k(u)|} & \text{if } \frac{|N'(u)|}{|N_k(u)|} > \theta^n \\ 0 & \text{otherwise,} \end{cases} \quad (4.6)$$

where  $N'(u)$  denotes the set of neighbors of user  $u$ , who rated item  $i$ , and  $N_k(u)$  depicts the list of the  $k$  nearest neighbors of  $u$ .

Table 4.14: MEP@10 significance test results in the movie domain ( $K = 10$  and  $\theta^n = 0.25$ ) using the neighborhood based explainability metrics.

Model 1	Model 2	p-value
MF	<b>LSDMF</b>	1.9e-21
EMF	<b>LSDMF</b>	1.9e-21
PMF	<b>LSDMF</b>	3.9e-17
AMF	<b>LSDMF</b>	1.2e-13
ASEMF_UIB	<b>LSDMF</b>	9.9e-19

Table 4.15: MER@10 significance test results in the movie domain ( $K = 10$  and  $\theta^n = 0.25$ ) using the neighborhood based explainability metrics.

Model 1	Model 2	p-value
MF	<b>LSDMF</b>	1.2e-21
EMF	<b>LSDMF</b>	1.2e-21
PMF	<b>LSDMF</b>	1.4e-15
AMF	<b>LSDMF</b>	5.3e-15
ASEMF_UIB	<b>LSDMF</b>	5.9e-19

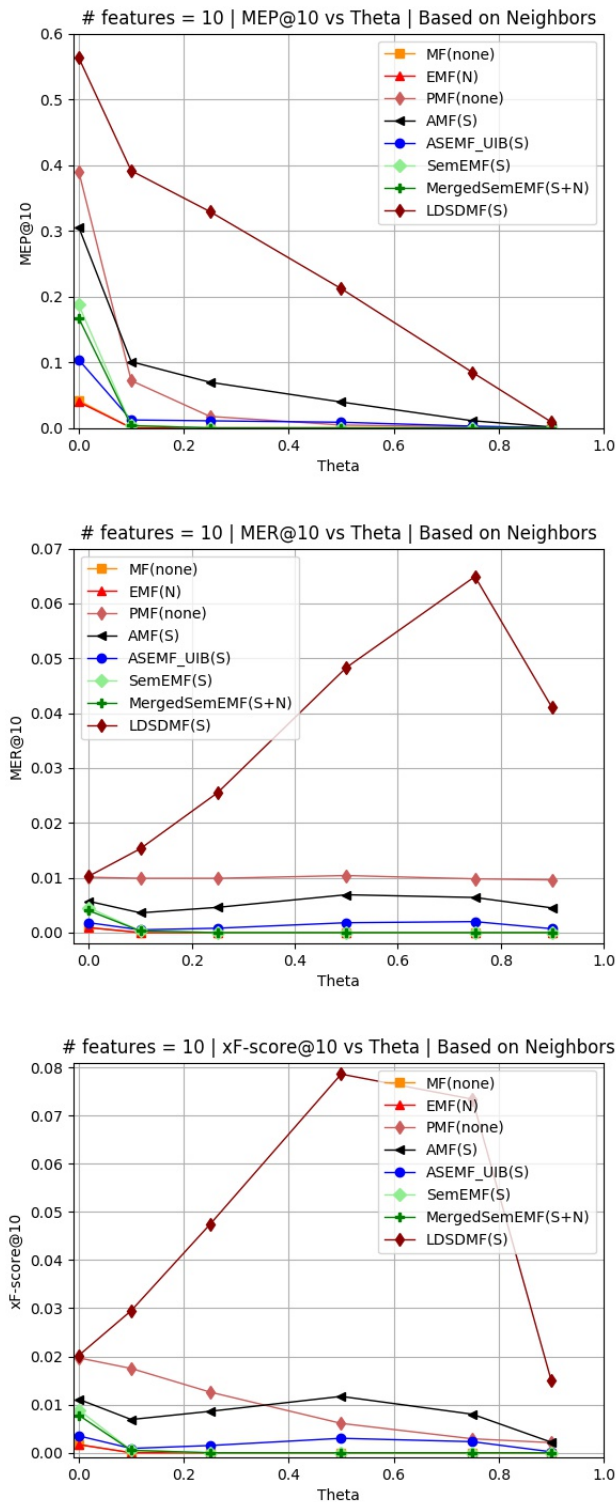


Figure 4.7: The upper graph shows the results of MEP@10 for all methods, while the middle one shows the MER@10 results for all methods, and the lower graph illustrates the results of all methods. All explainability metrics are based on the neighborhood explainability graph. All the results are in the movie domain.

Table 4.16: xF-score@10 significance test results in the movie domain ( $K = 10$  and  $\theta^n = 0.25$ ) using the neighborhood based explainability metrics.

Model 1	Model 2	p-value
MF	<b>LDSDMF</b>	1.1e-21
EMF	<b>LDSDMF</b>	1.1e-21
PMF	<b>LDSDMF</b>	9.2e-16
AMF	<b>LDSDMF</b>	6.4e-15
ASEMF_UIB	<b>LDSDMF</b>	5.9e-19

Figure 4.7 illustrates the performance of all methods when evaluated using MEP, MER, and xF-score metrics computed using the neighborhood explainability graph in the movie domain. Our model, LDSDMF, significantly exceeded all baseline methods on all three explainability metrics (see Tables 4.14, 4.15, and 4.16 for significance test results).

This observation shows that our proposed method recommends more accurate and more explainable items based on semantic KGs and neighborhood-based explainability graphs than all the baseline methods.

Figure 4.8 shows three graphs of the MEP, MER, and xF-score performance of all models when these metrics use the neighborhood explainability graph in the book domain. PMF was the winner, followed by AMF, while the performance of all other models was low. This means that PMF and AMF succeeded in recommending more neighborhood-based explainable items than all the other models. The lack of extra information that semantic KGs provide caused all proposed models not to perform well in this type of evaluation.

## 4.5 Analysis of Results

The experimental results showed that adding semantics resulted in improved recommendation and explainability. Although one would expect explainability to decrease recommendation accuracy, it is important to note that our proposed methods utilize more semantic data than mere user ratings. This additional data compensates for lack of rating and sparseness of data, thus improving accuracy.

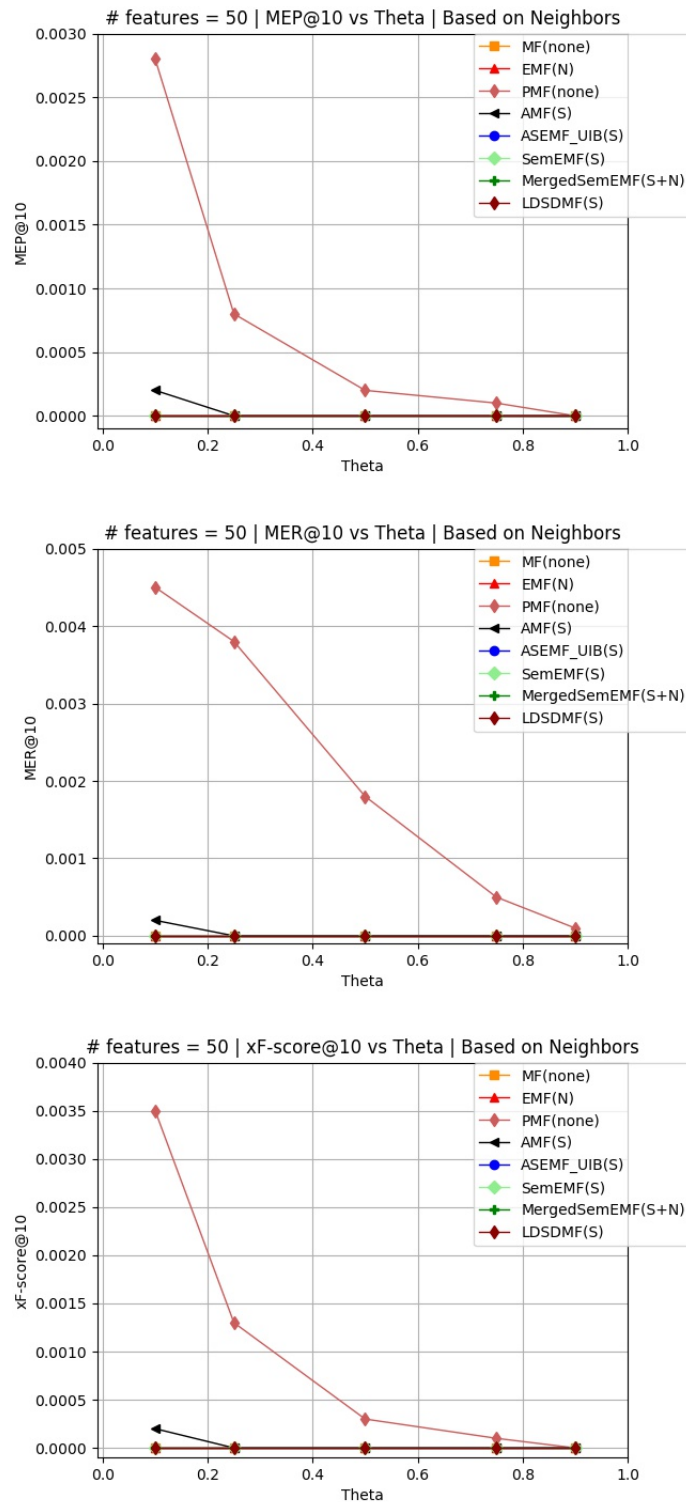


Figure 4.8: The upper graph shows the results of MEP@10 for all methods, while the middle one shows MER@10 for all methods, and the lower graph illustrates the results of all methods using the xF-score metric. All explainability metrics use semantic KGs. All the results are in the book domain.



## 4.6 Real User Study

In this section, we validate the explainability of the model, proposed in Sections 3.9 and 3.2.1, by conducting a user study experiment in the movie domain. Given the semantic KGs defined in Equations 3.1, 3.2, and 3.3, our research questions are as follows:

- RQ1: Does the number of semantic attributes used in the explanation, whether it is low or high, impact user satisfaction? Satisfaction is defined as the ease of system usability and the enjoyment of use [148].
- RQ2: Does an explanation that uses a higher number of semantic attributes increase perceived transparency? Transparency is providing information to the user so he or she can comprehend how the system works and the justification behind the recommendation [148].
- RQ3: Does the number of semantic attributes used in the explanation (Low (1 semantic attribute), Medium (3 semantic attributes), or High (5 semantic attributes)) impact the perceived effectiveness? Effectiveness is defined as the ability of the explanation to help users make good decisions [148].

### 4.6.1 Hypothesis

Suppose that a recommender system recommends two items  $i_1$  and  $i_2$  alongside their explanations. Given the explanation definition in Sections 3.9 and 3.2.1 and Equations 3.1, 3.2, and 3.3, if  $i_1$  uses more semantic attributes in the explanations than  $i_2$  (Figure 4.9), does recommending  $i_1$  result in a better satisfaction than recommending  $i_2$  from the user perspective? Our hypothesis can be summarized as follows: Recommending an item with explanation that shows more semantic attributes will lead to higher user satisfaction.

### 4.6.2 Methods

A web app platform, similar to commercial movie recommender engines used by Netflix, Amazon Video, and Hulu, was designed to conduct the study. The application used the MovieLens benchmark data set <sup>3</sup>.

---

<sup>3</sup><https://grouplens.org/datasets/movielens/>

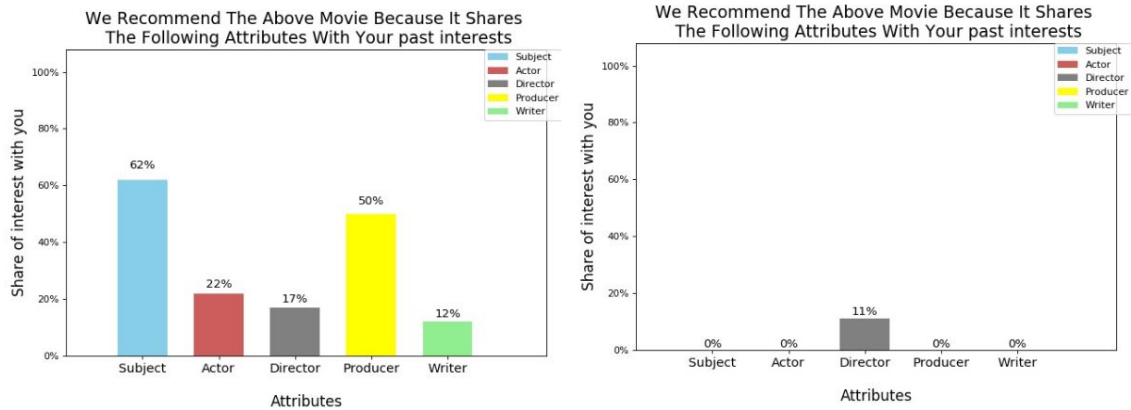


Figure 4.9: A comparison of two explanations, high on the left and low on the right, that are exposed to the user during the experiment. The explanation on the left shows more semantic attributes than the one on the right.

The explanations are divided into three groups based on the number of semantic attributes randomly chosen to explain the recommended movie as follows:

- Low: Up to one semantic attribute used for explanation.
- Medium: Up to three semantic attributes used for explanation.
- High: Up to five semantic attributes used for explanation.

### 4.6.3 Subject Recruitment

The Institutional Review Board at University of Louisville reviewed and authorized our study. Participants were students in a large urban, southern university and were recruited to participate in the study via personal and email invitations. A Surface Pro laptop and a desktop were provided to the participants to use for this experiment. Google forms was used to construct and host the questions and the results were securely stored on Google drive.

### 4.6.4 Sample Size Estimation

To estimate the sample size, we performed a statistical power analysis. The effect size in this study is large using Cohen's [149] criteria. When  $\alpha$  is set to 0.05, and power is set to 0.8, the sample size needed is approximately 10.

The 34 participants were randomly assigned to either the *low*, *medium*, or *high* group representing the number of semantic attributes used in explanation. The number of people in each group are as follows:

- Low = 11
- Medium = 12
- High = 11

#### **4.6.5 Procedures**

The process of the experiment was as follows:

1. The participant is asked to rate from 1 to 5 at least 10 movies they have watched previously from a selection of movies.
2. Based on the group the participant was assigned to, a recommendation alongside an explanation will be provided to the user.
3. The recommendation and explanation will be selected from a pool of recommendations that are calculated using the method proposed in Sections 3.9 and 3.2.1, such that the correct number of semantic attributes to be used in the explanation is displayed to the user depending on the experimental group that the participant was assigned to (i.e. “low (1)”, “medium (3)”, or “high (5)”).
4. The participant is asked to fill out a Likert Scale questionnaire. Table 4.17 shows the questions used in this study.
5. Demographic information is collected from the participant including age, gender, major of study, weekly hours watching movies, and favorite movie attributes. Table 4.18 presents the questions used in this experiment.

This information is requested to study potential confounding factors on the participant’s satisfaction with the explanations. A snapshot of the application is shown in Figures 4.10 and 4.11. The duration of the experiment is around 30 minutes.

Table 4.17: Likert scale survey questions.

Question 1	"Based on the share of semantic attributes between the recommended movie and your interest in these semantic attributes, this is a good recommendation."
Question 2	"This explanation helps me understand why this movie was recommended."
Question 3	"Based on the share of semantic attributes between the recommended movie and my interest in these semantic attributes, I will watch this movie."
Question 4	"Based on the share of semantic attributes between the recommended movie and my interest in these semantic attributes, I can determine how well I will like this movie."
Question 5	"This explanation helps me understand how the recommender system works."

Table 4.18: Demographic questions.

Question 1	"What is your gender?"
Question 2	"What is your age?"
Question 3	"What is your major of study?"
Question 4	"How many hours per week do you watch movies on average?"
Question 5	"What are the most influential attributes that encourage you to watch a movie?"
Question 6	"How familiar are you with automated recommender systems?"
Question 7	"Check all the online entertainment services that you have used in the past."

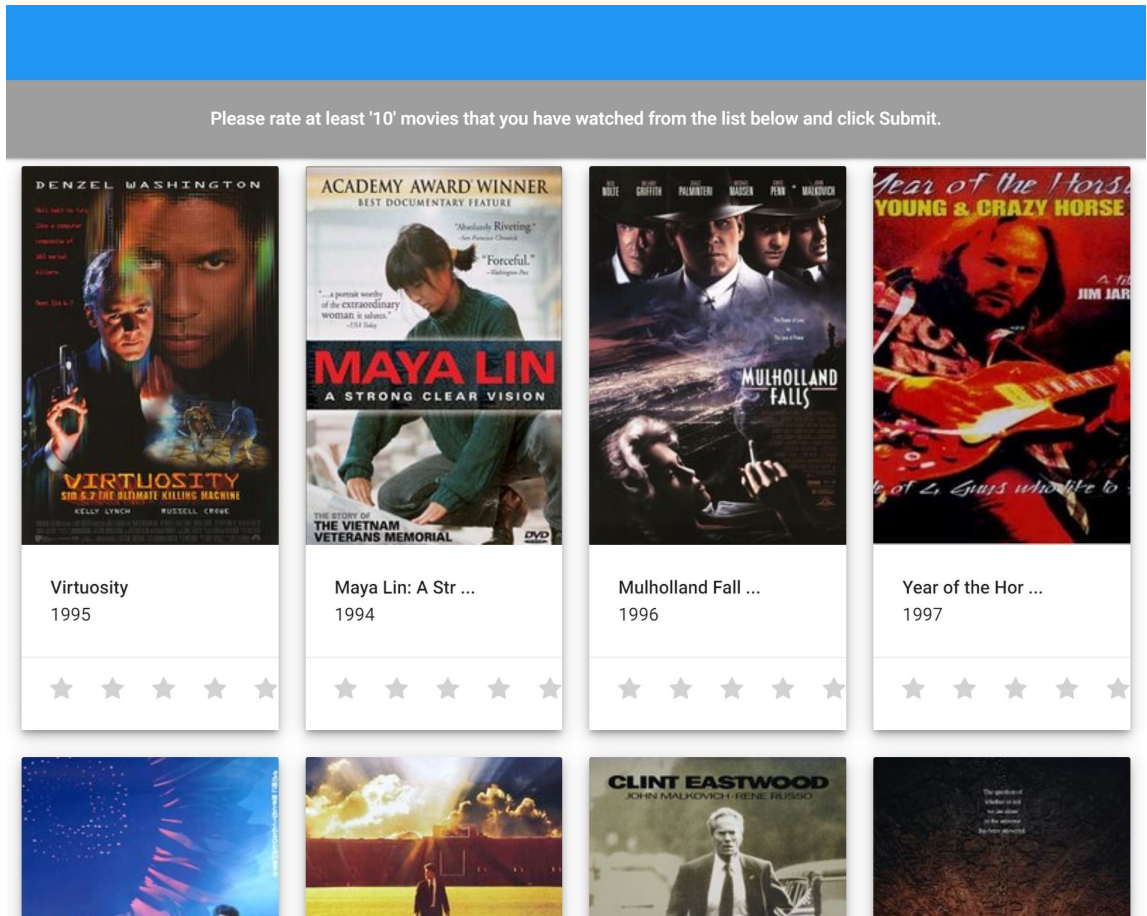


Figure 4.10: A snapshot of the recommender system app showing a list of movies for the user to rate.

Our recommendation is:

Leaving Las Veg ..., 1995



Our explanation for this recommendation is:

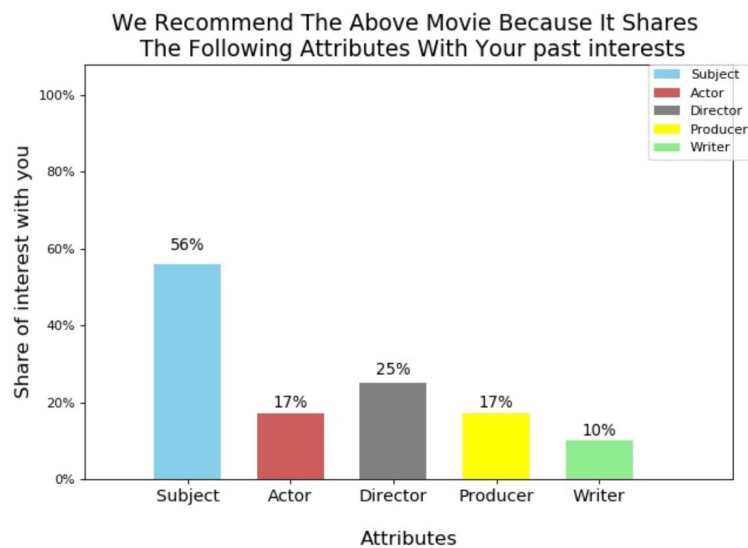


Figure 4.11: A snapshot of a recommendation and its semantic explanation presented to a user. The share of interest is the likeability degree computed using Equation 3.2

#### 4.6.6 Analysis of Results

In this study, participants were asked to answer five questions regarding their experience after using the model. Figure 4.12 shows a vertical bar chart for all participants' answers to all of Table 4.17's five questions. The most repeated answer was "Somewhat Agree" across all questions, followed by the "Neutral" answer option, then by "Strongly Agree" respectively. The answers "Somewhat Disagree" and "Strongly Disagree" were the least chosen answers by all participants for all questions. Figure 4.13 depicts the answers for participants in the group High. People were assigned randomly to each group. The answers "Strongly Agree" and "Somewhat Agree" were the most popular answers to all questions. Only four participants were neither agreeing nor disagreeing to question one, and only one participant disagreed in questions four and five. Worth noting that more than half of the participants strongly agreed to question two, which is about how the explanation helped them understand the recommendation. Figure 4.14 shows the answers of participants in the group Medium. "Somewhat Agree" and "Neutral" responses were the most chosen responses by people in this group. Followed by "Strongly Agree" and "Somewhat Disagree". Two participants preferred "Strongly Disagree" as their answer to question three. In the group Low, as shown in Figure 4.15, more than half of the participants chose "Strongly Disagree" and "Somewhat Disagree" as their answers to all questions. The next response in line was the Natural response choice, followed by "Somewhat Agree", and only one participant gave a "Strongly Agree" answer to question five in this group.

Figure 4.16 depicts a Heat map plot showing the distribution of all answers to all questions by all participants. The most popular answer is "Somewhat Agree" followed by "Neutral" as the second most popular. "Strongly agree" is next in line, then "Somewhat Disagree" and "Strongly Disagree" answers were the least preferred answers by participants in the Medium group.

Figure 4.17 shows the responses from participants in the group High. The figure shows a clear tendency to the Agree than to the Disagree answers. In contrast, responses from participants in the group Low, as illustrated in Figure 4.19, tend to the Disagree side more than the Agree side. Lastly, the heat map in Figure 4.18 is scattered over all responses to all question from participants in the group Medium.

Figure 4.20 indicates the satisfaction level with the explanation for all participants in this study.

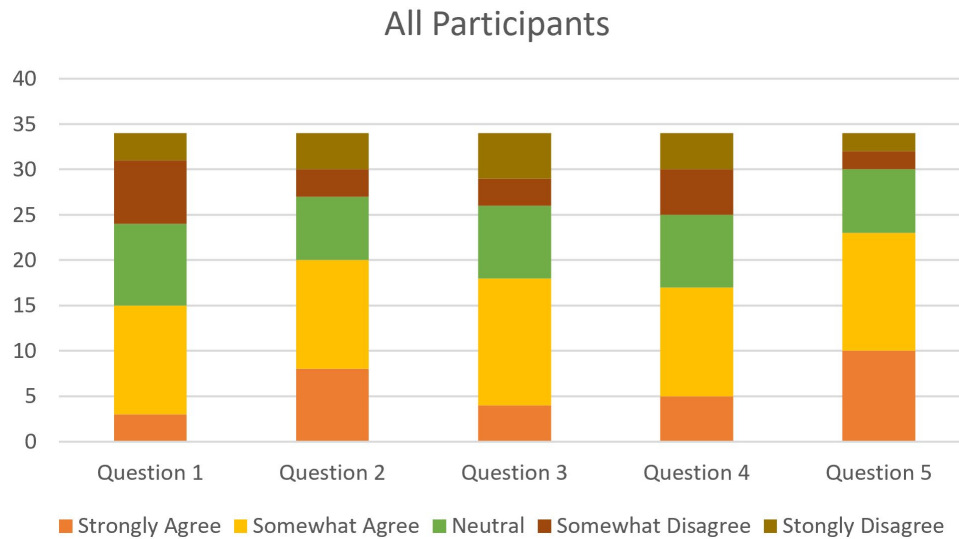


Figure 4.12: A Vertical bar chart of the answers to the questions in Table 4.17 for all participants.

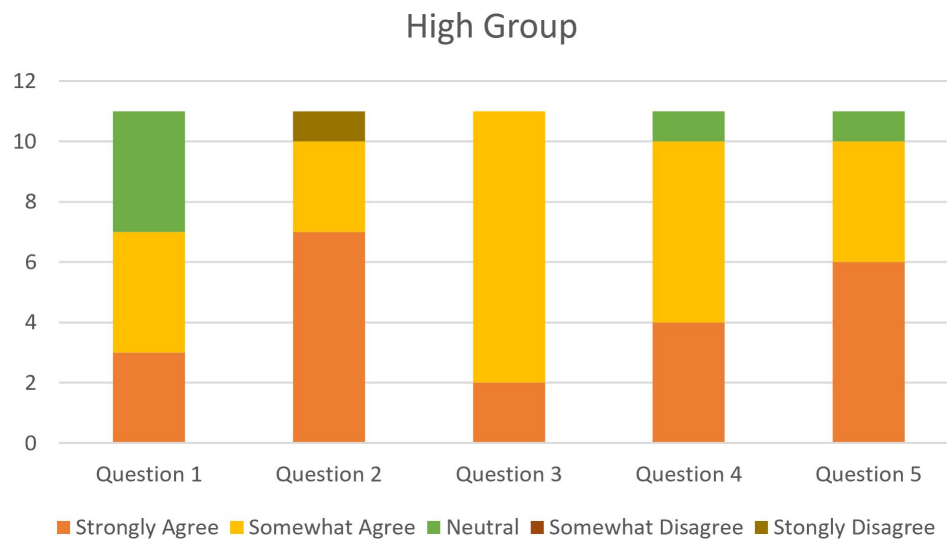


Figure 4.13: A Vertical bar chart of the answers to the questions in Table 4.17 for participants in the group "High".



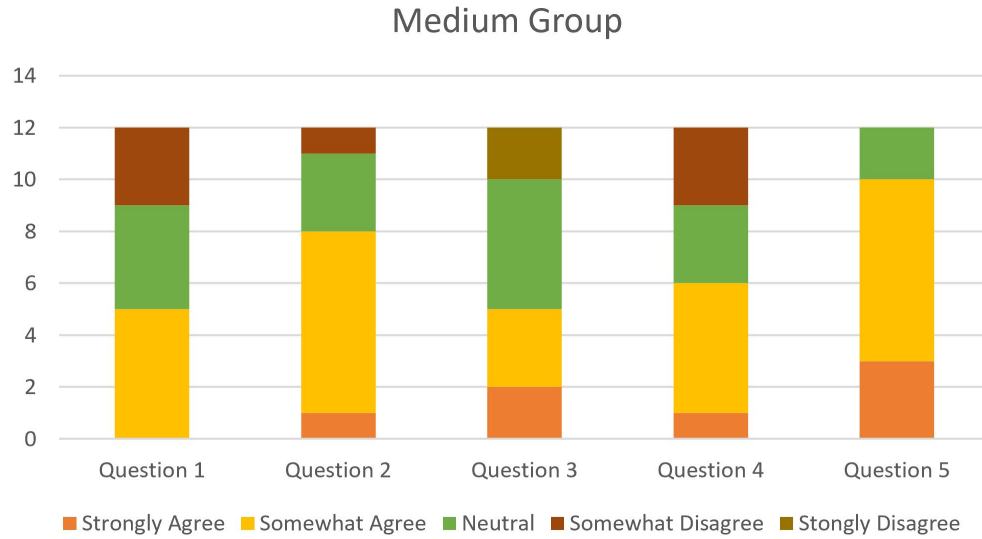


Figure 4.14: A Vertical bar chart of the answers to the questions in Table 4.17 for participants in the group "Medium".

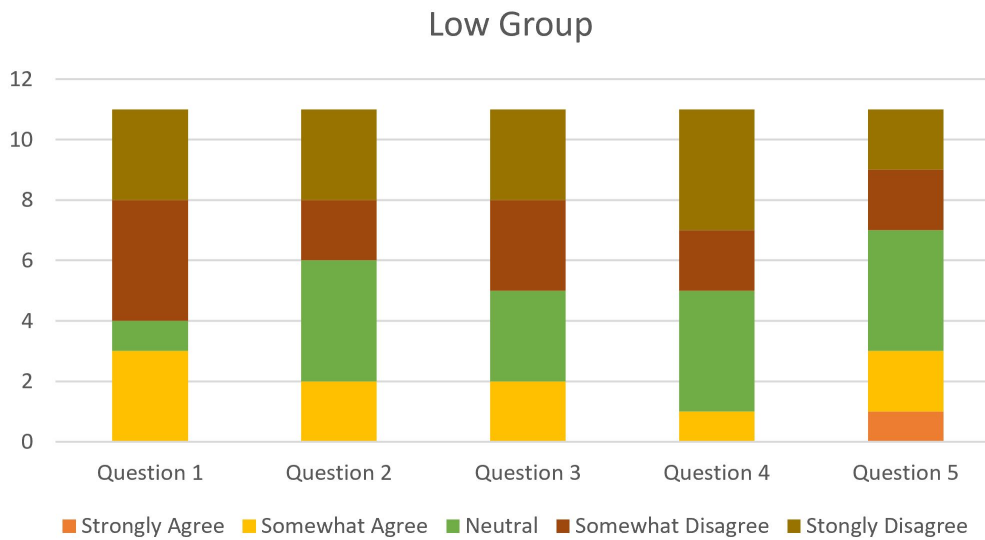


Figure 4.15: A Vertical bar chart of the answers to the questions in Table 4.17 for participants in the group "Low".

Strongly Agree	3	8	4	5	10
Somewhat Agree	12	12	14	12	13
Neutral	9	7	8	8	7
Somewhat Disagree	7	3	3	5	2
Strongly Disagree	3	4	5	4	2
	Question 1	Question 2	Question 3	Question 4	Question 5

Figure 4.16: A Heat-map plot of the answers to the questions in Table 4.17 for all participants.

Strongly Agree	3	7	2	4	6
Somewhat Agree	4	3	9	6	4
Neutral	4	0	0	1	1
Somewhat Disagree	0	0	0	0	0
Stongly Disagree	0	1	0	0	0
	Question 1	Question 2	Question 3	Question 4	Question 5

Figure 4.17: A Heat-map plot of the answers to the questions in Table 4.17 for participants in the group "High".

Strongly Agree	0	1	2	1	3
Somewhat Agree	5	7	3	5	7
Neutral	4	3	5	3	2
Somewhat Disagree	3	1	0	3	0
Stongly Disagree	0	0	2	0	0
	Question 1	Question 2	Question 3	Question 4	Question 5

Figure 4.18: A Heat-map plot of the answers to the questions in Table 4.17 for participants in the group "Medium".

Strongly Agree	0	0	0	0	1
Somewhat Agree	3	2	2	1	2
Neutral	1	4	3	4	4
Somewhat Disagree	4	2	3	2	2
Stongly Disagree	3	3	3	4	2
	Question 1	Question 2	Question 3	Question 4	Question 5

Figure 4.19: A Heat-map plot of the answers to the questions in Table 4.17 for participants in the group "Low".

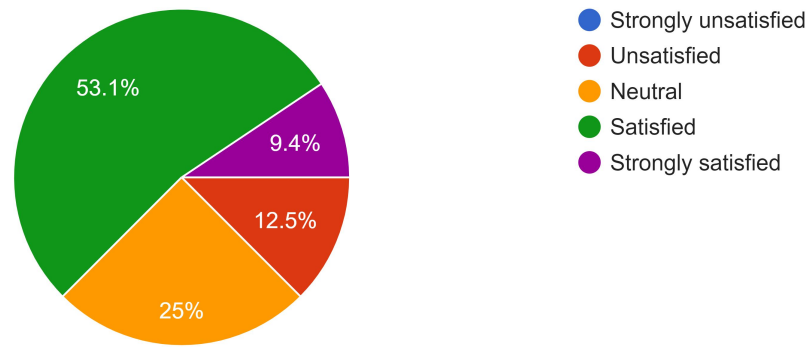


Figure 4.20: Distribution of the participants' satisfaction with the explanation.

More than half of them were satisfied, whereas around 10% were strongly satisfied. 25% of the participants were neither satisfied nor unsatisfied, and 12.5% were not satisfied. No participant responded with the strongly unsatisfied answer option.

Figures 4.21, 4.22, 4.23, 4.24, 4.25, 4.26, and 4.27, show the responses of all participants to the demographic questions in Table 4.18. The answers for these questions were optional.

Three-quarters of the participants were males and the rest were females as shown in Figure 4.21. Figure 4.22 represents the age distribution of all participants. More than 60% are between the age of 25 and 34 years, followed by 20% participants aged between 35 and 44 years. The rest of the participants' ages are distributed in the other groups.

The majority of the volunteers' major of study is Computer science followed by other majors as shown in Figure 4.23. Most of the participants watch movies for around 0 to 5 hours a week as reported in Figure 4.24. Figure 4.26 shows half of the volunteers were either moderately or somewhat familiar with the automated recommender systems, whereas 32.4% are slightly familiar. 14.7% were extremely familiar and a small portion of the participants were not familiar at all.

Figure 4.25 denotes the distribution of the participants regarding the most influential semantic attributes that encourage them to watch a movie. It indicates that subject, genre and actor were the most influential ones, however, producer and music-composer were the least influential. In Figure 4.27 represents the online entertainment services that the volunteers have used in the past. YouTube and Netflix are the most popular services followed by Amazon Video and Hulu. Google Play and HBO were the least used services by participants.

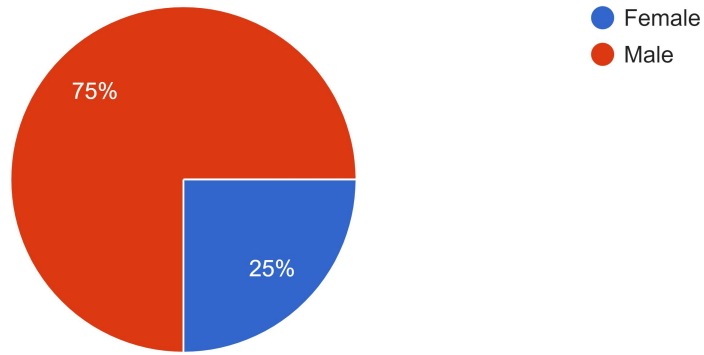


Figure 4.21: Distribution of the participants' gender.

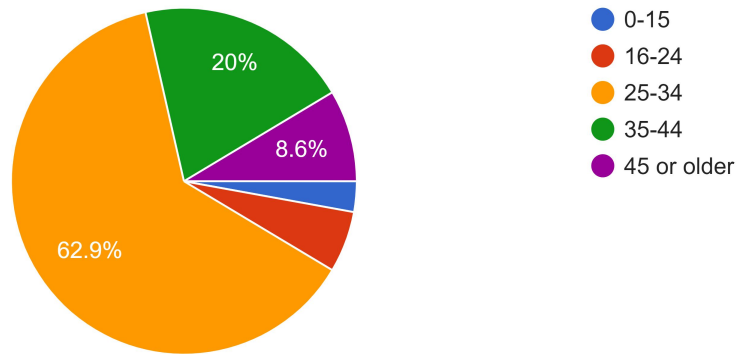


Figure 4.22: Distribution of the participants' age.

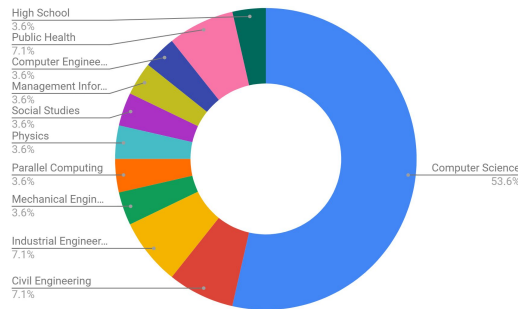


Figure 4.23: Distribution of the participants' major of study.

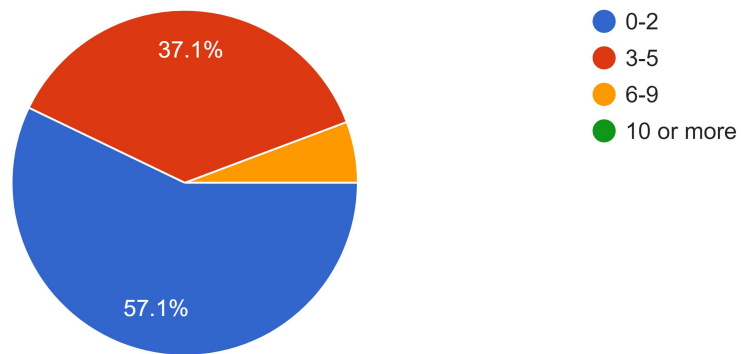


Figure 4.24: Distribution of the participants' weekly hours watching movies.

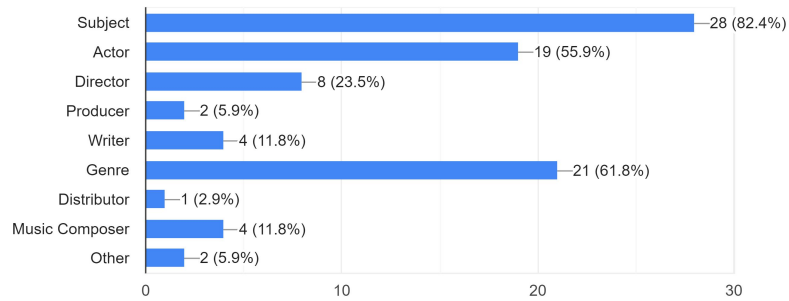


Figure 4.25: Distribution of the participants' favorite movies' attributes.

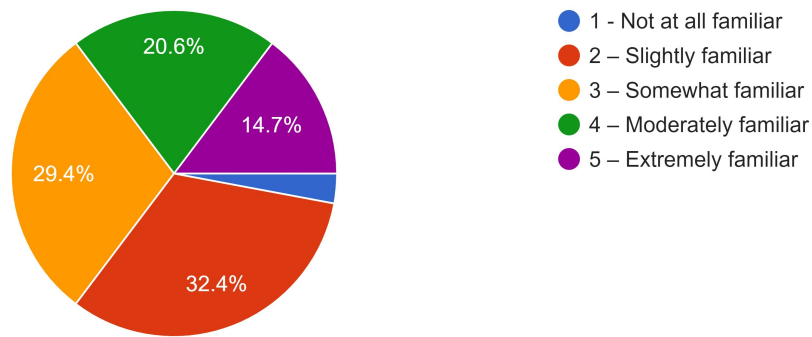


Figure 4.26: Distribution of the participants' familiarity with recommender systems.

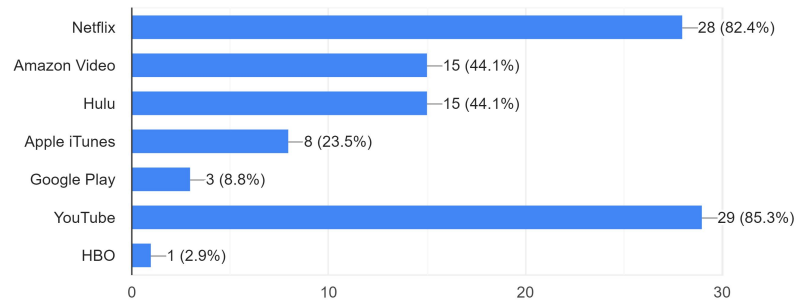


Figure 4.27: Distribution of the participants' most used online entertainment services.

#### 4.6.7 Hypothesis Testing

In the previous section 4.6.6, we showed how the responses of the participants varied according to the designated groups (High, Medium, and Low) where participants were assigned randomly. The plots indicate that people in the group "High" tend to give more positive responses than others in the other groups.

In this section, analytical testing is conducted to determine the significance of the those findings of this study. First of all, it is essential to evaluate the reliability of the Likert scale questionnaire by calculating Cronbach's Alpha [150]. The correlation of the survey questions and the 34 participants was 0.86, which is above the threshold of 0.7 for an acceptable level of reliability.

Table 4.20 presents the relationship between the explanation aspects, satisfaction, transparency, and effectiveness, and the questions in the survey listed in Table 4.17.

An Analysis of Variance (ANOVA) test is conducted to study the effect of the explainability variable on the designated aspects in Table 4.17. In this statistical test, The null hypothesis is that the mean of the three groups, high, medium, and low, are equal.

- Satisfaction:

Questions 1 and 3 evaluate the participant's satisfaction with the explanation in the ANOVA test. The degree of freedom is 2, f-value is 9.273, and the p-value is 0.0006. The p-value is less than 0.05 threshold, indicating that there exists a significant difference between the three groups, hence, a significant correlation between satisfaction and explainability. The eta-squared measure of effect size is 0.374. We conducted a Tukey's HSD (Honestly Significant Difference) post-hoc test to determine which pair of groups were significantly different from each other. The Family-wise significance interval was at 95%, and Table 4.21 summarizes the results. From this table, it is clear that there is a significant difference between group High and group Low with a very small p-value resulting in rejecting the null hypothesis. However, there is no significant difference between group High and group Medium nor between group Medium and group Low. Figure 4.28 shows a visualization of the differences between the means of the three groups.

- Transparency

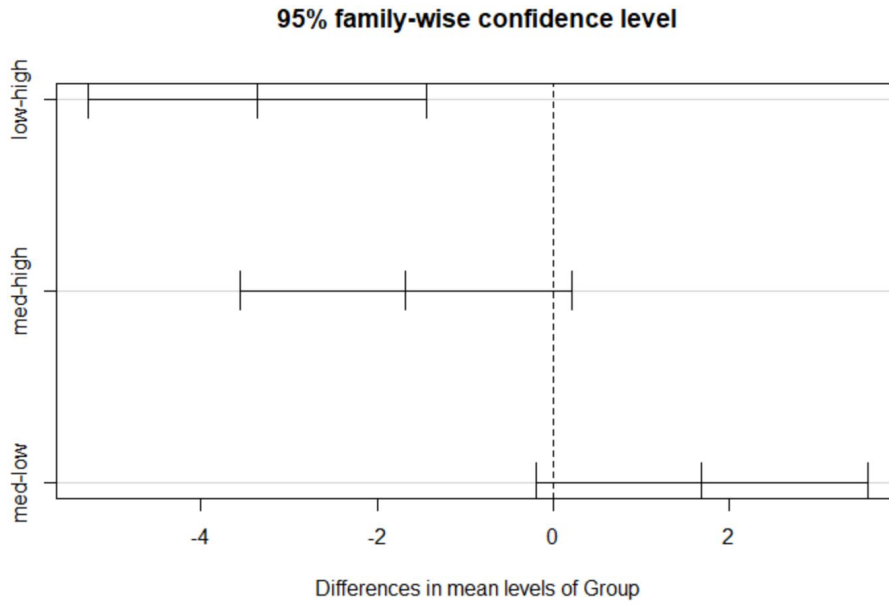


Figure 4.28: Visualization of differences of mean levels of pairs of groups for **satisfaction**.

Questions 2 and 5 evaluate the participants assessment of the recommendation transparency in the ANOVA test. The degree of freedom is 2, f-value is 14.491, and the p-value is  $3.6 \times 10^{-5}$ . The eta-squared degree of effect size is 0.483. With a family-wise significance interval at 95%, we conducted a Tukey’s HSD post-hoc test to decide which pairs of groups were significantly distinct from each other. Table 4.22 presents the outcome. As shown in this table, with a very small adjusted p-value, there is a significant difference between the groups High and Low, as well as between the groups Medium and Low. Nevertheless, there is no significant difference between the groups Medium and High. Figure 4.29 displays a visualization of the mean differences between the groups.

- Effectiveness

Effectiveness of the explanation is measured by the perceived responses to Question 4 in Table 4.17. We conducted the ANOVA test, and the results are as follows: The degree of freedom is 2, f-value is 14.123, and the p-value is  $4.3 \times 10^{-5}$ . The eta-squared degree of effect size is 0.476. A Tukey’s HSD post-hoc test is conducted to determine if there exists any significant signed difference between the groups. Table 4.23 indicates that there is a significant signed difference between groups High and Low as well as between groups Medium and Low. p-values are below the threshold of 0.05, resulting in rejection of the null hypothesis. Mean-

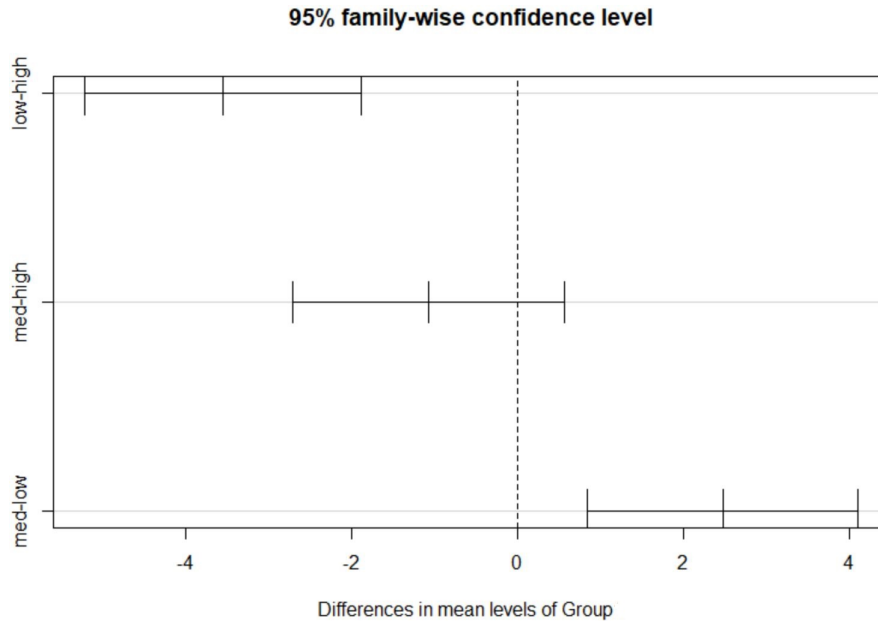


Figure 4.29: Visualization of differences of mean levels of pairs of groups for **transparency**.

Table 4.19: Mean and standard deviation for all groups for regarding all three explanation aspects.

Explanation Aspect	Groups					
	High		Medium		Low	
	Mean	STD	Mean	STD	Mean	STD
Satisfaction	8.09	0.94	6.41	2.02	4.72	2.24
Transparency	8.81	1.32	7.75	1.35	5.27	2
Effectiveness	4.27	0.64	3.33	0.98	2.18	2.1

while, the relationship between groups High and Medium shows no sign of any significance.

Figure 4.30 presents the differences in means between the three designated groups.

Table 4.19 shows the mean and standard deviation for all groups regarding the tested explanation styles, satisfaction, transparency, and effectiveness.

Table 4.20: Categorization of the survey questions from Table 4.17 according to the research questions.

Explanation Aspect	Question
Satisfaction	1 and 3
Transparency	2 and 5
Effectiveness	4



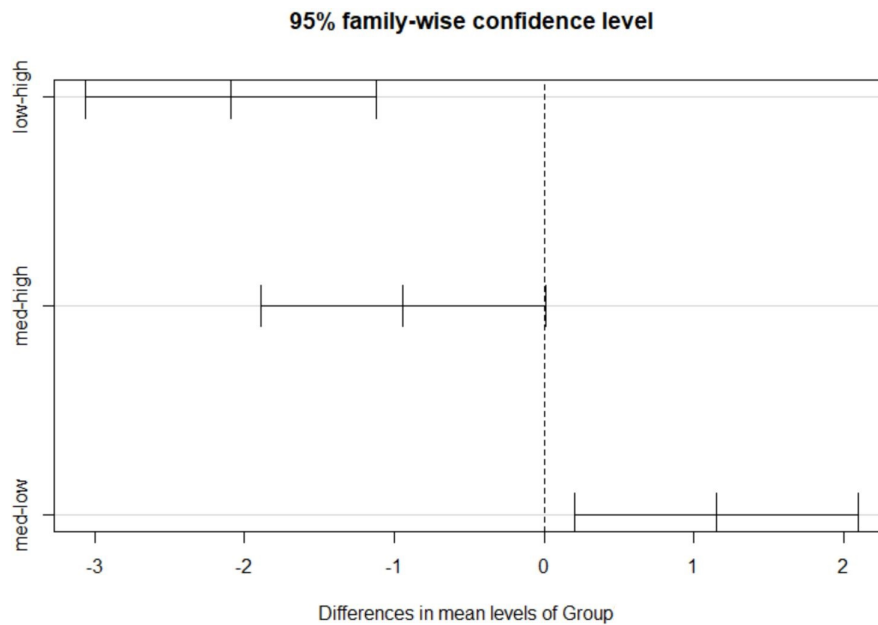


Figure 4.30: Visualization of differences of mean levels of pairs of groups for **effectiveness**.

Table 4.21: Tukey multiple comparisons of means at 95% family-wise confidence interval for **satisfaction**.

Group pairs	Difference	Adjusted p-value	reject
High-Low	-3.3636	0.0004	True
High-Medium	-1.6742	0.0888	False
Medium-Low	1.6893	0.0852	False

Table 4.22: Tukey multiple comparisons of means at 95% family-wise confidence interval for **transparency**.

Group pairs	Difference	Adjusted p-value	reject
High-Low	-3.5454	0.0000	True
High-Medium	-1.0681	0.2555	False
Medium-Low	2.4772	0.0021	True

Table 4.23: Tukey multiple comparisons of means at 95% family-wise confidence interval for **effectiveness**.

Group pairs	Difference	Adjusted p-value	reject
High-Low	-2.0909	0.0000	True
High-Medium	-0.9393	0.0529	False
Medium-Low	1.1515	0.0147	True

## **4.7 Summary of Experimental Evaluation**

In this chapter, we presented the results of an offline and online evaluation of the methods proposed in Chapter 3. In offline evaluation, we used objective metrics to measure the recommendation accuracy of the proposed methods as well as the explainability. The overall results indicate that our model succeeded in increasing the explainability of the system while keeping the error rate at a low level.

In the online evaluation, the final results indicate that the participants had a good perception of the explanation capability, especially when including more item properties in the explanation generation process.

## CHAPTER 5

### CONCLUSION AND FUTURE WORK

As recommendation systems become an essential component of big data and artificial intelligence (A.I.) systems, and as these systems embrace more and more sectors of society, it is becoming ever more critical to build trust and transparency into machine learning algorithms without significant loss of prediction power.

Our research harnesses the power of A.I., such as Knowledge Graphs and semantic inference, to help build explainability into accurate Black Box predictive systems in a way that is modular and extensible to a variety of prediction tasks within and beyond recommender systems.

In this study, we concentrated on collaborative filtering (CF) techniques, as they excel in handling the big data with which the web is abundant and tend to outperform content-based filtering techniques. More specifically, we focused on matrix factorization, a state-of-the-art CF technique that builds low-dimensional spaces for hidden features to predict unseen items' ratings and efficiently deals with sparse data. Nevertheless, the lack of transparency significantly reduces user satisfaction and trust in the system. The cold start problem is another issue from which CF techniques suffer.

To tackle these issues, we proposed to use semantic knowledge graphs (KG) that correlate the user with the item's semantic attributes based on the number of interactions between them in the user's history. Item properties are retrieved by SPARQL, the SQL-like semantic web query language, from semantic web databases such as DBpedia. The semantic KGs are used in the latent spaces to build the final model and to generate justifications for the recommendations. They also work as a warm-up solution for the cold start problem.

We proposed four techniques in Chapter 3. The first, an asymmetric semantic explainable ma-

trix factorization user-item-based (ASEMF-UIB) method, consists of a two-phase model-building process; in the first phase, the semantic KGs are incorporated, and, in the second step, the user's history is introduced to the model. The second model is semantic explainable matrix factorization (SemEMF), a one-step formulation in which semantic KGs are integrated into the objective function as a new term. The third model is merged semantic explainable matrix factorization (MergedSemEMF); in this model, semantic KGs and the neighborhood method are both used as soft constraints as regularization in the loss function. For the last model, we proposed the linked data semantic distance matrix factorization (LDSDMF) method, in which two algorithms are integrated into one loss function, resulting in a more robust prediction mechanism.

Finally we proposed the inferred fact style explanation (IFSE) technique. This method incorporates indirect knowledge inferred from the designated semantic KGs to generate explanations for the output by extracting new facts about the user and the new recommended item's semantic attributes in a numerical form.

We conducted an offline evaluation to measure the error rate, recommendability, and the explainability of the recommended items. We also evaluated the explainability of all models, using neighborhood based explainability measures, in two different domains of knowledge, movies and books.

An online evaluation was conducted with a user study of 34 individuals. The results clearly show that the proposed explanation style increased the user perception of system transparency, while being more effective in encouraging the user to accept the recommendation, leading to higher user satisfaction.

Our results have been partially disseminated in [128] and [151] which to be appearing in Proceedings of KDIR 2019.

For future work, we intend to include more semantic attributes in the process of building the explainable recommender model and to experiment with more knowledge and item domains. We also plan to integrate multiple explanation styles, such as NSE, ISE, etc., with the proposed explanation style to increase the transparency of the black box recommender system.

## REFERENCES

- [1] Marco de Gemmis, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. Semantics-aware content-based recommender systems. In *Recommender Systems Handbook*, pages 119–159. Springer US, 2015.
- [2] Alexandre Passant. dbrec: music recommendations using dbpedia. In *Proceedings of the 9th international semantic web conference on The semantic web - Volume Part II*, pages 209–224, Shanghai, China, 2010. Springer-Verlag.
- [3] Svetlin Bostandjiev, John O’Donovan, and Tobias Höllerer. Tasteweights: A visual interactive hybrid recommender system. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys ’12, pages 35–42, New York, NY, USA, 2012. ACM.
- [4] Rose Catherine, Kathryn Mazaitis, Maxine Eskénazi, and William W. Cohen. Explainable entity-based recommendations with knowledge graphs. *CoRR*, abs/1707.05254, 2017.
- [5] F. Yang, N. Liu, S. Wang, and X. Hu. Towards interpretation of recommender systems with sorted explanation paths. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 667–676, Nov 2018.
- [6] Jiawei Hu, Zhiqiang Zhang, Jian Liu, Chuan Shi, Philip S. Yu, and Bai Wang. Recexp: A semantic recommender system with explanation based on heterogeneous information network. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys ’16, pages 401–402, New York, NY, USA, 2016. ACM.
- [7] Cataldo Musto, Fedelucio Narducci, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Linked open data-based explanations for transparent recommender systems. *Inter-*

- national Journal of Human-Computer Studies*, 121:93 – 107, 2019. Advances in Computer-Human Interaction for Recommender Systems.
- [8] Vito Bellini, Angelo Schiavone, Tommaso Di Noia, Azzurra Ragone, and Eugenio Di Sciascio. Knowledge-aware autoencoders for explainable recommender systems. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems, DLRS 2018*, pages 24–31, New York, NY, USA, 2018. ACM.
- [9] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Moviexplain: A recommender system with explanations. In *Proceedings of the Third ACM Conference on Recommender Systems, RecSys '09*, pages 317–320, New York, NY, USA, 2009. ACM.
- [10] Jesse Vig, Shilad Sen, and John Riedl. Tagsplanations: Explaining recommendations using tags. In *Proceedings of the 14th International Conference on Intelligent User Interfaces, IUI '09*, pages 47–56, New York, NY, USA, 2009. ACM.
- [11] Raza Ul Haq. Hybrid recommender system towards user satisfaction. Master’s thesis, University of Ottawa, 2013.
- [12] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*, 11(9), 2018.
- [13] Behnoush Abdollahi and Olfa Nasraoui. Explainable matrix factorization for collaborative filtering. In *Proceedings of the 25th International Conference Companion on World Wide Web*. ACM Press, 2016.
- [14] Behnoush Abdollahi and Olfa Nasraoui. Using explainability for constrained matrix factorization. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 79–83, Como, Italy, 2017. ACM.
- [15] Behnoush Abdollahi. *Accurate and justifiable : new algorithms for explainable recommendations*. PhD thesis, 2017.
- [16] Jose Jesus Castro-Schez, Raul Miguel, David Vallejo, and Lorenzo Manuel López-López. A highly adaptive recommender system based on fuzzy logic for b2c e-commerce portals. *Expert Syst. Appl.*, 38(3):2441–2454, March 2011.

- [17] J. Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM Conference on Electronic Commerce, EC '99*, pages 158–166, New York, NY, USA, 1999. ACM.
- [18] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, January 2003.
- [19] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- [20] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.
- [21] SPARQLsparql query language for rdf. <https://www.w3.org/TR/rdf-sparql-query/>. Accessed: 2018-04-23.
- [22] Pascal Hitzler, Markus Krotzsch, and Sebastian Rudolph. *Foundations of semantic web technologies*. CRC press, 2009.
- [23] Grigoris Antoniou, Paul Groth, Frank van Harmelen, and Rinke Hoekstra. *A Semantic Web Primer*. The MIT Press, 2012.
- [24] G. Stumme, B. Berendt, and A. Hotho. Usage mining for and on the semantic web. In *Proc. NSF Workshop on Next Generation Data Mining*, pages 77–86, Baltimore, November 2002.
- [25] Bettina Berendt, Andreas Hotho, and Gerd Stumme. Towards semantic web mining. In *Proceedings of the First International Semantic Web Conference on The Semantic Web, ISWC '02*, pages 264–278, London, UK, UK, 2002. Springer-Verlag.
- [26] Bettina Berendt, Andreas Hotho, and Gerd Stumme. Semantic web mining and the representation, analysis, and evolution of web space. In *Proceedings of RAWS 2005 Workshop*, pages 1–16, 2005.
- [27] Gerd Stumme, Andreas Hotho, and Bettina Berendt. Semantic web mining: State of the art and future directions. *J. Web Sem.*, 4(2):124–143, 2006.

- [28] R. Cooley, B. Mobasher, and J. Srivastava. Web mining: information and pattern discovery on the world wide web. In *Proceedings Ninth IEEE International Conference on Tools with Artificial Intelligence*, pages 558–567, Nov 1997.
- [29] Myra Spiliopoulou and Lukas C. Faulstich. Wum: A web utilization miner. In *In Proceedings of EDBT Workshop WebDB98*, pages 109–115. Springer Verlag, 1998.
- [30] Olfa Nasraoui, Maha Soliman, Esin Saka, Antonio Badia, and Richard Germain. A web usage mining framework for mining evolving user profiles in dynamic web sites. *IEEE transactions on knowledge and data engineering*, 20(2):202–215, 2008.
- [31] Palakorn Achananuparp, Hyoil Han, Olfa Nasraoui, and Roberta Johnson. Semantically enhanced user modeling. In *Proceedings of the 2007 ACM symposium on Applied computing*, pages 1335–1339, Seoul, Korea, 2007. ACM.
- [32] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.
- [33] Leyla Zhuhadar and Olfa Nasraoui. Semantic information retrieval for personalized e-learning. In *Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on*, volume 1, pages 364–368. IEEE, 2008.
- [34] Miguel-Ángel Sicilia and Miltiadis D. Lytras. The semantic learning organization. *The Learning Organization*, 12(5):402–410, 2005.
- [35] Olfa Nasraoui and Esin Saka. Web usage mining in noisy and ambiguous environments: Exploring the role of concept hierarchies, compression, and robust user profiles. In *Workshop on Web Mining*, pages 82–101. Springer, 2006.
- [36] Anupam Joshi, C. K. Punyapu, and P. Karnam. Personalization & asynchronicity to support mobile web access. In *Workshop on Web Information and Data Management*, 1998.
- [37] Mike Perkowitz and Oren Etzioni. Adaptive web sites: an ai challenge. In *IJCAI*, 1997.
- [38] Osmar R. Zaiane and Jiawei Han. Webml: Querying the world-wide web for resources and



- knowledge. In *Proc. ACM CIKM'98 Workshop on Web Information and Data Management (WIDM'98)*, pages 9–12, 1998.
- [39] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW '94*, pages 175–186, New York, NY, USA, 1994. ACM.
- [40] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, jan 2003.
- [41] Francesco Ricci, Lior Rokach, and Bracha Shapira. Recommender systems: Introduction and challenges. In *Recommender Systems Handbook*, pages 1–34. Springer US, 2015.
- [42] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.
- [43] Francesco Ricci. Recommender systems: Models and techniques. In *Encyclopedia of Social Network Analysis and Mining*, pages 1511–1522. Springer New York, 2014.
- [44] Robin Burke. The adaptive web. chapter Hybrid Web Recommender Systems, pages 377–408. Springer-Verlag, Berlin, Heidelberg, 2007.
- [45] Prem Melville, Raymod J. Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Eighteenth National Conference on Artificial Intelligence*, pages 187–192, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.
- [46] Michael J. Pazzani and Daniel Billsus. The adaptive web. chapter Content-based Recommendation Systems, pages 325–341. Springer-Verlag, Berlin, Heidelberg, 2007.
- [47] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

- [48] Gerald Salton, editor. *Automatic Text Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1988.
- [49] Joseph John Rocchio. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*, pages 313–323, 1971.
- [50] B.V. Dasarathy. *Nearest Neighbor (NN) Norms: Nn Pattern Classification Techniques*. IEEE Computer Society Press tutorial. IEEE Computer Society Press, 1991.
- [51] Stuart E. Middleton, David De Roure, and Nigel R. Shadbolt. Ontology-based recommender systems. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, pages 477–498. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [52] Giovanni Semeraro, Pasquale Lops, Pierpaolo Basile, and Marco de Gemmis. Knowledge infusion into content-based recommender systems. In *Proceedings of the third ACM conference on Recommender systems*, pages 301–304, New York, New York, USA, 2009. ACM.
- [53] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker. Linked open data to support content-based recommender systems. In *Proceedings of the 8th International Conference on Semantic Systems*, pages 1–8, Graz, Austria, 2012. ACM.
- [54] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244, 1990.
- [55] Warren Weaver. Translation. In William N. Locke and A. Donald Boothe, editors, *Machine Translation of Languages*, pages 15–23. MIT Press, Cambridge, MA, 1949/1955. Reprinted from a memorandum written by Weaver in 1949.
- [56] Marco Degemmis, Pasquale Lops, and Giovanni Semeraro. A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction*, 17(3):217–255, 2007.

- [57] Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, and Pierpaolo Basile. Integrating tags in a semantic content-based recommender. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 163–170, Lausanne, Switzerland, 2008. ACM.
- [58] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):34–43, 2001.
- [59] Dean Jones, Trevor Bench-Capon, and Pepijn Visser. Methodologies for ontology development. 1998.
- [60] Randall Davis and Douglas B. Lenat. *Knowledge-Based Systems in Artificial Intelligence: 2 Case Studies*. McGraw-Hill, Inc., 1982.
- [61] Somnath Banerjee, Krishnan Ramanathan, and Ajay Gupta. Clustering short texts using wikipedia. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 787–788, Amsterdam, The Netherlands, 2007. ACM.
- [62] Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242, Lisbon, Portugal, 2007. ACM.
- [63] Paolo Ferragina and Ugo Scaiella. Fast and accurate annotation of short texts with wikipedia pages. *IEEE Software*, 29(1):70–75, jan 2012.
- [64] Evgeniy Gabilovich and Shaul Markovitch. Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*, 34:443–498, 2009.
- [65] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, sep 1990.
- [66] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell.*, 193:217–250, 2012.

- [67] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Soren, ren Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semant.*, 7(3):154–165, 2009.
- [68] Alexandre Passant and Stefan Decker. Hey! ho! let’s go! explanatory music recommendations with dbrec. In Lora Aroyo, Grigoris Antoniou, Eero Hyvönen, Annette ten Teije, Heiner Stuckenschmidt, Liliana Cabral, and Tania Tudorache, editors, *The Semantic Web: Research and Applications: 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30 – June 3, 2010, Proceedings, Part II*, pages 411–415. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [69] Alexandre Passant. Measuring semantic distance on linking data and using it for resources recommendations. In *AAAI spring symposium: linked data meets artificial intelligence*, volume 77, page 123, 2010.
- [70] Benjamin Heitmann and Conor Hayes. Using linked data to build open, collaborative recommender systems. In *AAAI spring symposium: linked data meets artificial intelligence*, volume 2010, 2010.
- [71] Yves Raimond, Mark B Sandler, and Q Mary. A web of musical information. In *ISMIR*, pages 263–268, 2008.
- [72] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, Vancouver, Canada, 2008. ACM.
- [73] Oktie Hassanzadeh and Mariano Consens. Linked movie data base. *Linked Data on the Web (LDOW2009)*, 16, 2009.
- [74] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244, 2014.

- [75] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8. ACM, 2011.
- [76] Giuseppe Rizzo and Raphaël Troncy. Nerd: a framework for unifying named entity recognition and disambiguation extraction tools. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 73–76. Association for Computational Linguistics, 2012.
- [77] Herbert Rubenstein and John B. Goodenough. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633, 1965.
- [78] Z. S. Harris. *Mathematical structures of language*. Interscience Publishers, 1968.
- [79] Peter D Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188, 2010.
- [80] Saif Mohammad and Graeme Hirst. Distributional measures as proxies for semantic distance: A survey. *Computational Linguistics*, 1(1), 2006.
- [81] Will Lowe. Towards a theory of semantic space. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 23, 2001.
- [82] Peter W Foltz and Susan T Dumais. Personalized information delivery: An analysis of information filtering methods. *Communications of the ACM*, 35(12):51–60, 1992.
- [83] Frank McCarey, Mel O Cinneide, and Nicholas Kushmerick. Recommending library methods: An evaluation of the vector space model (vsm) and latent semantic indexing (lsi). In *International Conference on Software Reuse*, pages 217–230. Springer, 2006.
- [84] Panagiotis Symeonidis. Content-based dimensionality reduction for recommender systems. In *Data Analysis, Machine Learning and Applications*, pages 619–626. Springer Berlin Heidelberg, 2008.
- [85] Maria Terzi, Maria-Angela Ferrario, and Jon Whittle. Free text in user reviews: Their role in

- recommender systems. In *Workshop on Recommender Systems and the Social Web at the 5th ACM International Conference on Recommender Systems (RecSys' 11)*, pages 45–48, 2011.
- [86] Mark D Dunlop. The effect of accessing nonmatching documents on relevance feedback. *ACM Transactions on Information Systems (TOIS)*, 15(2):137–153, 1997.
- [87] Dominic Widdows. Orthogonal negation in vector spaces for modelling word-meanings and document retrieval. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 136–143. Association for Computational Linguistics, 2003.
- [88] Yehuda Koren and Robert Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*, pages 77–118. Springer US, 2015.
- [89] Douglas W Oard, Jinmook Kim, et al. Implicit feedback for recommender systems. In *Proceedings of the AAAI workshop on recommender systems*, volume 83. WoUongong, 1998.
- [90] Xia Ning, Christian Desrosiers, and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender Systems Handbook*, pages 37–76. Springer US, 2015.
- [91] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, UAI'98*, pages 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
- [92] Mukund Deshpande and George Karypis. Item-based top-*n* recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, 2004.
- [93] Simon Funk. Netflix update: Try this at home. Technical report, 2006.
- [94] Yue Shi, Martha Larson, and Alan Hanjalic. Mining contextual movie similarity with matrix factorization for context-aware recommendation. *ACM Trans. Intell. Syst. Technol.*, 4(1):16:1–16:19, February 2013.

- [95] Nidhi Kushwaha, Shubham Mehrotra, Ronish Kalia, Dhruv Kumar, and O. P. Vyas. Inclusion of semantic and time-variant information using matrix factorization approach for implicit rating of last.fm dataset. *Arabian Journal for Science and Engineering*, 41(12):5077–5092, Dec 2016.
- [96] Jaafar BenAbdallah, Juan C. Caicedo, Fabio A. Gonzalez, and Olfa Nasraoui. Multimodal image annotation using non-negative matrix factorization. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01, WI-IAT '10*, pages 128–135, Washington, DC, USA, 2010. IEEE Computer Society.
- [97] Behnoush Abdollahi and Olfa Nasraoui. A cross-modal warm-up solution for the cold-start problem in collaborative filtering recommender systems. In *Proceedings of the 2014 ACM Conference on Web Science, WebSci '14*, pages 257–258, New York, NY, USA, 2014. ACM.
- [98] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250, Philadelphia, Pennsylvania, USA, 2000. ACM.
- [99] Nava Tintarev and Judith Masthoff. Explaining recommendations: Design and evaluation. In *Recommender Systems Handbook*, pages 353–382. Springer US, 2015.
- [100] A. Lopez-Suarez and M. Kamel. Dykor: a method for generating the content of explanations in knowledge systems. *Know.-Based Syst.*, 7(3):177–188, 1994.
- [101] Mustafa Bilgic and Raymond J. Mooney. Explaining recommendations: Satisfaction vs. promotion. In *Beyond Personalization Workshop, IUI*, volume 5, page 153, 2005.
- [102] Jordan Barria-Pineda and Peter Brusilovsky. Making educational recommendations transparent through a fine-grained open learner model. In *IUI Workshops*, 2019.
- [103] Pigi Kouki, James Schaffer, Jay Pujara, John O'Donovan, and Lise Getoor. Personalized explanations for hybrid recommender systems. In *Proceedings of the 24th International Conference on Intelligent User Interfaces - IUI '19*. ACM Press, 2019.

- [104] Masahiro Sato, Budrul Ahsan, Koki Nagatani, Takashi Sonoda, Qian Zhang, and Tomoko Ohkuma. Explaining recommendations using contexts. In *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval - IUI '18*. ACM Press, 2018.
- [105] Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth. Thinking positively-explanatory feedback for conversational recommender systems. In *Proceedings of the European Conference on Case-Based Reasoning (ECCBR-04) Explanation Workshop*, pages 115–124, 2004.
- [106] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit 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, SIGIR '14*, pages 83–92, New York, NY, USA, 2014. ACM.
- [107] Raymond J. Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries, DL '00*, pages 195–204, New York, NY, USA, 2000. ACM.
- [108] Nava Tintarev and Judith Masthoff. Evaluating the effectiveness of explanations for recommender systems. *User Modeling and User-Adapted Interaction*, 22(4-5):399–439, October 2012.
- [109] Liliana Ardissono, Anna Goy, Giovanna Petrone, Marino Segnan, and Pietro Torasso. Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices. *Applied Artificial Intelligence*, 17:687–714, 2003.
- [110] Daniel Billsus and Michael J. Pazzani. A personal news agent that talks, learns and explains. In *Proceedings of the Third Annual Conference on Autonomous Agents, AGENTS '99*, pages 268–275, New York, NY, USA, 1999. ACM.
- [111] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Justified recommendations based on content and rating data. 2008.
- [112] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker. Linked open data to support content-based recommender systems. In *Proceedings of*



*the 8th International Conference on Semantic Systems, I-SEMANTICS '12*, pages 1–8, New York, NY, USA, 2012. ACM.

- [113] Chuan Shi, Zhiqiang Zhang, Yugang Ji, Weipeng Wang, Philip S. Yu, and Zhiping Shi. Sem-rec: a personalized semantic recommendation method based on weighted heterogeneous information networks. *World Wide Web*, 22:153–184, 2018.
- [114] Xiang Wang, Xiangnan He, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. Tem: Tree-enhanced embedding model for explainable recommendation. In *Proceedings of the 2018 World Wide Web Conference, WWW '18*, pages 1543–1552, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.
- [115] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18*, pages 417–426, New York, NY, USA, 2018. ACM.
- [116] Roberto Mirizzi, Tommaso Di Noia, Azzurra Ragone, Vito Claudio Ostuni, and Eugenio Di Sciascio. Movie recommendation with dbpedia. In *IIR*, 2012.
- [117] Carlos Luis Sanchez Bocanegra, Jose Luis Sevillano Ramos, Carlos Rizo, Anton Civit, and Luis Fernandez-Luque. Healthrecsys: A semantic content-based recommender system to complement health videos. *BMC Medical Informatics and Decision Making*, 17(1):63, May 2017.
- [118] Chuan Shi, Chong Zhou, Xiangnan Kong, Philip S. Yu, Gang Liu, and Bai Wang. Heterecom: A semantic-based recommendation system in heterogeneous networks. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12*, pages 1552–1555, New York, NY, USA, 2012. ACM.
- [119] B. Alhijawi, N. Obeid, A. Awajan, and S. Tedmori. Improving collaborative filtering recommender systems using semantic information. In *2018 9th International Conference on Information and Communication Systems (ICICS)*, pages 127–132, April 2018.

- [120] Xiao Yu, Xiang Ren, Quanquan Gu, Yizhou Sun, and Jiawei Han. Collaborative filtering with entity similarity regularization in heterogeneous information networks. 2013.
- [121] Sultan Alfarhood. *Exploiting Semantic Distance in Linked Open Data for Recommendation*. PhD thesis, University of Arkansas, 2017.
- [122] Walter Carrer-Neto, María Luisa Hernández-Alcaraz, Rafael Valencia-García, and Francisco García-Sánchez. Social knowledge-based recommender system. application to the movies domain. *Expert Syst. Appl.*, 39(12):10990–11000, September 2012.
- [123] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM '11*, pages 287–296, New York, NY, USA, 2011. ACM.
- [124] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems, RecSys '16*, pages 233–240, New York, NY, USA, 2016. ACM.
- [125] Mao Ye, Zhi Tang, Jianbo Xu, and Lifeng Jin. Recommender system for e-learning based on semantic relatedness of concepts. *Information*, 6(3):443–453, 2015.
- [126] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems, NIPS'07*, pages 1257–1264, USA, 2007. Curran Associates Inc.
- [127] Behnoush Abdollahi and Olfa Nasraoui. Explainable restricted boltzmann machines for collaborative filtering. *presented at 2016 ICML Workshop on Human Interpretability in Machine Learning (WHI 2016), New York, NY, arXiv preprint arXiv:1606.07129*, 2016.
- [128] Mohammed Alshammari, Olfa Nasraoui, and Behnoush Abdollahi. A semantically aware explainable recommender system using asymmetric matrix factorization. In *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*. SCITEPRESS - Science and Technology Publications, 2018.

- [129] Khalil Damac and Olfa Nasraoui. Seer: An explainable deep learning midi-based hybrid song recommender system. *arXiv Archive*, 2019.
- [130] Stuart E. Middleton, Harith Alani, and David De Roure. Exploiting synergy between ontologies and recommender systems. *CoRR*, cs.LG/0204012, 2002.
- [131] William Yang Wang, Kathryn Mazaitis, and William W. Cohen. Programming with personalized pagerank: A locally groundable first-order probabilistic logic. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management, CIKM '13*, pages 2129–2138, New York, NY, USA, 2013. ACM.
- [132] Yizhou Sun, Jiawei Han, Peixiang Zhao, Zhijun Yin, Hong Cheng, and Tianyi Wu. Rankclus: Integrating clustering with ranking for heterogeneous information network analysis. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, EDBT '09*, pages 565–576, New York, NY, USA, 2009. ACM.
- [133] Robert W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345–, June 1962.
- [134] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '99*, pages 230–237, New York, NY, USA, 1999. ACM.
- [135] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, Hong Kong, Hong Kong, 2001. ACM.
- [136] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2009.
- [137] James P. McGlothlin and Latifur R. Khan. Materializing and persisting inferred and uncertain knowledge in RDF datasets. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, 2010.

- [138] Miriam Martínez-García, Aida Valls, and Antonio Moreno. Inferring preferences in ontology-based recommender systems using wowa. *Journal of Intelligent Information Systems*, Oct 2018.
- [139] Vicen Torra. The weighted OWA operator. *International Journal of Intelligent Systems*, 12(2):153–166, feb 1997.
- [140] Daniel Lemire, Harold Boley, Sean McGrath, and Marcel Ball. Collaborative filtering and inference rules for context-aware learning object recommendation. *Interactive Technology and Smart Education*, 2(3):179–188, aug 2005.
- [141] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, jan 2016.
- [142] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a web of open data. In *The Semantic Web*, pages 722–735. Springer Berlin Heidelberg, 2007.
- [143] Denny Vrandečić. Wikidata: a new platform for collaborative data collection. In *Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion*. ACM Press, 2012.
- [144] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: Core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web - WWW '07*. ACM Press, 2007.
- [145] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, Jr., and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI'10*, pages 1306–1313. AAAI Press, 2010.
- [146] A. Singhal. Introducing the knowledge graph: things, not strings. Technical report, Google, 2012.

- [147] Jiwei Zhong, Haiping Zhu, Jianming Li, and Yong Yu. Conceptual graph matching for semantic search. In Uta Priss, Dan Corbett, and Galia Angelova, editors, *Conceptual Structures: Integration and Interfaces*, pages 92–106, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
- [148] Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop, ICDEW '07*, pages 801–810, Washington, DC, USA, 2007. IEEE Computer Society.
- [149] J. Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 1988.
- [150] Lee J. Cronbach. Coefficient alpha and the internal structure of tests. *Psychometrika*, 16(3):297–334, Sep 1951.
- [151] Mohammed Alshammari and Olfa Nasraoui. Augmented semantic explanations for collaborative filtering recommendations. In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*. SCITEPRESS - Science and Technology Publications, 2019.

## CURRICULUM VITAE

- NAME:** Mohammed Sanad Alshammari
- ADDRESS:** Computer Engineering & Computer Science Department  
J.B Speed School of Engineering  
University of Louisville  
Louisville, KY 40292  
United States of America.
- EDUCATION:** Ph.D., Computer Science & Engineering, August 2019  
**University of Louisville**, *Louisville, KY, USA.*  
M.Sc. with distinction in Advanced Computer Science, November 2011.  
**University of Leicester**, *Leicester, United Kingdom.*  
Bachelor of Education in Computer with GPA 4.05 out of 5.00, June 2008.  
**University of Ha'il**, *Ha'il, Kingdom of Saudi Arabia.*
- WORK EXPERIENCE:** Computer Teacher, Public high school, Ha'il, Saudi Arabia, 2008.  
Activities Coordinator, Sahma Schools, Ha'il, Saudi Arabia, 2012.  
Lecturer, Northern Borders University, Rafha, Saudi Arabia, 2012-2014.  
Member of Quality and Accreditation Committee, NBU, SA, 2012-2014.  
Finance Director, Saudi Students Club, University of Louisville, 2017.  
President, Saudi Students Club, University of Louisville, 2018.

**AWARDS:** Academic excellence award from Saudi Arabia Cultural Bureau in London, UK, 2010.  
Academic excellence award from Saudi Arabia Cultural Bureau in London, UK, 2011.

**LANGUAGES:** Arabic, mother tongue.  
English.

**PUBLICATIONS:**

1. Mohammed Alshammari, Olfa Nasraoui, and Behnoush Abdollahi. A semantically aware explainable recommender system using asymmetric matrix factorization. In Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management. SCITEPRESS - Science and Technology Publications, 2018.
2. Mohammed Alshammari and Olfa Nasraoui. Augmented Semantic Explanations for Collaborative Filtering Recommendations. In Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management. SCITEPRESS - Science and Technology Publications, 2019.
3. Mohammed Alshammari, Olfa Nasraoui, and Scott Sanders. Mining Semantic Knowledge Graphs to Add Explainability to Black Box Recommender Systems. *in IEEE Access*, 2019.
4. Mohammed Alshammari. A Web based Interface for Recording Ontologies with Deductive Reasoning Support, Masters Thesis, University of Leicester, 2011.
5. Mohammed Alshammari and Olfa Nasraoui, Humanizing Big Data Predictors: An Explainable Recommender System using Semantically-Aware Matrix Factorization, A poster presented at the Speed Research Exposition, Spring 2018.
6. Mohammed Alshammari and Olfa Nasraoui, Building Semantically Intelligent Explanations into a Black Box Recommender System, A poster presented at the the Graduate Student Regional Research Conference, Spring 2019.
7. Mohammed Alshammari, Building Semantically Intelligent Explanations into a Black Box Recommender System, A seminar talk in the Computer Engineering & Computer Science department weekly seminar in Spring 2019.