A HYPERGRAPH BASED FRAMEWORK FOR REPRESENTING AGGREGATED USER PROFILES, EMPLOYING IT FOR A RECOMMENDER SYSTEM AND PERSONALIZED SEARCH THROUGH A HYPERNETWORK METHOD

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HILAL TARAKCI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

JUNE 2017

Approval of the thesis:

A HYPERGRAPH BASED FRAMEWORK FOR REPRESENTING AGGREGATED USER PROFILES, EMPLOYING IT FOR A RECOMMENDER SYSTEM AND PERSONALIZED SEARCH THROUGH A HYPERNETWORK METHOD

submitted by **HILAL TARAKCI** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering**

Assoc. Prof. Dr. Murat Manguoğlu
Supervisor, **Computer Engineering Department, METU**

Prof. Dr. Nihan Kesim Çiçekli
Co-supervisor, **Computer Engineering Department, METU**

Examining Committee Members:

Prof. Dr. Özgür Ulusoy
Computer Engineering Department, Bilkent University

Assoc. Prof. Dr. Murat Manguoğlu
Computer Engineering Department, METU

Prof. Dr. Ahmet Coşar
Computer Engineering Department, METU

Assoc. Prof. Dr. Pınar Karagöz
Computer Engineering Department, METU

Assist. Prof. Dr. Gönenç Ercan
Institute of Informatics, Hacettepe University

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: HILAL TARAKCI

Signature :

ABSTRACT

A HYPERGRAPH BASED FRAMEWORK FOR REPRESENTING AGGREGATED USER PROFILES, EMPLOYING IT FOR A RECOMMENDER SYSTEM AND PERSONALIZED SEARCH THROUGH A HYPERNETWORK METHOD

Tarakci, Hilal
Ph.D., Department of Computer Engineering
Supervisor : Assoc. Prof. Dr. Murat Manguoğlu
Co-Supervisor : Prof. Dr. Nihan Kesim Çiçekli

June 2017, 131 pages

In this thesis, we present a hypergraph based user modeling framework to aggregate partial profiles of the individual and obtain a complete, semantically enriched, multi-domain user model. We also show that the constructed user model can be used to support different personalization services including recommendation. We evaluated the user model against datasets consisting of user's social accounts including Facebook, Twitter, LinkedIn and Stack Overflow. The evaluation results confirmed that the proposed user model improves the quality of the constructed user model in every case. The results also showed that the improvement is higher for generic domain datasets than datasets representing the user in terms of one domain. We propose a recommender system which exploits the proposed framework as case study. The presented system is capable of displaying semantic user model, making domain based, cross domain and general recommendations, discovery of similar users, discovery of users that might be interested in a given item and computation of a user's interest on a given item. We also show that the proposed framework is extendible by extending the framework by adding context information.

We also present another user modeling approach based on hypernetworks. The methodology is based on modelling the individual as hypernetwork with a multi-level approach. Initially, lower level terms are represented with hyperedges. Afterwards, higher level terms are modeled by reusing lower level hyperedges. Hypernetwork is clustered to obtain a dynamically tailored user profile. Basically, tailoring a user profile is achieved by filtering the clusters which we want to focus on. Other clusters are eliminated. Q-Analysis technique is used to cluster the hypernetwork. The technique clusters the hypernetwork at level q by listing hyperedges which share q vertices. Eccentricity is a metric which indicates the amount of new and unshared vertices introduced by a hyperedge. We optimize clustering algorithm by using eccentricity of clusters. We define an eccentricity threshold by trial and error. When there exist clusters which have eccentricity at least equal to this threshold, clustering iterations are terminated. The methodology is evaluated against one month long Yandex search logs which contain over 167 million records and slightly improved Yandex's non-personalized ranking which is already a well performing baseline.

Keywords: User Modeling, User Profile, Hypergraph Based User Model, Graph Traversal, Knowledge Representation, Recommender System

ÖZ

BİRLEŞTİRİLMİŞ KULLANICI PROFİLLERİ İÇİN HİPERÇİZGE-TABANLI BİR ÇATI, BU ÇATININ BİR ÖNERİ SİSTEMİNDE KULLANIMI VE BİR HİPERÇİZGE AĞ METODU İLE KİŞİLEŞTİRİLMİŞ ARAMA

Tarakci, Hilal
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Murat Manguoğlu
Ortak Tez Yöneticisi : Prof. Dr. Nihan Kesim Çiçekli

Haziran 2017 , 131 sayfa

Bu tezde, kişinin kısmi profillerini eksiksiz, anlamsal açıdan zenginleştirilmis, çoklu alanlı bir kullanıcı modeli elde etmek amacıyla birleştirmek için hyperçizge tabanlı bir kullanıcı modelleme çerçevesini sunuyoruz. Ayrıca, oluşturulan kullanıcı modelinin öneri sistemleri dahil değişik kişiselleştirme servislerini destekleyebileceğini gosteriyoruz. Kullanıcı modelini kullanıcının Facebook, Twitter, LinkedIn ve StackOverflow sosyal hesaplarından oluşturulmuş bir veri kümesine karşı değerlendirdik. Değerlendirme sonuçları, öne sürülen kullanıcı modelinin her durumda oluşturulan kullanıcı modeli kalitesini iyileştirdiğini doğruladı. Sonuçlar ayrıca iyileştirmenin genel veri kümelerinde, belli bir alana ait özel veri kümelerine göre daha yüksek olduğunu gösterdi. Örnek çalışma olarak, öne sürülen çerçeveyi kullanan bir öneri sistemi sunuyoruz. Sunulan sistem kullanıcının anlamsal profilini gösterebilir, alan tabanlı, alanlar arası ya da genel önerilerde bulunabilir, benzer kullanıcıları keşfedebilir, verilen bir objeye ilgi duyabilecek kullanıcıları keşfedebilir ve bir kullanıcının bir objeye olan ilgisini hesaplayabilir. Ayrıca bağlam bilgisi ile genişleterek, sunulan çerçevenin genişletilebilir olduğunu da gösteriyoruz.

Ayrıca hiperağ tabanlı başka bir kullanıcı modelleme yaklaşımı da sunuyoruz. Yaklaşım, kişiyi çoklu-seviyeli bir yolla modellemeye dayanmaktadır.Önce alt seviye terimler ifade edilir. Sonrasında, daha üst seviye terimler, daha önce ifade edilmiş alt terimler yeniden kullanılarak modellenir. Hiperağ dinamik olarak uyarlanmış bir kullanıcı modeli elde edilmek amacıyla kümelenir. Temel olarak, uyarlanmış bir kullanıcı modeli elde edilmesi, odaklanmak istediğimiz kümeleri seçilmesiyle başarılır. Diğer kümeler elenir. Hiper-ağı kümelemek için Q-Analiz tekniği kullanılır. Teknik, q seviyesinde, q adet düğüm paylaşan hiperkenarları aynı kümede toplar. Egzantriklik, bir hiperkenarın sunduğu yeni ve paylaşılmayan düğümlerin miktarını ifade eden bir metriktir. Kümeleme algoritmasını, kümelerin egzantrikliğini kullanarak optimize ediyoruz. Deneme yanılma yöntemi ile bir egzantriklik eşiği tanımlıyoruz. Belirlenen bu egzantriklik eşiğine eşit veya daha yüksek egzantrikliğe sahip kümeler oluşmuş ise, kümeleme döngüsünü sonlandırıyoruz. Bu metod, 167 milyondan fazla kayıt içeren bir aylık uzun Yandex arama logları üzerinde denenmiştir ve çok iyi sonuç veren Yandex'in kişiselleştirilmemiş sıralama algoritmasını biraz iyileştirmiştir.

Anahtar Kelimeler: Kullanıcı Modelleme, Kullanıcı Profili, Hiperçizge-Tabanlı Kullanıcı Modeli, Çizge Gezintisi, Bilgi Reprezantasyonu, Öneri Sistemi

To My Beloved Father..

ACKNOWLEDGMENTS

This has been a very long journey for me. I met lots of great people, learned from them, get more experienced along the way. I am glad i did this, because it was more than a study. It was an experience of a life time. It was difficult, required a lot of patience and i am glad i am where i am now. I would like to express my gratitude to everyone who helped me during this journey.

First of all, I would like to thank my supervisor(my co-supervisor now since she is on Sabbatical at Syracuse University) Prof. Nihan Kesim Çiçekli for her brilliant support and incredible guidance throughout this study. She always trusted me and showed me the direction when I felt lost inside the study. Most importantly, she became my role model as I witnessed her strong, bright and sweet personality.

I would like to thank Assoc. Prof. Murat Manguoğlu for accepting me as his student, when i needed a supervisor. I also want to express my gratitude to Prof. Özgür Ulusoy, Prof. Ahmet Coşar, Prof. Ferda Nur Alpaslan and Assoc. Prof. Pınar Karagöz Şenkul for their guidance during my thesis committees. Their comments and guidance helped me to put my study in a better shape. Besides, they were always friendly to me and it has been always a pleasure for me to attend thesis committees with them. I will miss these committee days.

I also want to thank Prof. Halit Oğuztüzün, Assoc. Prof. Gönenç Ercan, Assoc. Prof. Tolga Can and Assoc. Prof. Çiğdem Turhan for being members in my thesis defense committee.

I am grateful to Özgür Kaya and his lab for their technical support during the online demo of the thesis study.

I thank my friends for long discussions during narrowing down my thesis topic. They informed me about the process of writing a dissertation and they warned me about the ups and downs through this long journey. Most important of all, they inspired me with their accomplishments, personalities and advice. I want to thank to my other friends for their understanding, support and for believing in me during this process.

I would like to thank my bosses Prof. Muzaffer Elmas, Prof. Ümit Kocabıçak and Evrim Erdoğuş for making my life easier while i am struggling setting up a balance between my academic studies and enterprise work. I worked with them in different times, and they have always been very understanding. I also thank to my colleagues for their feedback and comments on my study.

This work is partially supported by The Scientific and Technical Council of Turkey Grant “TUBITAK EEEAG-112E111”. Thanks to the institution for their support.

Last but not the least, I want to thank my lovely family for their continuous support and assistance throughout this study.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

LIST OF ABBREVIATIONS

TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
FOAF	Friend of a Friend
SIOC	Semantically-Interlinked Online Communities
MOAT	Meaning of a Tag
GUMO	The General User Model Ontology
API	Application Programming Interface
MQL	Metaweb Query Language
ODP	Ontology Design Patterns
JSON	JavaScript Object Notation

CHAPTER 1

INTRODUCTION

1.1 Motivation

1.1.1 Why we need personalization?

Today, we live in the digital age and are exposed to information overload as the amount of data expands exponentially. In the past, majority of data was coming from enterprise systems and was structured. However, today's data mainly comes from social sources including social web sites, blogs, chat rooms, product review sites, communities, web pages, emails etc. and it is unstructured [36]. In addition, smart phone and social network usage trend will continue to contribute to the dramatic data growth in the foreseeable future [77].

A web site [1] keeps track of the data produced by several social web sites in real time. In 10 minutes, $3.4M$ tweets were tweeted in Twitter [2], $1.2K$ hours of video was uploaded and $1.4M$ hours of video was watched in YouTube [3], $33M$ posts were shared and $31M$ items were liked in Facebook [4], $2T$ emails were sent, $31K$ items were purchased in Amazon [5] and $7M$ files were saved in Dropbox [6]. During 10 minutes, 14 million GBs of data was transferred over the internet. This means that current average data growth rate is 23 thousands GBs per second and 2000 million GBs in 24 hours. Since data growth is exponential, this value is going to get much

[1] The Internet in Real Time, http://pennystocks.la/internet-in-real-time/

[2] Twitter, https://twitter.com/

[3] YouTube, https://www.youtube.com/

[4] Facebook, https://www.facebook.com/

[5] Amazon, http://www.amazon.com/

[6] Dropbox, https://www.dropbox.com/

bigger every day.

The huge amount of data requires smart search algorithms, effective information extraction and useful personalization techniques. By definition, personalization is adapting the functionality of a system or service to a particular individual. To increase the relevance of the search results, Google applies personalized search by examining the individual's previous searches and web history since 2009 [7]. Amazon uses personalization to provide the most relevant recommendations to the users. Personalization is very crucial for online advertising, since the aim is to show the user the most relevant advertisements. The key to successful personalization is to extract a complete and structured profile of the individual.

The exponentially increasing amount of content also makes the requirement for personalization services inevitable. Personalization services are several utilities which help users to manage the content according to their needs and areas of interest. To support these services, users' profiles should be constructed and stored in a model which can be employed by different personalization services effectively.

Personalization services differ in terms of their domain of interest. For instance, a book recommender focuses on books that might be interesting to an individual and a health monitoring application focuses on the nutrition habits of the user. Besides, most of the personalized services are designed to operate on different environments including mobile devices.

Our first goal is to construct a holistic user profile which models the user from different perspectives by aggregating several partial distributed profiles of the user. Our second goal is to provide these services the most relevant information about the individual regarding the service's context. In other words, our usage scenario is as follows: A personalized service provides its purpose and a test query (if applicable) as context and requests a tailored user model for provided test query.

[7] Google Patent, System and method for personalized search, http://www.google.com/patents/US20140129539

1.1.2 How to extract user profiles?

The easiest way to construct a user profile is by asking the user himself/herself. However, this is a cumbersome task and obtaining a complete profile and maintaining it by this methodology is practically impossible. Alternate approaches to build a user profile are based on using the data which is already available to extract relevant information about the user.

This century is going to be defined by the ability to monitor people by the data they produce or share [79], since we live in a *data driven society*. With the advent of Web 2.0, users are allowed to actively participate in the web by creating content and interacting with each other by means of social networking and tagging platforms [102]. Thus, the social web structures which link people to several concepts and to other users have emerged. The large scale data created in Web 2.0 reflects the interests and preferences about the content contributors and is an invaluable data source for personalization purposes.

The goal of Web 3.0 [67] is to close the gap between reality and virtual world by personalizing the web. In order to achieve this goal, Web 3.0 focuses on the individuals and supports pervasive and ubiquitous computing. Ubiquitous applications should be capable of running on different devices and should be aware of the preferences of the individual and the context. Personalization services are several utilities which help the user to manage the content according to his/her needs and areas of interest. To support these services, users' profile should be constructed and stored in a model which can be employed by personalization services effectively.

As stated above, the habit of using social networks spreads exponentially in recent years. People tend to use different social web sites for distinct purposes [5]. For instance, Facebook is used for entertainment and personal activities, LinkedIn [8] is exploited to expose professional skills, Twitter is employed to share ideas and follow friends or influencers and Stack Overflow [9] is used to post questions in computer science domain.

[8] LinkedIn, www.linkedin.com

[9] Stack Overflow, www.stackoverflow.com

The user's activities on social websites reveal important information about his/her profile. The individual's fields of interest can be exposed by mining these social accounts. Therefore, mining separate social networks independently results in partial profiles of the user which merely represent user's preferences for one or few domains depending on the usage purpose of the social web site. On the other hand, aggregating partial profiles for several social web accounts results in a multi-domain, holistic profile of the individual.

The user model should be capable of representing the narration about the individual correctly. Narrations consist of statements describing the user. If a statement relates two entities, it is modeled as a binary relation. *User u likes movie m* is an example binary relation which relates user u and movie m. A statement which relates three entities is a 3-ary relation. *User u likes watching movies in rainy days* is a 3-ary relation which relates user u with activity *watching movies* and context *rainy days*. In general, statements express n-ary relations between entities. In certain n-ary relations, order of entities is important. For instance, in a statement which provides the recipe to bake cookies, the order of steps is important. Therefore, an efficient user model should represent n-ary relations and preserve order of entities in these relations.

1.1.3 How to model users and why?

The user profile construction process is defined in three steps: collection of data from knowledge sources such as social media websites or personal devices, construction of the profile by extracting user's fields of interest and the consumption of the user model by personalization based applications [1]. There is a considerable amount of work on extracting user profiles from social websites [1].

Representing a user profile with a vector of terms is a common strategy. The terms in the vector could be words or concepts extracted from the user's texts. The terms could be assigned weights which are calculated by using a weighting function. The weighting scheme could be term frequency (TF), term frequency-inverse document frequency (TF-IDF) [74] or a user-defined algorithm.

The employed user profile structure is mutually associated with the aggregation methodology. The aggregation process depends on the predefined user model data structure, and this structure is defined according to the main goals of the aggregation. If the main purpose is producing an interoperable user model, the profile is generally defined by a standard [85] or user-defined [129, 50] ontology. There are also predictive statistical user models which employ machine learning approaches [135, 25, 28, 61]. Statistical approaches require large amounts of user information.

User modeling domain basically consists of the users, the items and relationships betwen these objects. This structure constitutes a connected data environment. In a connected data environment, most of the queries are solved by introducing a navigation algorithm in the connected data structure. *Connected data problems* are queries that can be solved by defining structure traversal algorithms. In this thesis, one of our main goals is solving connected data problems such as recommendation effortlessly. An effective solution strategy for connected data problems is matching an entrance point to the data structure and traversing the neighbours according to the specified algorithm. Therefore, graphs naturally support connected data problems [92]. The vertices usually represent the items and the users where an edge between a user and an item indicate user's interest on that item. The edges could be associated with weights which represent the strength of the relation between the vertices.

Since the graph is only capable of representing binary relations, other approaches have been proposed for handling higher order relations in user modeling domain. There are a few studies which define user model as bipartite [121] and tripartite graphs [31]. In bipartite graphs, vertices can be grouped in two disjoint sets. For instance, a simple user model which only focuses on movie domain and relate users with movies might be modeled with bipartite graph, since there are two vertex types user and movie and all relations are between users and movies. Similarly, in a tripartite graph, vertices form three disjoint sets and relations are binary, between different sets of vertices. A sample user model which models music listening habits in the format *User u likes to listen song s and song s is from album a* can be modeled with tripartite graph, since it has three vertex types user, song and album and all relations are binary. In general, if the number of vertex types *n* is known in advance and the relations in the user model are binary, an *n*-partite graph is capable of representing the profile. However, if there are higher-order relations, a hypergraph is more appropriate to represent the user model [68, 65, 23].

Theoretically, a hyperedge is a *set* of arbitrary vertices. In sets, the order of elements is irrelevant. For instance, sets $\{a, b, c\}$, $\{a, c, b\}$, $\{b, a, c\}$, $\{b, c, a\}$, $\{c, a, b\}$ and $\{c, b, a\}$ correspond to the same hyperedge. The order of elements are important for certain relations. In such cases, not keeping order might result in ambiguity.

Simplicial complexes represent geometric realizations of elements in a set. In other words, they introduce topology of entities when a statement is represented using a simplicial complex. A hypernetwork connects vertices basically using simplicial complexes instead of sets. Therefore, hypernetworks are capable of representing n-ary relations by preserving order. Using hypernetworks is a brand new approach in user modelling domain [132, 109]. Q-analysis [60] is a technique which provides a hierarchical listing of connected hyperedges by inspecting their topology. Eccentricity [53] is a metric which is used to decide which hyperedge provides more information, namely more *eccentric*.

1.1.4 What we present in this thesis?

Seamless aggregation of partial user profiles obtained from different knowledge sources is still an unsolved problem. In this thesis, we present a hypergraph based user modeling framework to aggregate partial profiles of the individual to obtain a complete, semantically enriched, multi-domain user model and show that it can be used to support different personalization services including recommendation.

In this thesis, we also introduce another approach to construct a multi-level user model using hypernetworks. We aim our proposed user model to be consumed by personalized services. Therefore, we provide a dynamic tailoring feature which filters only the most related parts of the user model based on requester personalized service context, so that requester personalized service can apply heuristics to the tailored user model instead of the entire profile. We use Q-analysis and eccentricity in user model tailoring. To the best of our knowledge, this thesis is the first study which uses Q-analysis and eccentricity to cluster a hypernetwork and dynamically tailor a user model with this approach.

Main reasons for selecting hypernetworks to approach this problem are as follows: (i) Hypernetworks support representing n-ary relations by preserving order and (ii)in our multi-level model, Q-analysis technique provides an easy to implement, scalable tailoring solution.

Personalized search is the task of providing the most relevant results for the individual in a web search. There are various strategies in literature [43, 59, 27, 118, 14, 76, 119, 54]. We re-rank non-personalized search results by defining simple heuristics and applying them to dynamically tailored user profile. We evaluated this case study by using one-month log data of Yandex search engine. The dataset contains more than 167 million records. We improved Yandex's non-personalized ranking algorithm. This case study illustrates how a personalized service is provided with a tailored user model based on context and how basic heuristics is applied on this tailored model.

1.2 Contributions of the Thesis

Main contributions of this thesis can be summarized as follows:

- The huge amount of data available on the internet makes the need for effective personalization and recommendation techniques inevitable. The personal and professional interests of the individual are already available in several social web accounts. We aggregate those partial profiles of the user obtained from distributed social web sites into one holistic user model.

- The representation capability of the system basically depends on the user modeling structure. We propose a hypergraph based user modeling framework, since hypergraph is capable of representing higher order relations effectively.

- The hypergraph based structure facilitates aggregating partial profiles into a complete user profile by using the proposed semantic aggregation methodology. The defined aggregation methodology disambiguates and semantically enhances the given partial user profile terms by using a knowledge base.

- The proposed framework exploits a middle ontology to semantically enhance the user model. The domains are also managed by the employed middle ontology. Using a middle ontology which is small in size is advantageous when writing domain based algorithms compared to a large ontology.

- The user modeling structure directly effects the querying capability of the system. The proposed framework aims to provide effortless solutions to connected data problems. Most of the user modeling domain problems can be transformed into connected data problems. Therefore, our user model is designed to be beneficial in user modeling domain applications.

- We utilized the hypergraph based user modeling framework in several case studies to illustrate the solution for various connected data problems. The proposed framework naturally supports writing specific algorithms for user modeling domain problems. A recommendation system is presented as case study in order to show the straightforwardness and simplicity of writing algorithms for user modeling domain problems. The system is capable of exposing the semantic profile of the individual, recommending items, computing the user's interest on a specific item, discovering the users who might be interested in a particular item and discovering similar users.

- The model is widely evaluated with several social web sites including Facebook, Twitter, LinkedIn and StackOverflow and scores are high.

- We also presented another user modeling approach based on hypernetworks. The methodology is based on modelling the individual as hypernetwork with a multi-level approach.

- This thesis is the first which applies clustering on hypernetwork using Q-analysis and eccentricity.

- The proposed system provides user model to several personalized services based on their context.

- How the proposed methodology is used in personalized search is illustrated by evaluating the methodology against one month long Yandex search logs which contain over 167 million records and slightly improved Yandex's non-personalized ranking which is already a well performing baseline.

1.3 Organization of the Thesis

This thesis is organized as follows:

In *Chapter 2*, we provide the background knowledge for the main topics covered in this thesis and review the related work. For background knowledge we focus on profile extraction and representation, consumption of the constructed profile, aggregation of partial profiles, hypergraphs and graph traversal and hypernetworks. The relevant literature is reviewed.

In *Chapter 3*, we introduce our hypergraph based user modeling framework. This chapter also covers the user model construction approach we propose which mainly consists of entity disambiguation, domain identification, semantic enhancement and user profile aggregation.

In *Chapter 4*, we present a case study to illustrate the employment of the proposed hypergraph based user modeling framework for a recommender system. The capabilities of the recommender system are presented as subsections including semantic user model, discovering potential users who are interested in an item, cross-domain recommendation and discovery of similar users.

In *Chapter 5*, we explain the evaluation details for profile aggregation and discuss the results. The chapter consists of the datasets, methodology and the results of the evaluation.

In *Chapter 6*, we extend the proposed hypergraph based user modeling framework by adding context information.

In *Chapter 7*, we provide another user modelling approach based on multi-level hypernetworks. We also propose a dynamically tailoring algorithm on hypernetwork using Q-Analysis and eccentricity.

In *Chapter 8*, we evaluated dynamically tailoring approach in personalized search case study. We present the evaluation details.

In *Chapter 9*, we conclude the thesis and address possible future work.

CHAPTER 2

BACKGROUND AND RELATED WORK

In this chapter, we present the related work on user modeling, recommender systems and hypergraphs. We focus on methodologies to extract user profiles, different user model representation structures and several partial profile aggregation approaches. We discuss the ways of profile consumption including recommendation. In this thesis, we propose a hypergraph based user modeling framework which provides effective solution to several user modeling domain problems such as recommendation. Therefore, in this chapter we also present the background information on recommender systems and hypergraphs.

2.1 Profile Representation

User model is the representation of an individual's interests, preferences, goals, demographic or physical information, characteristic properties etc. in a structured format. *User profile* is the instantiation of the user model for a specific individual. However, the terms *user model* and *user profile* are used interchangeably. There are different possible ways to structure the user's profile information. *Profile representation* is the definition of the structure which is specialized to store the *user profile*. For instance, if the user's profile consists of keywords and the system stores the keywords constituting the profile in a comma separated file; then the profile representation is the comma separated file. In this section, we introduce related work for fundamental profile representation approaches.

[48] classifies user model representation methodologies as *keyword profiles*, *semantic*

network profiles and *concept profiles*. [49] extends this classification by introducing two dimensions. *Data structure* dimension considers how the user profile is stored. *Keyword* and *semantic network profiles* are categories for *data structure* dimension. *Content* dimension considers the nature of the terms in the profile which may be free keywords or entities from a knowledge base. We introduce a hypergraph based user profile which uses a knowledge base. Therefore, our user model could be classified as a combination of semantic network model and conceptual model.

Keyword based is the simplest profile representation methodology. Basically a set of keywords are used to define the user. Keyword based profiles are generally represented by using vectors, therefore they are also called as *vector based* user models. *Term* or *keyword* means the items of this type of user representation. In general, weights, which are numerical values representing the importance of the item for the user, are associated with the terms in the user profile.

For the illustration of vector based profile representation, let us say $V = v_1, v_2, .., v_n$ is the set of terms. Then $X = x_1, x_2, ..., x_n$ is a weighted keyword based profile in which x_i shows the weight for the term i. In another representation, $P = v_1 : x_1, v_2 : x_2, .., v_k : x_k$, the user profile keeps track of the terms that are in the profile and their weights. An example user profile *{tennis:0.5, football:0.1, reading:0.9, cooking:0.6}* shows that the user likes reading and cooking and she does not like football so much. Representation of user profile as a weighted keyword vector is very common in literature [81, 82, 97, 10, 101, 108, 33, 106].

When the terms in the keyword based profile are free keywords that are not attached to a knowledge source or vocabulary, then ambiguity problem arises due to polysemy and synonymy. [105] improves weighted keyword representation by using weighted word sequences. The study represents user profiles as word sequences which contains *n* terms. This is called weighted *n-grams* representation. Using word sequences means derivation of phrases instead of keywords, which helps to solve ambiguity issue to some extent.

Despite ambiguity issue and being the simplest representation methodology, [13] states that keyword based user modeling is practically effective in real world situations. The study presents a system which tracks the web pages the user visits and efficiently extracts keywords. Therefore, they try to increase the performance of their keyword based model.

In our study, we did not employ a vector based user profile representation because of two reasons: *(i)* In our model, we need to represent the semantics of the concepts and *(ii)* We need to model relationships inside the user model. In other words, we are building a highly connected user model and keyword based profiles are incapable of supporting relations and semantics.

Semantic network profile representation is capable of modeling high level concepts and relationships between them. Semantic network profile representation uses network of nodes instead of vector [48]. The node represents a word or concept which is an idea and its associated collection of words. For instance, *dog* is a word, whereas *Animal rights* is a concept and it contains the word *dog* in its associated word set [48]. Therefore, semantic network profiles are better at solving polysemy and synonymy problem than keyword based profiles. To solve polysemy issue, [9] models the user by using a weighted semantic network. In the network, the nodes correspond to concepts found in documents and arcs connect the concepts that co-occur in the same document. Similarly, [107] uses nodes for concepts and connect them with weighted arcs when they co-occur in same documents.

Our proposed user model resembles a semantic network profile in terms of using nodes and arcs that connect them. However, we aim to represent more complicated semantics and relations in our model than co-occurence information of concepts.

Conceptual profiles use concepts from a knowledge source or a vocabulary instead of keywords [49]. Knowledge sources could be domain specific databases created by experts, general knowledge sources such as Wikipedia, Wordnet or ODP (Ontology Design Patterns) hierarchy. In literature, ontology based user profile representation is common, because ontology usage results in structured knowledge in user profiles. Moreover, since ontology provides a common language, interoperability between applications using similar ontologies is naturally supported [29, 30, 24, 100].

It is possible to exploit different ontologies in different ways to represent a user profile. For instance, [30] develops the ontology based user model as overlay over conceptual hierarchies, whereas [24] constructs the user ontology by tailoring the YAGO general purpose ontology according to user's interests. In this thesis, we use Freebase knowledge base indirectly. As the system is populated by user profiles, newly encountered concepts are disambiguated by using Freebase and then imported to the system. This means tailoring Freebase from a point of view, however our user model uses its own defined relationships instead of relationships in Freebase.

Ontologies empower using propagation on the structure to calculate weight and similarity of user interests [29, 30]. In an ontology, horizontal propagation enables traversal among siblings whereas vertical propagation visits ancestors and descendants. By applying propagation, it is possible to extend the user profile. For instance, the user profile states that the user is interested in *tennis*, and the ontology locates *tennis* under the more general term *sports*. In this scenario, by propagating in the ontology, it could be inferred that the user also likes *sports*.

When the user profile is very sparse and it is not adequate to personalize, it means that *data sparsity* or *cold start problem* arises. Enhancing the original user profile by propagating in the structure, contributes to the solution of the cold start problem [30]. In this study, our solution to this problem is propagating in the user model to extend it, as well. Extension of the user profile means the *semantic enrichment* of the user model. The *semantic enrichment* is accomplished by disambiguating the concept by linking to an external vocabulary, using a secondary vocabulary when the concept could not be linked, enriching the concept by adding sysnsets, expanding the concept by retrieving related concepts from the external vocabulary according to a predefined traversal algorithm, by using friends or like minded users' profiles as explained in the survey [1]. We achieve semantic enhancement by using a middle ontology in front of the external vocabulary and calibrating the middle ontology concepts according to system requirements.

Besides ontologies, graphs are appropriate to represent user profiles. [100] proposes a graph based framework which extracts named entities from the individual's tweets and links them to a knowledge base. The user model could be represented by using bipartite[121], tripartite [31] graphs and hypergraphs [68, 65, 23]. In this thesis, we define a hypergraph based data structure to represent the profile. Hypergraphs are very powerful in terms of representation, they are capable of representing not only binary relations as ordinary graphs, but also higher order relations.

2.2 Profile Extraction from Social Networks

In order to populate the user profile, information about the individual should be obtained implicitly or explicitly. The user could be asked about himself/herself to gather information explicitly. However, explicitly asking users about themselves is an awkward and unreliable task. Therefore, using platforms that already contain information about the users in order to *extract profile* implicitly is more appropriate.

Since social networks are satisfactory information sources to implicitly collect interest areas of the individual, they are used for user profile extraction. There are studies which analyse social media websites in terms of semantics [21, 34, 80, 1]. Social web sites are categorized according to the information sharing methodologies, user communication behaviours and user interaction with the media streams [21]. For instance, Twitter is classified as an interest-graph media where people connect based on shared interest areas, Facebook is a social network site where people connect with people whom they are connected in real life and LinkedIn is a professional networking service where people connect based on work life. Besides there are content sharing websites such as YouTube and discussion forums such as StackOverflow.

There are ontologies for social media such as FOAF (Friend of a Friend) which describes people, SIOC (Semantically-Interlinked Online Communities) which models community sites, MOAT (Meaning of a Tag) which enables describing a tag semantically, GUMO (The General User Model Ontology) which is a general user modeling ontology [21, 90]. Linked open data resources such as DBPedia [1] and Freebase [2] could be used for semantic annotation and entity disambiguation [21, 2].

In literature, there are studies which employ Twitter stream to extract entities and interests and build multi-domain user profiles [62]. Domain dependent user profile extraction is also possible. For instance, professional user profile could be extracted

[1] DBPedia, http://wiki.dbpedia.org/

[2] Freebase, https://www.freebase.com/

by only considering expertise interests of the individual [6, 52]. User profiles could be created by using structural and temporal nature of tagging data in social networks [78]. Tag frequency and co-occurrence information of tags increases the quality of extracted profiles.

There are several studies which exploits social networks for different purposes. For instance, the goal is to build a comprehensive view on social user profiles in [63] and a reference model for social user profiles is presented. The reference model includes a generic core and enables extensions and representation of meta information. Another study focuses on privacy and proposes a privacy aware, faceted user profile extraction system [95]. A different study concentrates on expanding the user's query based on the individual's social context to prevent disambiguation [65]. [70] utilizes inferred location information for advertisement and news recommendation applications. [55] uses social networks to construct user profiles as semantic interest graphs and employs them in a cross domain recommendation framework.

Besides social networks, observing the individual's web usage patterns reveals important details about him/her and could be used to extract user profile [12]. [96] presents a browser-based user modeling framework for saving lifelong user model efficiently in the limited web browser environment. [89] uses the individual's latest click history to personalize search results. [128] employs the user's queries in a session to determine the user's short term interests.

In this thesis, we extract profile information from social networks including Facebook, Twitter and LinkedIn. To construct the partial profiles from Facebook, we used items that are provided by Facebook API which includes posts, check-ins, page likes for categories activity, book, game, interests, movie, music, tv and uncategorised page likes. For LinkedIn partial profiles, we used LinkedIn API and it provides access to full profile of the user including user's skills, specialities and interests. For Twitter, we used the user's description in his/her Twitter profile and checked whom he/she follows, since in Twitter follow list is a good indication of interest on a named entity. For instance, if the user follows an account named as *Java Code Geeks*, this shows that he/she is interested in the programming language *Java*. It is possible to improve the partial profile extraction algorithms to obtain more qualified partial profiles. However, this is not in the scope of this thesis and left as future work. Moreover, the proposed system is also easily extendible for other information sources. For instance, during evaluation we extended the system for StackOverflow.

2.3 Profile Aggregation

An empirical study on the way how users distribute their information amongst different social web accounts shows that *aggregating* separate profiles increases the quality of the ultimate user model [34]. Aggregating partial profiles of the individual solves the cold start problem by enabling the reuse of the user profile across different applications and results in a more complete modeling of the user [85]. Several issues such as entity matching, resolution of duplicates and conflicts, and heterogeneity of the partial user profiles should be addressed to develop an effective aggregation methodology [85]. Furthermore, the objective of the aggregated user model influences the aggregation strategy. In literature, there are diverse aggregation approaches.

There are studies which aggregate distributed portions of the user profile with the aim of modeling the user more accurately [85, 62, 3, 72, 58, 4, 127, 17, 50]. Social web platforms are beneficial data sources to gather information about the individual. It is possible to extract partial profiles from social web accounts of the user and aggregate them into one complete user profile. In this thesis, we basically adopt this approach. There are examples of this approach in literature.

[85] provides a user profiling framework with an aggregation algorithm for scattered profiles over several social web sites. The study extracts data from each supported social web site, Facebook and Twitter specifically. During data extraction, they treat every social web account differently by considering its nature. In Twitter, they exploit the most recent statuses to extract the partial profile, whereas in Facebook they use status messages, liked entities, check-in data and demographic information. After raw profile data is collected from social networks, a named entity recognizer is used to extract entities such as people, places, etc. Entity disambiguation is accomplished by using DBPedia. The study keeps track of *provenance data* for each raw profile item. The provenance data contains metadata for the user profile item such as the

source of the item and the timestamps. Usage of provenance data is beneficial in two ways: *(i)* it allows to employ an exponential time decay function to assure giving higher weights for the latest interests and *(ii)* it enables the recalculation of item weights during aggregation of the partial profiles. Once partial profiles are ready for aggregation, the study merges them by assuring that *(i)* duplications for the items that reoccur in more than one partial profiles should be eliminated and *(ii)* a global weight should be calculated for the items in the profile. The global weight for the reoccurring items should be higher. The study assigns importance percentage to each social web account the partial profiles are extracted. They calculate the global weight by taking an accumulation of the weights in the partial profile factored by the importance of the partial profile. For example, assume both Twitter and Facebook profiles indicate that the user is interested in *Roger Federer* and weight for Twitter is 0.8 and for Facebook is 0.7. Assume Facebook is assigned an importance value of 0.6 and Twitter's weight is 0.4. Then the global weight is calculated as $0.6 * 0.7 + 0.4 * 0.8 = 0.74$. In [62], they extend the aggregation for LinkedIn and keep track of the public Twitter stream and filter the tweets for the user based on his/her aggregated profile. In our thesis, we delay entity recognition and disambugiation tasks until the aggregation phase. This eliminates the unnecessary preprocess applied to the partial profiles. Moreover, we calculate weight only once during the aggregation. In short, the delay results in performance gain. In our case studies, we focused specifically on recommendation. However, our proposed framework is capable of supporting tweet distribution based on user profile as well. Another good practice in the study is usage of provenance data. We also keep track of the provenance data by storing the knowledge source, the short term profile date and the exact keyword of the item. We extend this information each time the item and user is bound together. We introduce a specific hyperedge type for keeping track of the provenance data. We use a hypergraph based structure which both helps to simplify aggregation and answering queries which can be solved by providing traversal algorithms on the graph. Besides, our aggregation approach is highly scalable, it supports newly added knowledges sources once partial profiles are provided for them.

[3] aggregates partial public profile information from several social accounts including Facebook, LinkedIn, Twitter, Flickr and Google by representing partial profiles

as key-value pairs and integrating these pairs into a uniform user model. The study focuses on illustrating to what extend partial profiles complete each other. For example, it states that incomplete Twitter profiles could become 98% complete by adding profile information from other sources. According to the study, the completeness of profile means the existence of 17 distinct attributes about the individual. The study is important for us, since it shows that aggregating information from different social web sites indeed provides a more complete profile of the user.

In [5], social web accounts are categorized in two groups: the web sites that the user fills in forms providing demographic information and the web sites which enable user to tag items. The aim of the study is to analyse the content of the partial profiles. Therefore, the authors handle aggregation of form-based and tag-based profiles separately. The former is a list of attribute-value pairs whereas the latter is a set of weighted tags. The aggregation strategy for form-based profiles is unifying sets of attribute-value pairs. Heterogeneous attribute vocabularies is resolved by using an alignment function which maps profiles to unified attribute-value space. However, this alignment function may result in duplicate entries in the final user profile. Moreover, when there are conflicts in the aggregated profiles, both values are included in the result. The aggregation of tag-based profiles is accomplished by taking a weighted accumulation of partial tag-based profiles. The semantics for tag-based profiles is accomplished by linking entities to Wordnet categories and named entities to DBpedia. The authors do not consider aggregating tag-based profiles and form-based profiles with each other. In our study, we do not make such a distinction. We seamlessly aggregate received partial user profiles by taking their weighted accumulation. We solve heterogeneous vocabulary problem by using an external knowledge base such as Freebase.

The work in [123] does not classify user profiles either. The study sorts the old profile items according to assigned weights, drops the lowest weighted items and adds items from the new profile to merge old and new profiles. However, we think that pruning old profile prior to aggregation may lead to wrongly assigned weights. We handle conflicted information about the user by considering the origin of the information. The origin of data is the provenance data we keep. The provenance data contains metadata such as the knowledge sources and timestamps of the profile items. It is possible to resolve data conflict by defining hand crafted rules that check the provenance data. For example, assume the user's raw Facebook profile item states that he/she has a job at *ABC Company*, but his/her LinkedIn profiles claims that the user works at *XYZ Company*. The first check would be timestamps of the statements. The latest information is more reliable. If the timestamps are close, then LinkedIn is more trustworthy for professional profile and it is picked as the correct one. Not all rules are implemented for conflict detection and it is remained as future work.

There are studies which exploits an ontology during aggregation. They map partial profile terms to specific locations in the ontology. For instance, [127] proposes a FOAF based profile aggregation approach. The study concentrates on the connection network of the individual, therefore FOAF ontology is adequate in that context. In the study, partial profile terms are mapped to specific FOAF properties by using a set of hand crafted rules. [126] is another study which adopts FOAF for aggregation of partial profiles by mapping the user profile items to FOAF properties by defining hand-crafted rules. The aim of the study is to support group decision making. As we can see, FOAF usage for aggregation is useful as long as people and their friend network is concerned. However, in this thesis our focus is not the network of the user, but his/her interests. Therefore, we did not limit the content of user profile to demographic information which can be represented by FOAF.

In [129], the aggregation is handled by semi-automatically extracting schema from social web data and integrating the extracted schemata with existing integration tools. The study basically collects data for partial profiles from Facebook, LinkedIn and Google+ [3] social accounts and extracts schema for each knowledge source by examining the collected data. Afterwards, the extracted schemas are transformed to technical spaces which can be processed by existing schema integration tools. Finally, the preprocessed partial profiles are integrated by using external tools. In this thesis, we want to aggregate profiles fully automatically. Semi-automatic integration step prevents the system to serve in real time.

In [87], an aggregation ontology is proposed to semi-automatically aggregate partial user profiles. The presented ontology is generic and defines the mapping between

[3] Google+, https://plus.google.com/

all pairs of knowledge sources that the partial profile is extracted. In our study, we propose an extendible generic user modeling framework. However, the aggregation ontology in [87] requires each mapping to be defined for each knowledge source. When a new knowledge source is going to be supported, our system does it effortlessly, whereas [87] should manually add the mappings to the aggregation ontology.

In literature, automatic discovery of the user's social web accounts is also a studied research area [72, 58, 4]. For instance, [72] focuses on discovering different social web accounts belonging to the user by applying automated classifiers and using UserID and Name as discriminative features. Another study abstracts a social network account by separating it into three dimensions including profile, content and connection network [58]. The study compares social accounts in these dimensions to discover the accounts belonging to the individual. [4] discovers the user's several online accounts given one of his/her social account and collects and aligns profile information by defining hand-crafted rules. The study enriches the profiles by using Wordnet categories. In this study, discovery of different social web accounts of the user is out of scope. However, the system could be extended to support this feature.

[117] structures the profiles as high and low granularity levels. This separation supports detecting the user's most important interests. [121] states that feature selection during aggregation of profiles affects the quality of the final profile. [17] claims that the success of the ultimate profile mainly depends on the quality of the partial profiles. The study mediates the partial user profiles across the network of applications instead of aggregating them.

Applying entity disambiguation results in better aggregation of profiles. In general, entity disambiguation means to find an entity in an ontology or knowledge base for a keyword. Ontologies or knowledge bases could be very large in size, which makes querying them difficult. Therefore, effective entity disambiguation techniques are essential while using knowledge bases [44]. [64] uses social network context to infer additional keywords for a search query . [133] uses Freebase for entity disambiguation, since it contains more entities than Wikipedia and others.

Freebase is an ontology used to structure general human knowledge [19, 20]. Easy-to-use APIs (Application Programming Interfaces) or MQL which is an abbreviation for metaweb query language could be used to query the knowledge base. The graph-shaped database contains more than 4000 types and 7000 properties [19]. The large number of types and properties results in difficulty and inconvenience in writing general semantic algorithms. In Freebase, a *metaschema ontology* which constructs another layer over huge Freebase ontology is defined. The *metachema properties* provide higher order relations between concepts and there are 46 properties. The small size and abstraction of metaschema properties enables writing generic semantic algortihms which uses Freebase. In our thesis, we exploit a reduced subset of metaschema properties for semantic enhancement.

There are many studies which use Freebase for semantic enrichment [110, 42, 131, 100], alignment [40] and disambiguation [44, 133, 124]. In our work, we choose to use Freebase for entity disambiguation and semantic enhancement, since it is a general knowledge base, its API is easy to use and fast and it provides a middle ontology which enables us to write less code while semantically enhancing the user model. To the best of our knowledge, our user model is the first study which uses Freebase metaschema properties during semantic enhancement.

2.4 Recommender Systems

Aggregated user profiles could be consumed by several personalized applications such as adaptive web [8], personalized search and recommendation. In this thesis, the objective of the aggregation is two-fold: *(i)* to obtain a user model based on a hypergraph which reduces connected data problems such as recommendation into graph traversal algorithms and *(ii)* increasing recommendation accuracy with the proposed semantic enhancements. Therefore, in this section, we introduce basics and related work regarding recommender systems.

Recommender systems provide suggestions for items that might be interesting to the user [91]. *Item* is a term which states what the system recommends. The system has an internal decision making process to decide what to suggest.

Domain based recommendations focus on only specific domains such as movie, music or news recommendation. General recommendations may suggest any item from different domains. *Cross domain recommender systems* are able to exploit the user model for other domains providing a natural solution to data sparsity problem.

Cross domain recommender systems enhances recommendations in a domain by using other domains [26, 56]. Cross domain recommendations are available in social networks. [55] models user profiles as semantic interest graphs and exploits them to provide cross domain recommendations. [56] proposes *spreading activation* model that interconnects entities from different domains with each other.

Recommender systems are classified according to the suggestion algorithm [7]. In *content based recommendation*, the system suggests items to the user that are similar to the items in the user's profile. In *collaborative recommendations*, the items to suggest are selected by regarding user profiles of the other users that are known to be similar to the individual. Collaborative filtering and content based recommendation approaches mainly depends on the domain of concern and the source domains from which the user's profile is extracted. In *hybrid* approaches both content and collaboration information are considered. In this thesis, the proposed framework is capable of supporting all recommendation approaches.

When the recommender system tries to suggest items to a brand new user with an empty or sparse user profile, *cold start problem* occurs. [99] uses existing profile information in the user's Facebook profile to overcome this problem. The study shows that, using Facebook profile significantly improves the results when the user's profile is sparse or absent. [122] surveys several social web sites to examine their effectiveness in recommendation. [98] combines content and collaborative approaches to solve cold start problem.

[42] provides content based recommendations in movie domain by using Linked Open Data sources DBPedia, Freebase and LinkedMDB. [131] uses Freebase to bridge the gap between search engines and recommender systems.

[86, 11] proposes a hybrid video recommendation service on YouTube which uses *Adsorption* technique to propagate user's preference information efficiently. *Adsorption* is a collaborative filtering algorithm which uses relations between users and it is enhanced by content based filtering [86]. [57, 32, 31] provides personalized video

suggestions by exploiting the relations between users, videos and user's queries to search for videos. An iterative propagation algorithm on a tripartite graph between users, videos and queries executed by users is proposed in [31]. The algorithm is based on the behaviour information modelled in the graph and outputs the preference of each user for every video. We use a similar method of calculating the item weights of the user on each reachable item on the hypergraph. [66] aims to develop a system which is capable of understanding not only what people like, but why they like it. [88] focuses on evaluation of recommender systems.

Recommendation could be managed by separately constructing short term and long term user profiles [69]. User profiles are managed as a sequence of short term profiles for predefined time periods in [69]. The authors construct the long term profile by accumulating short term profiles with a time sensitive weight function. The employed weight function ensures that older short term profiles are assigned with lower weights. Another work which represents user models by using FOAF ontology, also uses an exponential time decay function [85]. The use of FOAF enables the integration of partial profiles by using semantic web technologies.

The user profiles in [69], are used in recommendation in two steps: Firstly, the long term user profile is exploited to roughly capture user's interests and select the most relevant clusters. Secondly, the latest short term profile is utilized to locally sort items in the clusters. We are inspired by the idea of constructing the user's long term profile by taking a weighted accumulation of short term profiles by using a time decay factor. Moreover, we adopted a similar approach in our case study: using long term user profile for detecting user's general areas of interest, and then applying the most recent short term profile to discover his current interests amongst them.

2.5 Graphs and Hypergraphs

A *graph* is a data structure which consists of *nodes* and *edges* where edges connect nodes to each other. *Node* and *vertice* are used interchangeably to denote the same concept. Ordinary graphs are capable of representing binary relations. Representation of relationships that are more complex than pairwise could be accomplished by utilizing hypergraph data structure [134]. Graph based data structures naturally support *connected data problems* which defines the problems that could be converted to graph traversal problems.

Most user modeling and recommendation problems are connected data problems. Connected data problems are solved by generating appropriate traversal algorithms which traverse the sub-graph related to the problem. The expressiveness of a data structure is evaluated by its ease of use rather than its representation capability [94]. Therefore, the proposed data structure should be traversed in an effective manner. The study also claims that user modeling and recommendation problems can be easily solved by making a *short-cut* to the graph with an external index and traversing the graph beginning from this short-cut. The authors formally define primitive graph traversal operations and present several examples. In our thesis, we adopted the approach illustrated in [94] in the formulation of our problems. Moreover, the node labels and edge types in the hypergraph based user model can be used for filtering purposes in the traversal algorithm.

Property graphs are obtained by adding key-value pair properties to ordinary graphs and it is possible to model hypergraphs by using property graphs [93, 92]. [22, 46, 47] explain hypergraph data structure in detail.

In literature, there are studies that exploit graphs [41, 37, 38, 39, 35, 130, 31] and hypergraphs [23, 111, 71, 68, 83, 94, 120] for proposing solutions to different kinds of problems. [41] proposes a movie recommendation system which represents movie domain by using graph. The study suggests movies by traversing the graph based on the initial nodes and the user's interests. Graph usage results in the performance of the recommendation to be acceptable to be used in real time. [37] represents the user profile for a query session as a graph and exploits the constructed user model in personalized search. [38, 39] use a conceptual graph based user model for personalized search by reranking the search results according to the profile of the individual by defining a distance measure. [35] provides a spreading activation algorithm on graphs which aims to minimize the execution time. [130] proposes a framework which integrates friendship and interest graphs. [31] presents a video recommendation approach which is based on an iterative propagation algorithm over the tripartite graph which represents users, videos and queries and relationships between them.

[23, 111, 120] propose a music recommendation algorithm which uses hypergraph to model the domain. The recommendation problem is defined as a ranking problem on the unified hypergraph. The ranking problem is solved by using a group sparse optimization approach [120].

[68] proposes a news personalization framework which uses hypergraphs to model the news domain. The study defines recommendation as ranking problem on the constructed hypergraph.

[83] proposes an algorithm for community detection which uses k-partite k-uniform hypergraphs. [134] utilizes hypergraphs for clustering purposes. [71] provides a reference model for representing folksonomies as graphs and derive a hypergraph.

In this thesis, we propose a hypergraph based data structure which contains specific nodes and hyperedges that simplify writing algorithms for user modeling domain problems. Chapters 4 and 6 illustrate the usage of the proposed model. In short, we embrace the representation and querying power of hypergraph and adapt this power to the user modeling domain by proposing the specified hypergraph data model.

2.6 Hypernetworks

In [68], n-ary relations in news domain are modeled using hypergraphs. News recommendation problem is decomposed into two sub-tasks: separating the hypergraph in partitions and ranking based on the most relevant partition. The authors partition the entire hypergraph which contains data about all the users. This is not scalable, since the hypergraph grows in time with new data and users. We eliminate scalability problem by processing only the individual's profile data. Moreover, they use spectral clustering algorithm which might result in imbalanced clusters. Spectral clustering algorithm constructs a matrix representation of the graph, computes eigenvalues and eigenvectors of the matrix, maps each point to a lower-dimensional representation based on one or more eigenvectors and assigns points to two or more clusters. Spectral clustering is expensive for large datasets because of the eigenvector computation step. [73] provides an efficient parallel algorithm to compute eigenvector faster. However, since we aim our algorithm to operate on mobile and ubiquitous devices with little memory, parallel processing is not suitable for our case. When few clusters capture most of the hypergraph and others contain few data, performance gain due to partitioning step is eliminated. To avoid this, we cluster the hypernetwork which contains the individual's profile by using Q-analysis and eccentricity. Since eccentricity is used as a control condition on clustering iterations, the possibility of imbalanced clusters is reduced.

In [65], users' web activities are modeled using hypergraphs. However, only one hyperedge type is defined and hypergraph operations or properties are not utilized. The authors handle personalized search problem by examining not only the individual's profile but also similar users' profiles. In this thesis, we tailor the individual's profile and only process this dynamically tailored profile. During tailoring we use Q-Analysis technique with eccentricity for clustering purpose. We use provided test query and tailor the user model by keeping the most relevant parts regarding this test query. To the best of our knowledge, our study is the first which tailors the user model by using Q-Analysis technique. In evaluation, we showed that using the tailored user model performed better than using the entire user model. Using similar users' tailored profiles might improve the results. However, we left this as future work.

In [23], music domain is modeled using unified hypergraphs. The number of vertex and hyperedge types are specified and only triple relations between entities are allowed. Our user model is generic and not restricted to specific vertex or hyperedge types.

Hypernetwork usage is a new approach in user modelling domain. In [132], the authors use Movielens dataset to construct a hypernetwork of two object sets: users and movies. The authors convert the hypernetwork to bipartite-hypernetwork to examine relations between users and movies. A hypernetwork can be converted to a bipartite-hypernetwork only when there are two object sets. Therefore, the study is not capable of representing n-ary relations.

In [109], objects rated by the same user are encapsulated in the same hyperedge. The authors define topological properties on hypernetworks. These properties are mainly based on vertex and edge degrees which defines the number of connected vertices and hyperedges and used to analyze the inner dynamics of the dataset. Both [132] and [109] are restricted to cases in which users rate objects. By using rating information, they define similarity between hyperedges. In this thesis, we cluster similar hyperedges together by using Q-Analysis and eccentricity without using rating information. Therefore, our approach is applicable to cases in which rating data is not available.

There are also predictive statistical models which use machine learning algorithms to personalize [135, 25, 28, 61]. In general, they perform well as we stated in evaluation of personalized search case study. However, they require training phases which prevents them to be used real time. Moreover, they require large amounts of data for their training phase. Since our goal is to support personalized services in real time, we employed hypernetworks instead of statistical approaches. Moreover, since these approaches use machine learning, feature selection is important which adds an extra step. Hypernetworks does not have such requirement.

In [18], the authors focus on structure and dynamics of multi layer networks. We inspired by the idea of using multiple layers and combined this inspiration with object-oriented approach. Our proposed methodology simply models the user by starting from lowest level entities and relations. Then higher level entities and relations are modeled by using previously modeled hyperedges.

In summary, we use hypernetworks to build a multi layer user model. In this model, the most specific items form the bottom level and upper levels are constructed by reusing items from lower levels. This allows us to use Q-Analysis technique for clustering purposes by applying it from bottom to top in the multi layer user model. Using eccentricity as a threshold eliminates creation of imbalanced clusters.

CHAPTER 3

HYPERGRAPH BASED USER MODELING FRAMEWORK

In this chapter, we propose the hypergraph based user modeling framework in detail. We first introduce the general hypergraph concept and then present our framework. The user model construction process is explained in detail by providing algorithms for entity disambiguation, domain identification, semantic enhancement and user profile aggregation.

3.1 Preliminaries

Hypergraphs:

Hypergraphs are powerful data structures and they facilitate the modeling problems in many application areas [47].

Definition 3.1. A hypergraph H can be defined as a pair $H = (V; E = (e_i)_{i \in I})$ where V is a set of vertices, and E is a set of hyperedges between the vertices. I is a finite set of indexes.

A *hypergraph* generalizes a binary edge of an ordinary graph by enabling the edge to connect an arbitrary number of vertices instead of two [93]. An example hypergraph might illustrate the given definition [22]. For instance, M denotes for a meeting which has $k \geq 1$ sessions. The sessions are denoted as $S_1, S_2, S_3, ..., S_k$. The assumption is that ast least one person attended each session. A hypergraph H which models this situation is $H = (V; E)$ where the set of vertices V stands for the set of people who attend the meeting whereas the set of hyperedges E is $(e_i)_{i \in 1,2,..,k}$ keeps track of the people's attendance to the sessions.

Hypergraph theory is originally developed by Berge in 1960 by generalizing the graph theory. [22] presents the hypergraph theory in detail.

Property Graphs:

From practical point of view, there are three types of graph data models which are used by graph database management systems: hypergraphs, RDF triples and property graphs [92]. Graph databases support create, read, update and delete (CRUD) operations on the selected graph data model.

Hypergraphs are difficult to implement. Therefore, in this thesis we implement hypergraphs indirectly, by using a data structure which facilitates implementation and totally convertable to a hypergraph. A *property graph* is a directed, labeled, attributed graph. A property graph *(i)* contains nodes and relationships, *(ii)* relationships are named and directed and *(iii)* both nodes and relationships can contain properties which are key value pair attributes [92]. In the simplest conversion algorithm, both vertices and edges in the hypergraph are denoted as vertices in the property graph. The equivalence of the structures in this context is illustarted with an example in Section 3.2.

3.2 Overview

A hypergraph is defined as the generalization of an ordinary graph by introducing hyperedges which are non-empty subsets of the vertex set [46]. In user modeling domain, vertices of a hypergraph represent the entities to be modelled such as people and concepts. Similarly, hyperedges represent the relations between those entities.

Figure 3.1 illusrates a scenario which shows that the user likes *Pride and Prejudice* which is related to *Jane Austen* and is a *Fictional Universe*. The user with name *dummyUser* is represented by the *dummyUser* node and the wrapping circle stands for the *Users* hyperedge which encloses all the users in the system. Similarly, *FictionalUniv.* node represents the *Fictional Universes* domain and resides in the *Domain* hyperedge. The rest of the nodes represent areas of interest and wrapped by *Items*

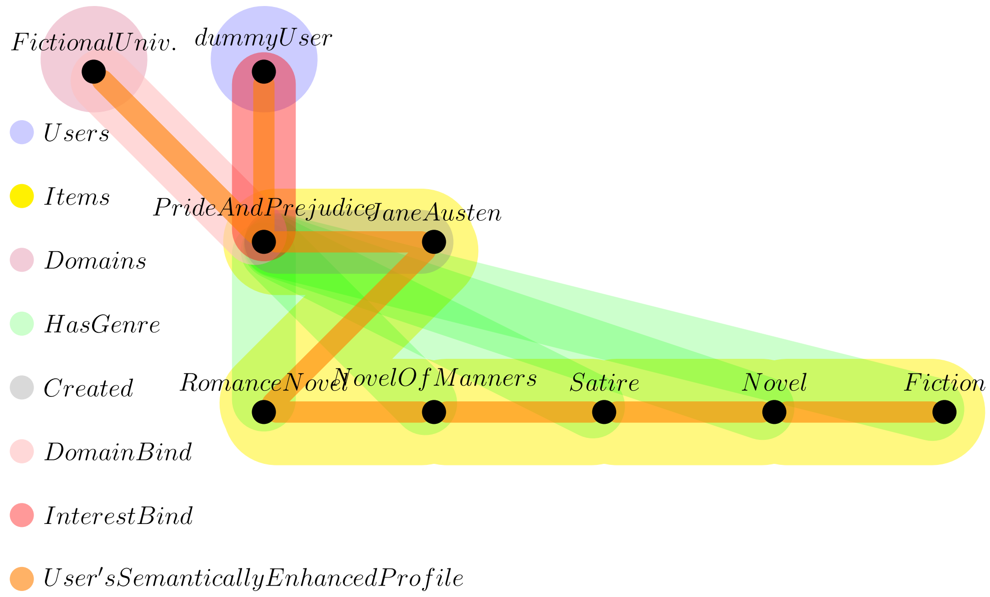


Figure 3.1: A Hypergraph

hyperedge. *HasGenre* and *Created* hyperedges indicate the semantic relations between items. The orange hyperedge shows the user's semantically enhanced profile which shows that the user is interested in a *fictional universe* item *Price and Prejudice* which is created by *Jane Austen* and *has genres romance novel, novel of manners, satire, novel* and *fiction*.

Property graphs are stated to be attributed, multi-relational graphs where nodes and edges are labelled and can have any number of key-value properties associated with them. They have the same representation power with hypergraphs [93]. Every hypergraph can be represented by a property graph by adding extra key-value pairs to annotate nodes which are connected by the same hyperedge.

In this thesis, we use property graphs in the implementation, since the graph database we adopted[1] supports property graphs. Moreover, defining traversal algorithms in property graphs is easier than in hypergraphs. In our study, using property graphs to implement hypergraphs is only an implementation decision, it is possible to directly use the hypergraph data structure for the proposed user model. Therefore, we named our user model data structure as *hypergraph based*. We presented traversal algorithms in property graph, since representing traversal algorithms in property graph is easier than visualising hypergraph.

[1] Neo4j, http://www.neo4j.org/

The equivalence of a hypergraph and the corresponding property graph is illustrated in Figure 3.1 and 3.2. Different node types are connected by different hyperedges in hypergraph, where they are assigned different labels or have distinct types in property graph. In the property graph, *dummyUser* is a node with type *UserAccount*. Similarly, the domain *Fictional Universes* is a node with type *Domain* and items *Jane Austen*, *Romance Novel*, *Novel of Manners*, *Satire*, *Novel* and *Fiction* are nodes with type *Item*.

In the hypergraph, the domain *Fictional Universes* and the item *Pride and Prejudice* are connected with a hyperedge indicating that the item belongs to the domain. In the corresponding property graph, the item *Jane Austen* is connected to the domain *Fictonal Universes* with an edge of type *DomainBind*. The edge is also labelled as *IsInDomain*. Likewise, the hyperedge between *dummyUser* and *Pride and Prejudice* indicates that the user is interested in the item. This information is represented with the edge labelled as *InterestedIn*.

Each semantic relation type between items are represented with different hyperedges in the hypergraph. For instance, *Created* hyperdge connects *Jane Austen* to *Pride and Prejudice* whereas *HasGenre* hyperedge connects *Pride and Prejudice* to its genres *RomanceNovel*, *Novel of Manners*, *Satire*, *Novel* and *Fiction*. In the corresponding property graph, edges with type *Inner* represents semantic relations between items. Different semantic relation types are labelled differently such as *CreatedBy* and *HasGanre*.

In the property graph, properties can be indexed by using a tree like structure. Therefore, a two step search on graph can be adopted: First the concept is located in the index structure and then with this *short-cut* to the graph, traversal algorithm can be applied. In graphs, cost of local read operations is constant, since adjacent vertices and edges are already connected. Since the traversal query performance is independent of the size of the graph, using graph databases for problems which can be solved by traversal-based approaches, is more efficient than using relational or NoSQL databases.

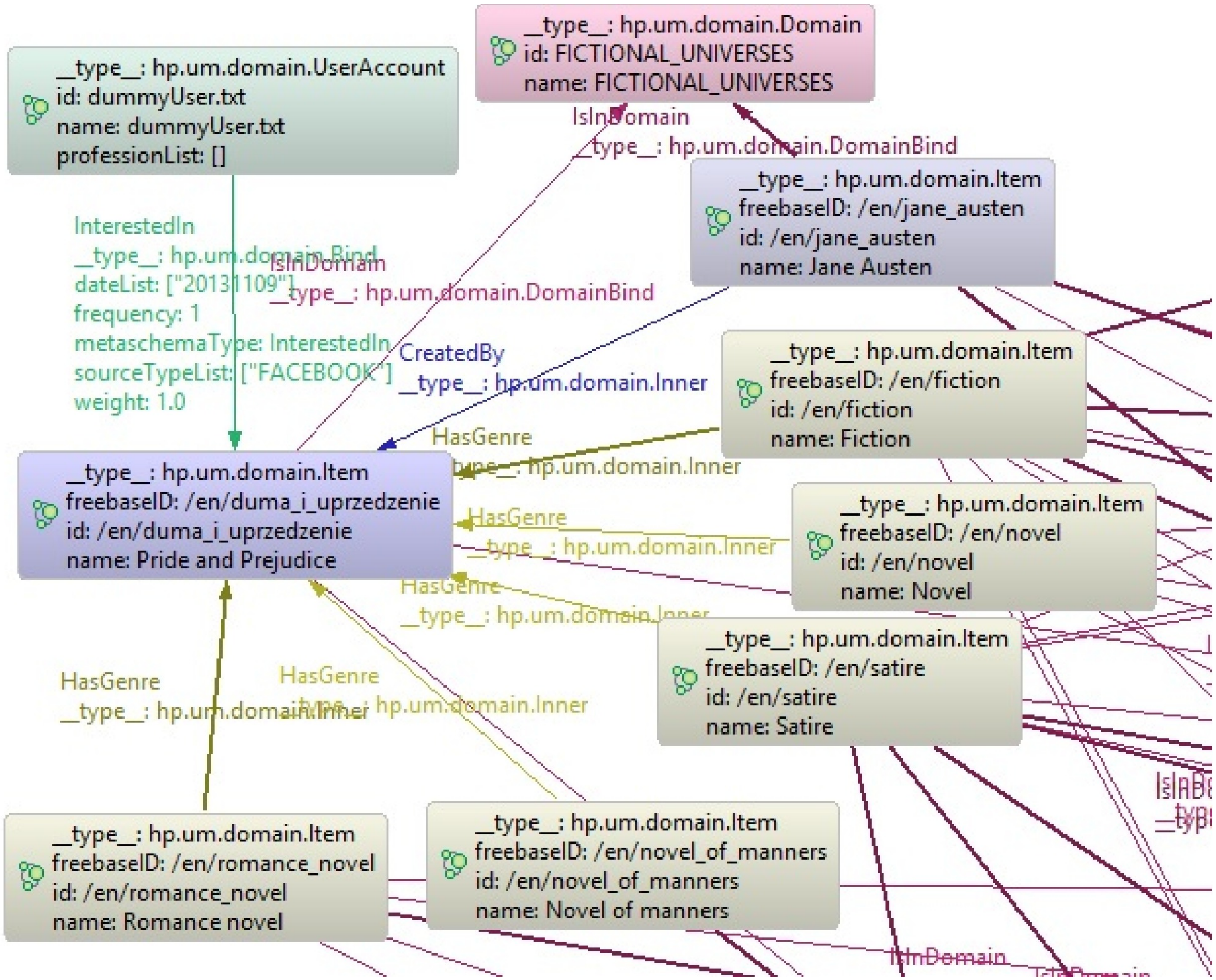

__type__: hp.um.domain.UserAccount
id: dummyUser.txt
name: dummyUser.txt
professionList: []
__type__: hp.um.domain.Domain
id: FICTIONAL_UNIVERSES
name: FICTIONAL_UNIVERSES
IsInDomain
__type__: hp.um.domain.DomainBind
__type__: hp.um.domain.Item
freebaseID: /en/jane_austen
id: /en/jane_austen
name: Jane Austen
InterestedIn
__type__: hp.um.domain.Bind
dateList: ["20131109"]
frequency: 1
metaschemaType: InterestedIn
sourceTypeList: ["FACEBOOK"]
weight: 1.0
IsInDomain
__type__: hp.um.domain.DomainBind
CreatedBy
__type__: hp.um.domain.Inner
__type__: hp.um.domain.Item
freebaseID: /en/fiction
id: /en/fiction
name: Fiction
HasGenre
__type__: hp.um.domain.Inner
__type__: hp.um.domain.Item
freebaseID: /en/duma_i_uprzedzenie
id: /en/duma_i_uprzedzenie
name: Pride and Prejudice
HasGenre
__type__: hp.um.domain.Inner
__type__: hp.um.domain.Item
freebaseID: /en/novel
id: /en/novel
name: Novel
HasGenre
__type__: hp.um.domain.Inner
__type__: hp.um.domain.Item
freebaseID: /en/satire
id: /en/satire
name: Satire
HasGenre
__type__: hp.um.domain.Inner
HasGenre
__type__: hp.um.domain.Inner
__type__: hp.um.domain.Item
freebaseID: /en/romance_novel
id: /en/romance_novel
name: Romance novel
__type__: hp.um.domain.Item
freebaseID: /en/novel_of_manners
id: /en/novel_of_manners
name: Novel of manners
IsInDomain

Figure 3.2: A Property Graph

3.3 Modeling Framework

Table 3.1: Our hypergraph based User Model

Notation	Description	Type
u	a user	Node
U	Set of users	Hyperedge
i	an item	A Node
I	Set of items	Hyperedge
$D_{[d]}$	Domain starter node for each domain d	Node
D	Set of domains	Node
E_{bind}	Metadata for user-item (interest) relation	Hyperedge
E_{inner}	The semantic relation between items	Hyperedge
E_{domain}	The domain bind between domain starter node and items	Hyperedge
E_{friend}	Friendship between users	Hyperedge
P_u	General (long term) user profile	A sub hypergraph

The proposed hypergraph based user model aims to facilitate aggregation of partial profiles of the individual. Moreover, the model expedites writing traversal algorithms for connected data problems in the user modeling domain such as recommendation.

The main components of the user model is summarized in Table 3.1. In the proposed framework, *users*, *items* and *domains* are represented with distinct node types U, I and D. The supported *domains* are predefined. Freebase commons package is used as domains. A domain starter node $D_{[d]}$ is created for each Freebase domain. The structure is in its initial state when domain starter nodes are created for each supported domain.

In the proposed model, different types of relations are represented by different edge types. E_{bind} is the edge with label *InterestedIn* and connects a user u to an item i to represent that *"user u is interested in item i"*.

Table 3.2: Thresholds and Functions for hypergraph based User Model

Notation	Description	Type
Υ_{inner}	The semantic relation threshold which defines the enhance limit	Integer
Υ_{domain}	Domain threshold value to decide the number of domain connections	Integer
$f_{ud}(u, d)$	User domain capsule function	Function
$f_{decay}(d, s)$	Profile decay function for domain d and source s	Function
$f_{sim}(i, u, d)$	Similarity function for item and user domain profile	Function
$f_{simUser}(u1, u2, d)$	Similarity function for two users under a domain	Function
$f_{agg}(u, wordList)$	Profile aggregation	Function

In order to model the semantic relations between items, E_{inner} is used and the label of the edge represents the nature of the semantic relation. For instance, in Figure 3.1 *CreatedBy* and *HasGenre* are E_{inner} edges with different semantics.

The item i is connected to its belonging domain d by using E_{domain} edge. In the proposed model, items without any domains are not allowed, every item must be connected to at least one domain starter node.

The friendship between users is represented with E_{friend} edges. E_{inner} and E_{domain} edges enable content-based recommendations where E_{friend} supports collaborative recommendations.

We collect short term profiles for registered users from predefined knowledge sources such as Facebook, Twitter and Linkedin. Besides, we allow users to add their interests manually via an interface. In this thesis, we focus on constructing a holistic, multi-domain user model by aggregating the received short term profiles by utilizing the proposed hypergraph based data structure. We use the term *partial profile* and *short term profile* interchangeably in the thesis.

Definition 3.2. The **hypergraph based user profile** H_u is the aggregated, semantically enhanced user model for the user u (Eqn.3.1). It is the union of the user's friends whom the user follows or is followed by (Eqn. 3.2), the user's explicit profile which is the set of user's declared interested items and their belonging domains (Eqn. 3.3) and the user's semantically enhanced profile (Eqn. 3.4).

The user's enhanced profile is defined as the set of items whose shortest path to the user node has at least *min*, at most *max* steps, and the associated domains of the items.

$$\begin{aligned} H_u\left(u; min; max\right) = U_{friends}(u) & \\ \cup \quad U_{explicitprofile}(u) & \\ \cup \quad U_{enhancedprofile}(u; min, max) & \end{aligned} \tag{3.1}$$

$$\begin{aligned} U_{\mathit{friends}}\left(u\right) = u \xrightarrow{follows} \left(u_f\right) & \\ \cup \quad \left(u_f\right) \xrightarrow{follows} u & \end{aligned} \tag{3.2}$$

$$U_{\mathit{explicit\,profile}}\left(u\right) = u \xrightarrow{interestedIn} (i) \xrightarrow{isInDomain} (d) \tag{3.3}$$

$$\begin{aligned} U_{\mathit{enhanced\,profile}}\left(u; min; max\right) = & \\ u \xrightarrow{*min..max} (i) \xrightarrow{isInDomain} (d) & \end{aligned} \tag{3.4}$$

Basically *the hypergraph based user model* consists of sets of nodes and strongly typed hyperedges. The proposed hypergraph consists of nodes for domains, interest items and users; and edges for explicitly stated interests, semantic relationships between interest items and domain relations of the items.

As an example scenario, assume that there are three users whose names are *GraceKelly*, *IngridBergman* and *TippiHedren*. *IngridBergman* states interest in three items: *Alfred*

Hitchcock who is a director and *Alfred Hitchcock Presents* and *The Twilight Zone* which were popular TV shows in 1950s. *GraceKelly* expresses interest in the director *Alfred Hitchcock* whereas *TippiHedren* does not declare any interest. Also these three users are friends. The hypergraph which models the illustration scenario is in Figure 3.3; for clarity friendships and domains are eliminated. The implementation of this hypergraph actually corresponds to the property graph shown in Figure 3.4.

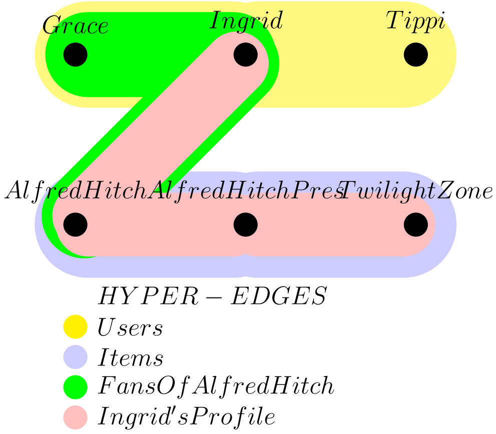


Figure 3.3: Illustration Scenario in Hypergraph

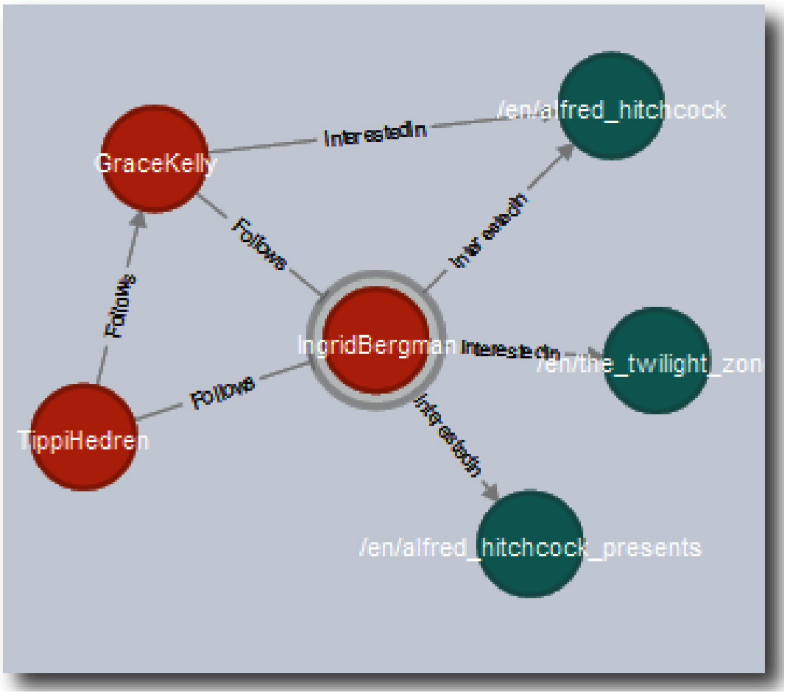


Figure 3.4: Illustration Scenario in Property Graph

In the hypergraph (Figure 3.3), the yellow hyperedge models the set of users, whereas in the property graph (Figure 3.4) the users are represented with red nodes. Similarly, the blue hyperedge in the hypergraph is a wrapper for the set of items where the green nodes in the property graph are item nodes. The pink hyperedge in the hypergraph links *Ingrid* with her declared interested items. In the property graph, this hyperedge is modeled by connecting *Ingrid* to the items with edges of type *InterestedIn*. All users are connected to each other via *following* mechanism to represent their friendship. The type of the edge between users is *Follows* and the type of edge between a user and an explicitly declared item is *InterestedIn*.

3.3.1 Entity Disambiguation

Entity disambiguation is the task of disambiguating keywords and linking them to a knowledge source. When a new keyword expressing the user's interest is considered for aggregation, the keyword is located in the external knowledge base. In this thesis, we use Freebase as the knowledge base and a *disambigation routine* which processes the keyword if the keyword does not match any entity in Freebase.

The disambiguation routine performs several text processing operations. For example it replaces the special characters with the nearest letters in English alphabet such as replacing *ş, ç* by *s, c*; removes the terms such as *"Fans Of", "Quotes"* from the keyword; splits the keyword if it contains characters such as *"&, /"*. Freebase search API returns matching concepts ordered by score, therefore we used the first concept with the highest score as the matching entity for the keyword.

3.3.2 Domain Identification

We defined a *domainizer routine* to assign the disambiguated concept to the domains it belongs. In the proposed model, Freebase domains which corresponds to Freebase commons package is used. The list of domains is presented in Appendix B. For each domain type, a starter domain node is created at system initiation. The type information of the concept is retrieved from Freebase. The retrieved type information not only includes domain knowledge, but also more specific type information. For instance, when the type information of *Alfred Hitchcock* is retrieved, types such as *Film director, Film producer, Film writer* are also retrieved under the type *Film* which is a domain. We exploit those specific types to compute the weight of the domain. In other words, we build a weighted domain structure by accumulating specific types under each domain. For example, in *Alfred Hitchcock* example, the weight of *Film* domain is 3, since this is the sum of subtypes retrieved. Afterwards, we prone the weighted domain structure according to the predefined *domain threshold* and relate the concept with the most frequent domains by using an edge with type *IsInDomain*. In Figure 3.5, the purple nodes represent the domain starter nodes. There is one starter node for each domain and all of the items belonging to that domain is related to that node. This design facilitates domain-based queries.

3.3.3 Semantic Enhancement

Semantic enhancement is the task of enriching the model semantically by retrieving related items. The *semantic enhancement* of a concept is achieved by retrieving predefined *Freebase Metaschema* properties which provide higher order relations between concepts. Metaschema ontology consists of 46 properties and constructs another layer over huge Freebase ontology which has over 3500 properties. Metaschema connects important information and eliminates excessively detailed semantics in Freebase. We further reduced 46 properties to 9 properties by considering their benefits in user modeling and apply a threshold on the number of retrieved relations. The complete list of metaschema properties is given in Appendix A.

The 9 properties we support for semantic enhancement include *BroaderThan/ NarrowerThan, ContributedTo/ HasContributor, Created/ CreatedBy, HasGenre/ GenreOf, HasName/ NameOf, HasChild/ HasParent, PractitionerOf/ HasPractitioner, HasSubject/ SubjectOf, SuperclassOf/ SubclassOf.* Using Freebase over a middle ontology enables writing domain-independent or domain-configured algorithms by using different thresholds for different domains. For instance, *ContributedTo* and *Created* properties reveal important information for *Film* and *Music* domains where *ChildOf* property is meaningful in *People* domain. The concepts retrieved during semantic enhancement are related to the key concept with an edge of type named after the metaschema property linking them. For instance, in Figure 3.5, *Alfred Hitchcock* which is represented by the green node at the center is related to his movies, TV

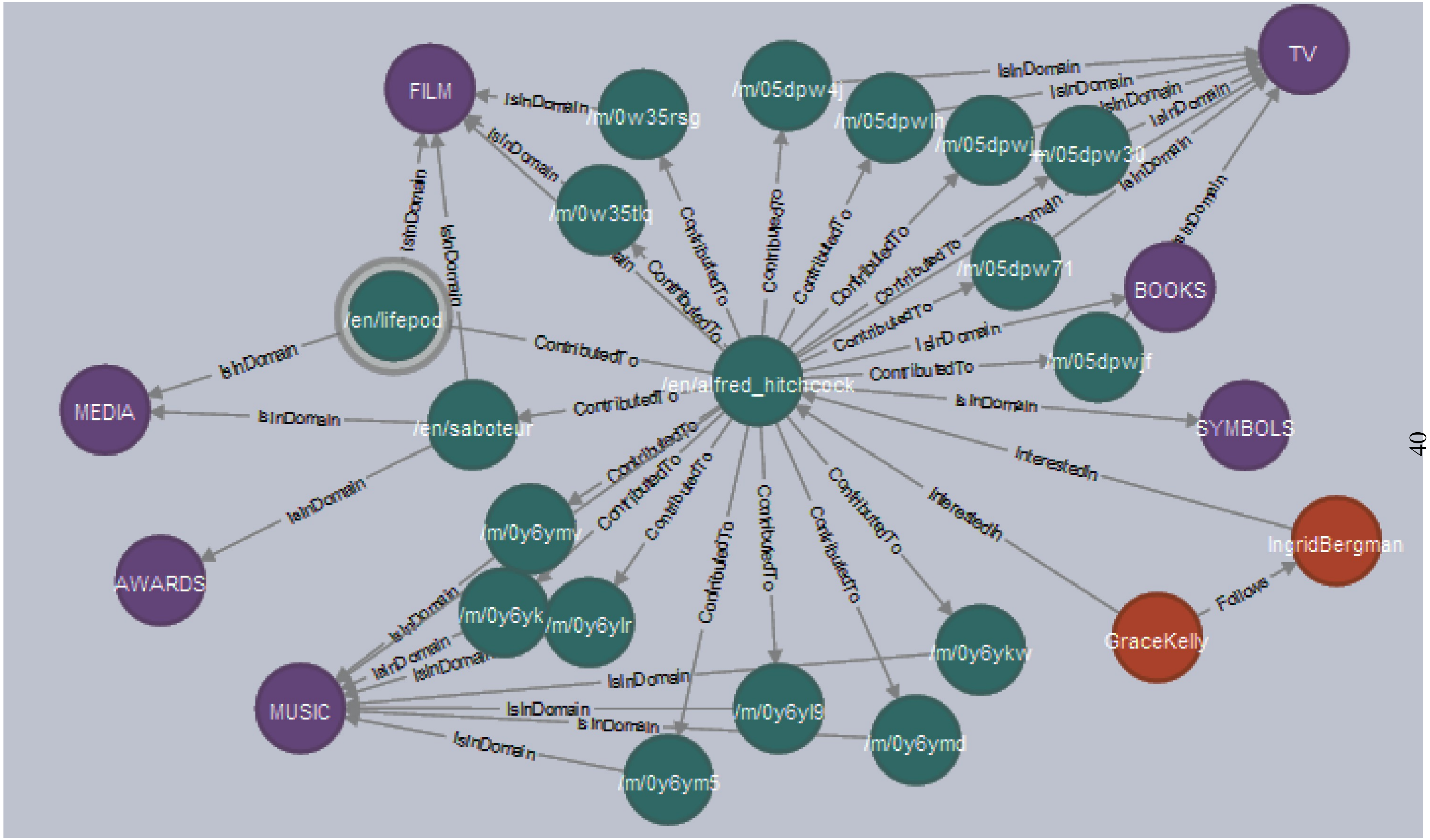


Figure 3.5: A Sample User Model

shows and songs with an edge of type *ContributedTo*.

3.4 User Model Construction

This section presents pseudo codes for the main steps of user model construction including disambiguating entities, identifying domains for the disambiguated entities, semantic enhancement and aggregation of profiles.

3.4.1 Entity Disambiguation Algorithm

When a list of keywords is given as a partial profile, each keyword in the list goes through the aggregation routine. The first step is disambiguating the term by using an external knowledge base, for instance Freebase (Alg. 1). During disambiguation, an MQL JSON query is created for the given keyword and the keyword is searched in Freebase by using Freebase search api. Freebase search api returns results in an array ordered according to the relevance score. Therefore, the first item in the array is taken as the corresponding Freebase item for the given keyword.

If the keyword could not be disambiguated, regional characters are replaced with letters from English alphabet and disambiguation is called for the processed keyword. In some situations such as *"Fans of Roger Federer"*, *"Raising Hope Quotes"*, processing the keyword by removing *"Fans Of"* and *"Quotes"* could succeed.

If the processed keyword could not be disambiguated, the keyword is split and separately disambiguated if it contains *"&"* or *"and"*. The keywords that could not be disambiguated are disregarded. The disambiguated term is added to the user model as item node and it is connected to the user with an E_{bind} edge indicating the user is interested in the item.

Algorithm 1: Disambiguation

```
Result: freebaseData
mqlQuery = makeJsonQuery(keyword)
 retJSONArray = executeFreebaseQuery(mqlQuery)
 freebaseData ← first of retJSONArray if freebaseData == null then
    keyword ← keyword.replace(ğ, g) keyword ← keyword.replace(ş, s) ...
     freebaseData = disambiguate(keyword) if freebaseData == null then
        keyword ← keyword.replace("Fans of", "")
         keyword ← keyword.replace("Fan Club", "") ...
         freebaseData = disambiguate(keyword) if freebaseData == null
         then
            freebaseData = disSplit("&", keyword)
             freebaseData = disSplit("and", keyword) ...
        end
    end
end
```

3.4.2 Domain Identification Algorithm

The second step after disambiguation is deciding domains for the disambiguated item (Alg. 2). A Freebase mqlread api call returns types of the disambiguated term. For each type of the item, a domain map which keeps track of the domain frequencies for the item is used. Afterwards, pruning is applied by connecting the item to the most frequent Υ_{domain} domains in the domain map. The item is connected to its belonging domains with E_{domain} edges.

```
Algorithm 2: Decide Domains
  Result: domainMap
  mqlQuery = makeJsonQuery(freebaseID)
  retJSON = executeFreebaseQuery(mqlQuery)
  typeArray ← type property of(retJSON) foreach type in typeArray do
      domainType = convert2DomainType(type)
      add to domain map,
      increment frequency if already exists domainMap.Add(domainType)
        domainMap = pruneDomMap($domainMap$)
  end
```

3.4.3 Semantic Enhancement Algorithm

The third major step in the lifecycle of the given keyword is semantically enhancing the item (Alg. 3). During semantic enhancement, we use the reduced set of Freebase *metaschema properties* stated above.

Using Freebase over a middle ontology enables writing domain-independent or domain-configured algorithms by using different thresholds for different domains. $\Upsilon_{enhance}$ properties are taken into account for semantic enhancement and the rest is ignored. In the user model, each semantic enhancement item is added to the hypergraph and connected to the item for the given keyword with an E_{inner} edge named after the metaschema property between them.

```
Algorithm 3: Enhance
  Result: metaschemaList
  mqlQuery = makeJsonQuery(freebaseID)
  retJSONArray = executeMetaschemaQuery(mqlQuery)
  retJSONArray ← limited(retJSONArray)
  metaschemaList ← parsed JSON(retJSONArray)
```

3.4.4 User Profile Aggregation

In order to aggregate a keyword to the user profile, basicly, the keyword is disambiguated, the disambiguated item is connected to the user and its domains and it is semantically enhanced. For each different keyword-knowledge source pair, the frequency of the edge between the user and the item is incremented. For instance, if the user's keyword *SOA* comes from two partial profiles whose knowledge sources are Facebook and LinkedIn, its frequency is 2. If the same item is disambiguated from different keywords from the same knowledge source such as *SOA* and *Service Oriented Architecture* from LinkedIn profile, its frequency is 2. However, when the same keyword comes from the same knowledge source, we disregard the duplicate of the keyword. In other words, only different keywords for the same semantic item or same keyword from different knowledge sources affect the frequency of the user's interest on the item.

The proposed aggregated user profile [114, 112, 115, 116, 113] is capable of supporting several user modeling domain problems that can be solved by providing traversal algorithms on graph such as recommendation. In the next chapter, we provide a sample recommender system which exploits the proposed framework. Moreover, the framework could also provide domain based or general user models to external personalization services. Besides, the model is able to extract enriched partial profiles for the needs of any application. Once the external application specifies the traversal that it needs for its specific query, it can employ our user modeling framework.

Algorithm 4: Aggregation

```
Result: Aggregated frofile
foreach keyWord in keyWordList do
    freebaseData = disambiguate(keyword)
     freebaseID $\leftarrow$ freebaseData.freebaseID if freebaseID in Hypergraph then
        if freebaseID already connected to User then
            increment frequency
        end
        connect freebaseID to the user
    end
    decideDomains (freebaseID ) enhance (freebaseID )
end
```

CHAPTER 4

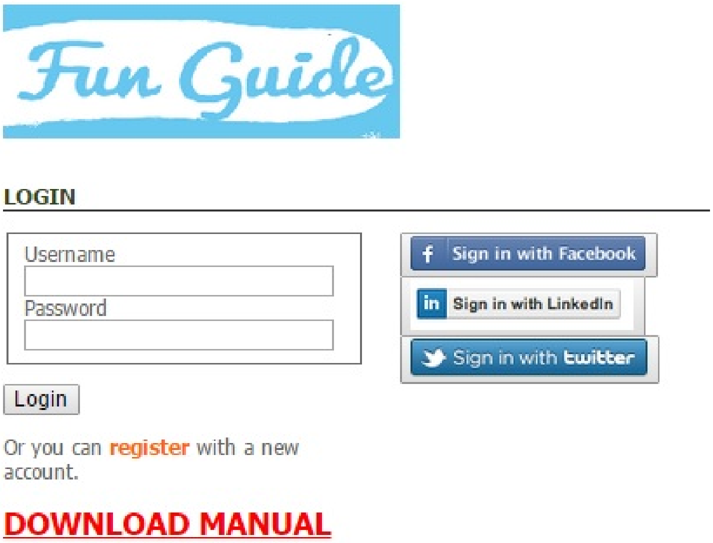
EMPLOYMENT OF THE HYPERGRAPH BASED MODELING FRAMEWORK FOR A RECOMMENDER SYSTEM

In this chapter, employment of the proposed hypergraph based modeling framework for a recommender system is introduced. The case study is designed as a web site named *FunGuide*. Various connection-based queries could be answered by defining traversals on the proposed hypergraph based data structure. The case study illustrates extraction of partial profiles, aggregation of profiles and domain-based and cross-domain recommendations. The system is also capable of discovering users who might be interested in a given item and finding similar users in terms of interests.

4.1 FunGuide Overview

FunGuide enables users to register and connect with each other. The system enables the user to *Login with Facebook* as in Figure 4.1 and imports his/her Facebook profile item by item using the proposed profile aggregation methodology. Similarly, the system provides *Login with LinkedIn* and *Login with Twitter* buttons to extract and aggregate partial profiles from LinkedIn and Twitter.

When the user logins with all social accounts, partial profiles from these social web accounts are extracted and aggregated into one holistic semantic user profile. Figure 4.2 shows a semantic user profile which contains 30 profile items. The profile items are ordered by frequency, then by alphabetically. The first profile item which is *News Satire*[*MEDIA,TV*] [*media genre, TV subject, TV genre*] (*Frequency:2*) shows that the user likes fake news and the domains that the profile item belong are classified as



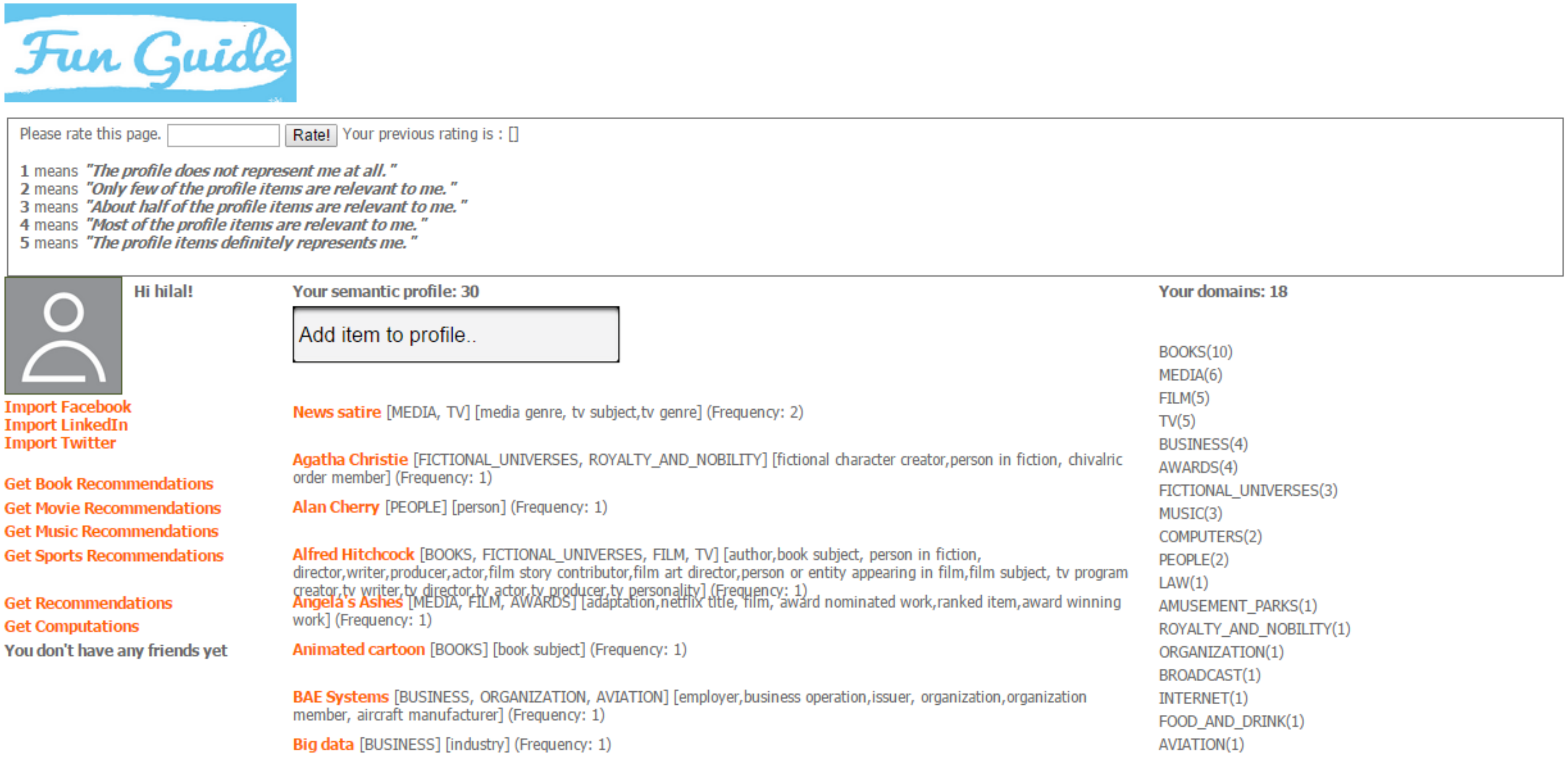
Or you can register with a new account.

Figure 4.1: Fun Guide - SignIn

MEDIA and TV. Since FunGuide is capable of providing domain-based recommendations, we also keep track of the secondary domain information about profile items. In this example, fake news profile item is a media genre, a TV subject and a TV genre. The frequency has a higher value as the number of partial profiles which supports the profile item increases. If the exact keywords comes from the same knowledge source, it does not affect the frequency. However, if another keyword mapping to the same entity comes, frequency is increased. In this case, two of the partial profiles show that the user is interested in fake news. When the profile item is supported by the same partial profile with different proofs, frequency is also increased. For instance, if the user states that he/she likes *Zaytung* and *ResmiGaste*, which are both fake news websites, in Facebook profile, the frequency is 2. As the time passes, the frequency resulted from a proof decays by a factor.

FunGuide shows the domain distribution for the user's profile as in Figure 4.2. The domains that the user is interested in are ordered according to their weight. Domain distribution could be considered as a user profile in very high granularity. For instance, in the example case the user is mainly interested in books, media, film and TV.

The proposed case study provides domain based recommendations for book, movie, music and sports domains besides supporting cross domain recommendations. The



Please rate this page. Rate! Your previous rating is : []

1 means *"The profile does not represent me at all."*
2 means *"Only few of the profile items are relevant to me."*
3 means *"About half of the profile items are relevant to me."*
4 means *"Most of the profile items are relevant to me."*
5 means *"The profile items definitely represents me."*

Hi hilal!

Import Facebook
Import LinkedIn
Import Twitter

Get Book Recommendations
Get Movie Recommendations
Get Music Recommendations
Get Sports Recommendations

Get Recommendations
Get Computations
You don't have any friends yet

Your semantic profile: 30

Add item to profile..

News satire [MEDIA, TV] [media genre, tv subject,tv genre] (Frequency: 2)

Agatha Christie [FICTIONAL_UNIVERSES, ROYALTY_AND_NOBILITY] [fictional character creator,person in fiction, chivalric order member] (Frequency: 1)

Alan Cherry [PEOPLE] [person] (Frequency: 1)

Alfred Hitchcock [BOOKS, FICTIONAL_UNIVERSES, FILM, TV] [author,book subject, person in fiction, director,writer,producer,actor,film story contributor,film art director,person or entity appearing in film,film subject, tv program creator,tv writer,tv director,tv actor,tv producer,tv personality] (Frequency: 1)

Angela's Ashes [MEDIA, FILM, AWARDS] [adaptation,netflix title, film, award nominated work,ranked item,award winning work] (Frequency: 1)

Animated cartoon [BOOKS] [book subject] (Frequency: 1)

BAE Systems [BUSINESS, ORGANIZATION, AVIATION] [employer,business operation,issuer, organization,organization member, aircraft manufacturer] (Frequency: 1)

Big data [BUSINESS] [industry] (Frequency: 1)

Your domains: 18

BOOKS(10)
MEDIA(6)
FILM(5)
TV(5)
BUSINESS(4)
AWARDS(4)
FICTIONAL_UNIVERSES(3)
MUSIC(3)
COMPUTERS(2)
PEOPLE(2)
LAW(1)
AMUSEMENT_PARKS(1)
ROYALTY_AND_NOBILITY(1)
ORGANIZATION(1)
BROADCAST(1)
INTERNET(1)
FOOD_AND_DRINK(1)
AVIATION(1)

Figure 4.2: Fun Guide - User Profile

system is also capable of answering some other user modeling domain queries. The system is easily extendible to support domain based recommendations in other domains as well. FunGuide is capable of supporting many user modeling domain problems.

4.2 Implementation Details

FunGuide is written in Java using Eclipse as IDE. Bitbucket is used as version tracking system. The system uses Neo4j which is a graph database that uses property graphs as graph data model. Since Neo4j graph database is used, the queries are written in *Cypher*, which is a pattern-matching language that helps to describe graphs using diagrams [92].

Cypher is composed of clauses, mainly *START*, *MATCH* and *RETURN* clauses. *START* clause specifies one or more starting points in the property graph. The starting point could be a node or a relationship. *MATCH* clause is the specification by example part of the query. *RETURN* clause defines the nodes, relationships, and properties in the matched data that are going to be returned as the result set of the query.

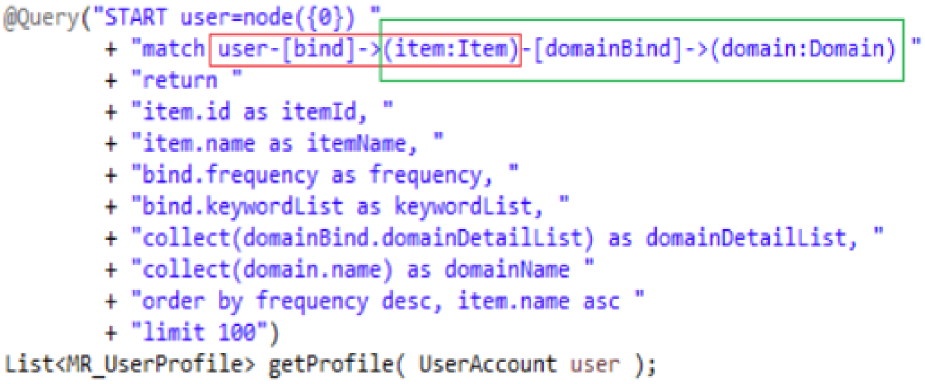
In the notation, nodes are represented by parentheses and relationships are denoted by using –> and <– signs indicating direction of the relation. Name of the relationship could be defined inside the relation signs as -[:<*relation name*>]->. For instance (Grace)[:FOLLOWS]->(Tippi) states that *Grace follows Tippi*.

4.3 Query: Semantic User Model

The proposed system is able to extract domain-based or general semantic profile of the user. In order to obtain the *domain-based user model* for user u and domain d, the user is located in the external index system for users and the user node in the hypergraph based structure is reached with a *short-cut*. Eqn. 4.1 computes domain-based user model by matching the items which are in domain *d* and have a shortest path with the user *u* with length at most *max*.

$$P_{domain}(u; d; max) = u \xrightarrow{*0..max} (i) \xrightarrow{IsInDomain} d \qquad (4.1)$$

The corresponding Cypher query is displayed in Figure 4.3. In the query, the red frame locates the items that are attached to the user and the green frame retrieves the domains of these items.



```
@Query("START user=node({0}) "
        + "match user-[bind]->(item:Item)-[domainBind]->(domain:Domain) "
        + "return "
        + "item.id as itemId, "
        + "item.name as itemName, "
        + "bind.frequency as frequency, "
        + "bind.keywordList as keywordList, "
        + "collect(domainBind.domainDetailList) as domainDetailList, "
        + "collect(domain.name) as domainName "
        + "order by frequency desc, item.name asc "
        + "limit 100")
List<MR_UserProfile> getProfile( UserAccount user );
```

Figure 4.3: Cyper Query - Semantic User Model

The json output for the query *"Retrieve the domain based profile for user GraceKelly for TV domain."* is as follows:

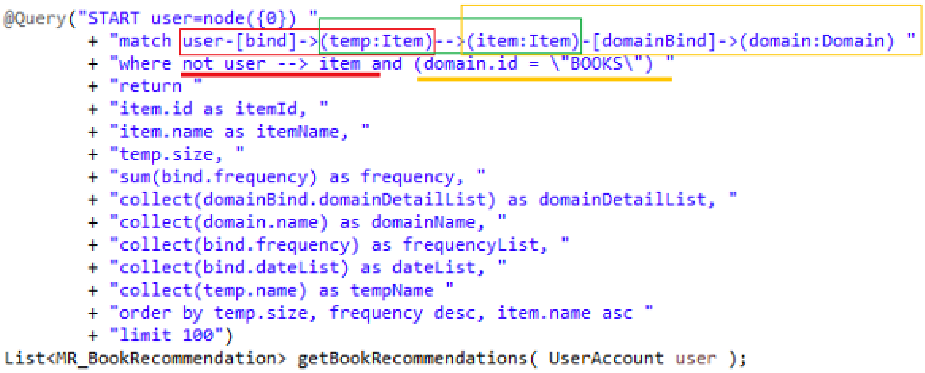
```
{ "data": [
    { "row": [
        "GraceKelly",
        "Alfred Hitchcock"
      ] },
    { "row": [
        "GraceKelly",
        "Alfred Hitchcock Presents"
      ] },
    { "row": [
        "GraceKelly",
        "The Case of Mr. Pelham"
      ] },
    ...
  ] }
```

According to the json output, the result set contains the user's declared interest *Alfred Hithcock* and the items in her enhanced profile such as the TV show *Alfred Hitchcock Presents* and its several episodes. To obtain the *general user profile*, domain is not included as a parameter to the traversal function (Eqn. 4.2).

$$P_{general}\left(u; max\right) = u \xrightarrow{*0..max} (i) \tag{4.2}$$

4.4 Query: Domain Based Recommendation

The system is capable of imposing domain to queries. For instance, Cypher query for getting book recommendations is displayed in Figure 4.4. The book recommendation interface is displayed in Figure 4.5. This is also an example for cross-domain recommendation, since user's profile in TV domain results in recommendations in book domain. For instance, user's interest in Alfred Hitchcock results in suggestion of a book about Hollywood directors including Hitchcock.



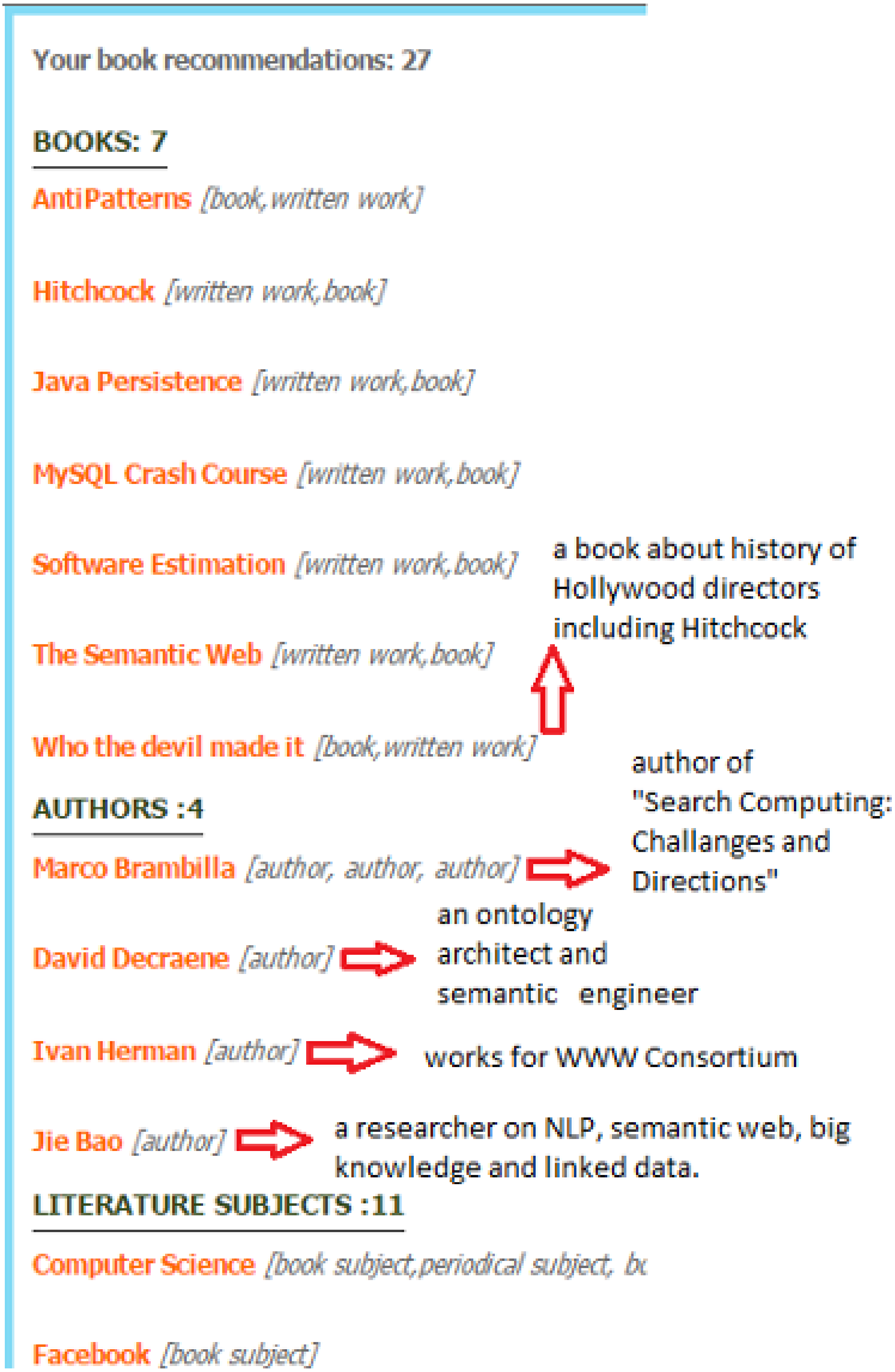
```
@Query("START user=node({0}) "
        + "match user-[bind]->(temp:Item)-->(item:Item)-[domainBind]->(domain:Domain) "
        + "where not user --> item and (domain.id = \"BOOKS\") "
        + "return "
        + "item.id as itemId, "
        + "item.name as itemName, "
        + "temp.size, "
        + "sum(bind.frequency) as frequency, "
        + "collect(domainBind.domainDetailList) as domainDetailList, "
        + "collect(domain.name) as domainName, "
        + "collect(bind.frequency) as frequencyList, "
        + "collect(bind.dateList) as dateList, "
        + "collect(temp.name) as tempName "
        + "order by temp.size, frequency desc, item.name asc "
        + "limit 100")
List<MR_BookRecommendation> getBookRecommendations( UserAccount user );
```

Figure 4.4: Cyper Query - Book Recommendation

4.5 Query: Discovering Potential Users Who Are Interested in a Domain or an Item

In order to discover the users interested in a *domain* d, the set of users that have shortest path with length at most *max* to d are retrieved (Eqn. 4.3).

$$U_{domain}\left(d; max\right) = d \leftarrow (i) \xleftarrow{*0..max} (u) \tag{4.3}$$



Your book recommendations: 27

BOOKS: 7

AntiPatterns [book,written work]

Hitchcock [written work,book]

Java Persistence [written work,book]

MySQL Crash Course [written work,book]

Software Estimation [written work,book]

The Semantic Web [written work,book]

Who the devil made it [book,written work]

a book about history of Hollywood directors including Hitchcock

AUTHORS :4

Marco Brambilla [author, author, author]

author of "Search Computing: Challanges and Directions"

David Decraene [author]

an ontology architect and semantic engineer

Ivan Herman [author]

works for WWW Consortium

Jie Bao [author]

a researcher on NLP, semantic web, big knowledge and linked data.

LITERATURE SUBJECTS :11

Computer Science [book subject,periodical subject, bc

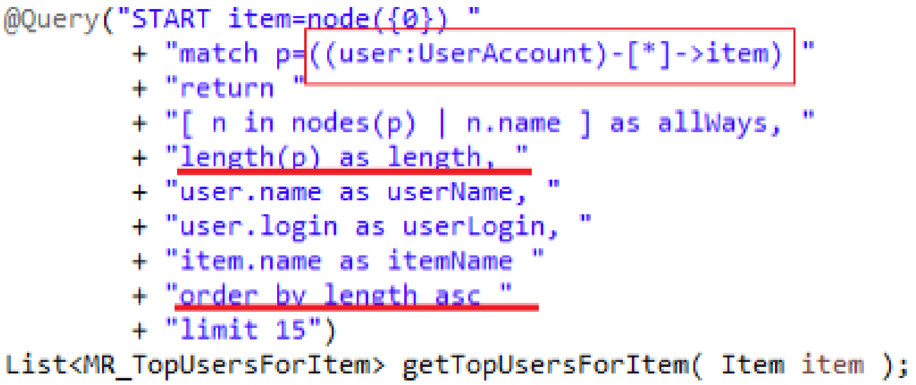
Facebook [book subject]

Figure 4.5: Fun Guide - Book Recommendations

As another query, to discover users interested in an *item* i, the set of users that have shortest path with length at most *max* to *i* are retrieved (Eqn. 4.4).

$$U_{item}\left(i; max\right) = i \xleftarrow{*0..max} \left(u\right) \tag{4.4}$$

The cypher query is given in Figure 4.6. The cypher query to compute the user's interest for an item is given in Figure 4.8 and the user interface is in 4.7.



```
@Query("START item=node({0}) "
        + "match p=((user:UserAccount)-[*]->item) "
        + "return "
        + "[ n in nodes(p) | n.name ] as allWays, "
        + "length(p) as length, "
        + "user.name as userName, "
        + "user.login as userLogin, "
        + "item.name as itemName "
        + "order by length asc "
        + "limit 15")
List<MR_TopUsersForItem> getTopUsersForItem( Item item );
```

Figure 4.6: Cyper Query - Discovering Potential Users Who Are Interested in an Item

4.6 Query: Cross-Domain Recommendation

The ability to discover related concepts of an item i in other domains as in Eqn. 4.5 enables answering questions such as *"What are the films about Nasa?"* or *"Find biographies about Mozart."*.

$$\begin{aligned} R_i\left(i; max\right) = i & \xrightarrow{IsInDomain} \left(d_1\right) \\ and \quad i & \xrightarrow{[*2..max]} \left(d_2\right) \\ and \quad & \left(otherItem\right) \rightarrow d_2 \\ and \quad & d_1 \neq d_2 \end{aligned} \tag{4.5}$$

4.7 Query: Discovering Similar Users

In order to calculate a user's interest on an item, shortest path algorithms could be applied as in Eqn. 4.6.

$$I_{\mathit{interest}}\,(u;i) = shortestPath(u,i) \tag{4.6}$$

The cypher query for discovering similar users is in Figure 4.9 and the interface is in Figure 4.7.

4.8 General Recommendation

FunGuide has an integrated interface which is dedicated for recommendation. Figure 4.10 shows the interface of the system that we implemented based on these traversal algorithms. In the illustration scenario (Figure 3.4), *GraceKelly* declared one interest item: director *Alfred Hitchcock*.

The integrated interface is divided into six columns. The first column shows the friendship information, the second column enables manual addition of an interest item and shows the user's declared interests. The number next to the declared interest is the frequency of that item and it is incremented by one whenever the same concept is matched with different keyword-information source pairs. The list next to the frequency information shows the domains of the item. The third column exposes the domain aggregation for the user. The fourth and fifth columns show the top 15 recommendations for the user.

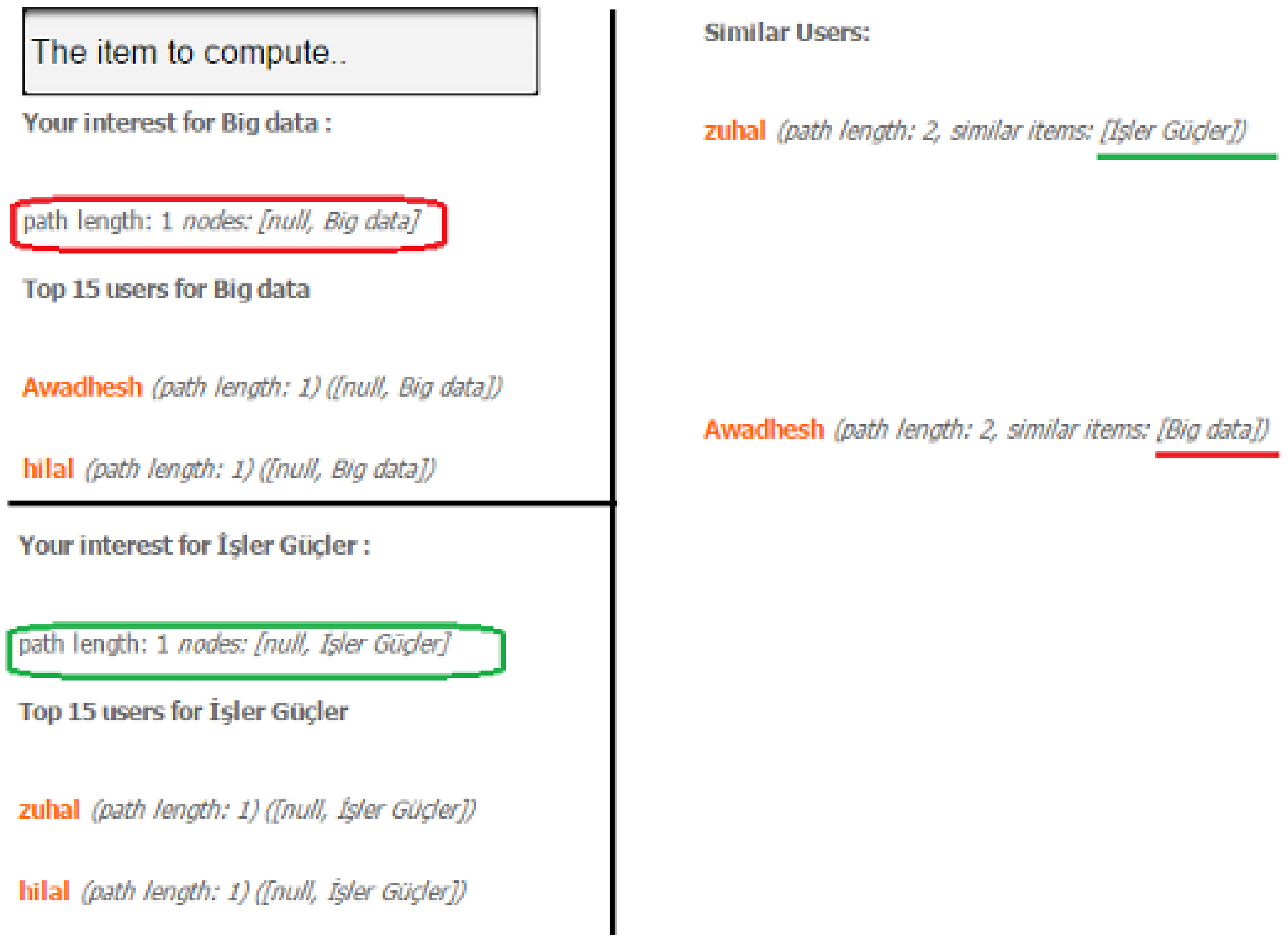
Random recommendations part recommends any item which is connected to the user in the graph via other items or users. *Detailed recommendations* part recommends items that are connected to the user's declared items and ranks the recommendation by checking two factors: the number of declared items of the user which constitute a path of length 2 between the user and the recommended item and the accumulated frequency of the items in that path. For instance, there are two paths of length 2 between *IngridBergman* and *Mystery* item over the user's two declared interests: *The Twilight Zone* and *Alfred Hitchcock Presents*. Since both items are assigned frequency

1, the accumulated frequency is 2.

In Figure 4.11, the *Horror, Anthology* and *Mystery* are recommended because of two declared interests: *The Twilight Zone* and *Alfred Hitchcock Presents* and the accumulated frequency is 2, each declared item has frequency 1.

Popular recommendations part recommends items only in popular domains and eliminates other domains. Path length ordering is applied. *Far recommendations* part recommends items at least three, at most five steps away from the user. The sixth column computes whether the user is interested in the specified item and lists the users who might be interested in. For instance, in Figure 4.10, *GraceKelly*'s interest for *Marnie*, which is a movie directed by Alfred Hitchcock, is over declared interest *Alfred Hitchcock* and the path length is 2.

In Figure 4.12,*TippiHedren*'s interest for *Marnie* has a longer path: *TippiHedren* is friends with *GraceKelly*; *GraceKelly* is interested in *Alfred Hitchcock* and *Alfred Hitchcock* contributed to *Marnie*. *TippiHedren* collaboratively gets recommendations although she has not declared any interests.



The item to compute..

Your interest for Big data :

path length: 1 *nodes: [null, Big data]*

Top 15 users for Big data

Awadhesh *(path length: 1) ([null, Big data])*

hilal *(path length: 1) ([null, Big data])*

Your interest for İşler Güçler :

path length: 1 *nodes: [null, Işler Güçler]*

Top 15 users for İşler Güçler

zuhal *(path length: 1) ([null, İşler Güçler])*

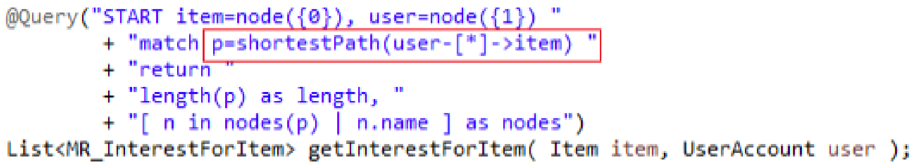
hilal *(path length: 1) ([null, İşler Güçler])*

Similar Users:

zuhal *(path length: 2, similar items: [Işler Güçler])*

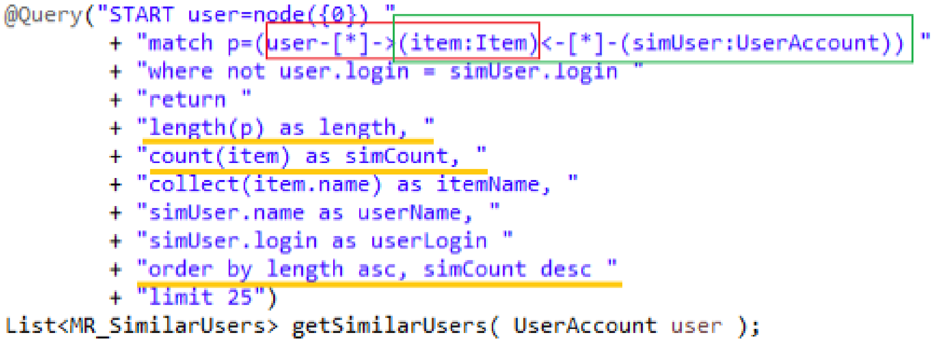
Awadhesh *(path length: 2, similar items: [Big data])*

Figure 4.7: Fun Guide - Computation Interface



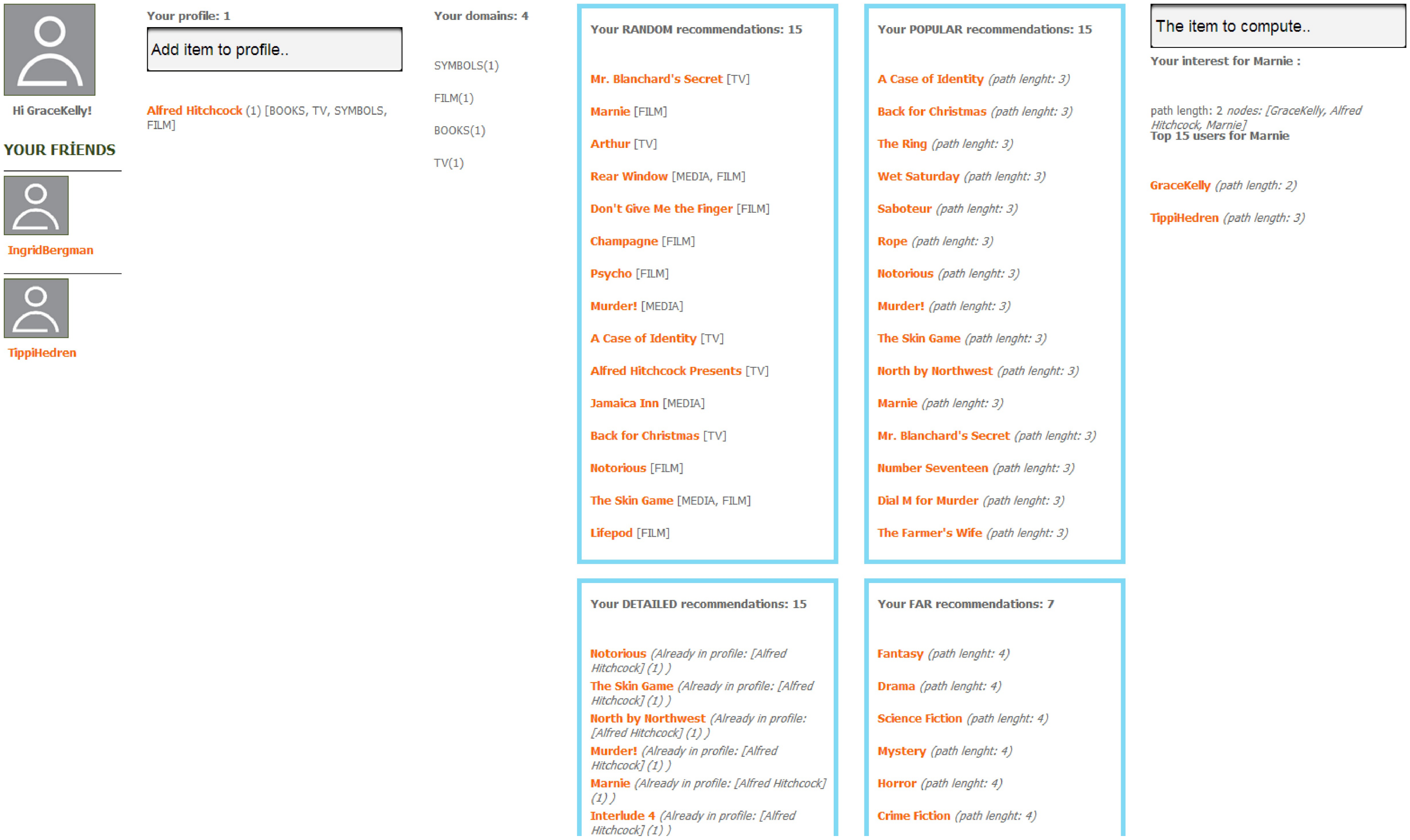
```
@Query("START item=node({0}), user=node({1}) "
        + "match p=shortestPath(user-[*]->item) "
        + "return "
        + "length(p) as length, "
        + "[ n in nodes(p) | n.name ] as nodes")
List<MR_InterestForItem> getInterestForItem( Item item, UserAccount user );
```

Figure 4.8: Cyper Query - Compute User’s Interest For an Item



```
@Query("START user=node({0}) "
        + "match p=(user-[*]->(item:Item)<-[*]-(simUser:UserAccount)) "
        + "where not user.login = simUser.login "
        + "return "
        + "length(p) as length, "
        + "count(item) as simCount, "
        + "collect(item.name) as itemName, "
        + "simUser.name as userName, "
        + "simUser.login as userLogin "
        + "order by length asc, simCount desc "
        + "limit 25")
List<MR_SimilarUsers> getSimilarUsers( UserAccount user );
```

Figure 4.9: Cyper Query - Discovering Similar Users



Hi GraceKelly!

YOUR FRIENDS

IngridBergman

TippiHedren

Your profile: 1

Add item to profile..

Alfred Hitchcock (1) [BOOKS, TV, SYMBOLS, FILM]

Your domains: 4

SYMBOLS(1)

FILM(1)

BOOKS(1)

TV(1)

Your RANDOM recommendations: 15

Mr. Blanchard's Secret [TV]

Marnie [FILM]

Arthur [TV]

Rear Window [MEDIA, FILM]

Don't Give Me the Finger [FILM]

Champagne [FILM]

Psycho [FILM]

Murder! [MEDIA]

A Case of Identity [TV]

Alfred Hitchcock Presents [TV]

Jamaica Inn [MEDIA]

Back for Christmas [TV]

Notorious [FILM]

The Skin Game [MEDIA, FILM]

Lifepod [FILM]

Your POPULAR recommendations: 15

A Case of Identity *(path lenght: 3)*

Back for Christmas *(path lenght: 3)*

The Ring *(path lenght: 3)*

Wet Saturday *(path lenght: 3)*

Saboteur *(path lenght: 3)*

Rope *(path lenght: 3)*

Notorious *(path lenght: 3)*

Murder! *(path lenght: 3)*

The Skin Game *(path lenght: 3)*

North by Northwest *(path lenght: 3)*

Marnie *(path lenght: 3)*

Mr. Blanchard's Secret *(path lenght: 3)*

Number Seventeen *(path lenght: 3)*

Dial M for Murder *(path lenght: 3)*

The Farmer's Wife *(path lenght: 3)*

The item to compute..

Your interest for Marnie :

path length: 2 *nodes: [GraceKelly, Alfred Hitchcock, Marnie]*
Top 15 users for Marnie

GraceKelly *(path length: 2)*

TippiHedren *(path length: 3)*

Your DETAILED recommendations: 15

Notorious *(Already in profile: [Alfred Hitchcock] (1))*
The Skin Game *(Already in profile: [Alfred Hitchcock] (1))*
North by Northwest *(Already in profile: [Alfred Hitchcock] (1))*
Murder! *(Already in profile: [Alfred Hitchcock] (1))*
Marnie *(Already in profile: [Alfred Hitchcock] (1))*
Interlude 4 *(Already in profile: [Alfred Hitchcock] (1))*

Your FAR recommendations: 7

Fantasy *(path lenght: 4)*

Drama *(path lenght: 4)*

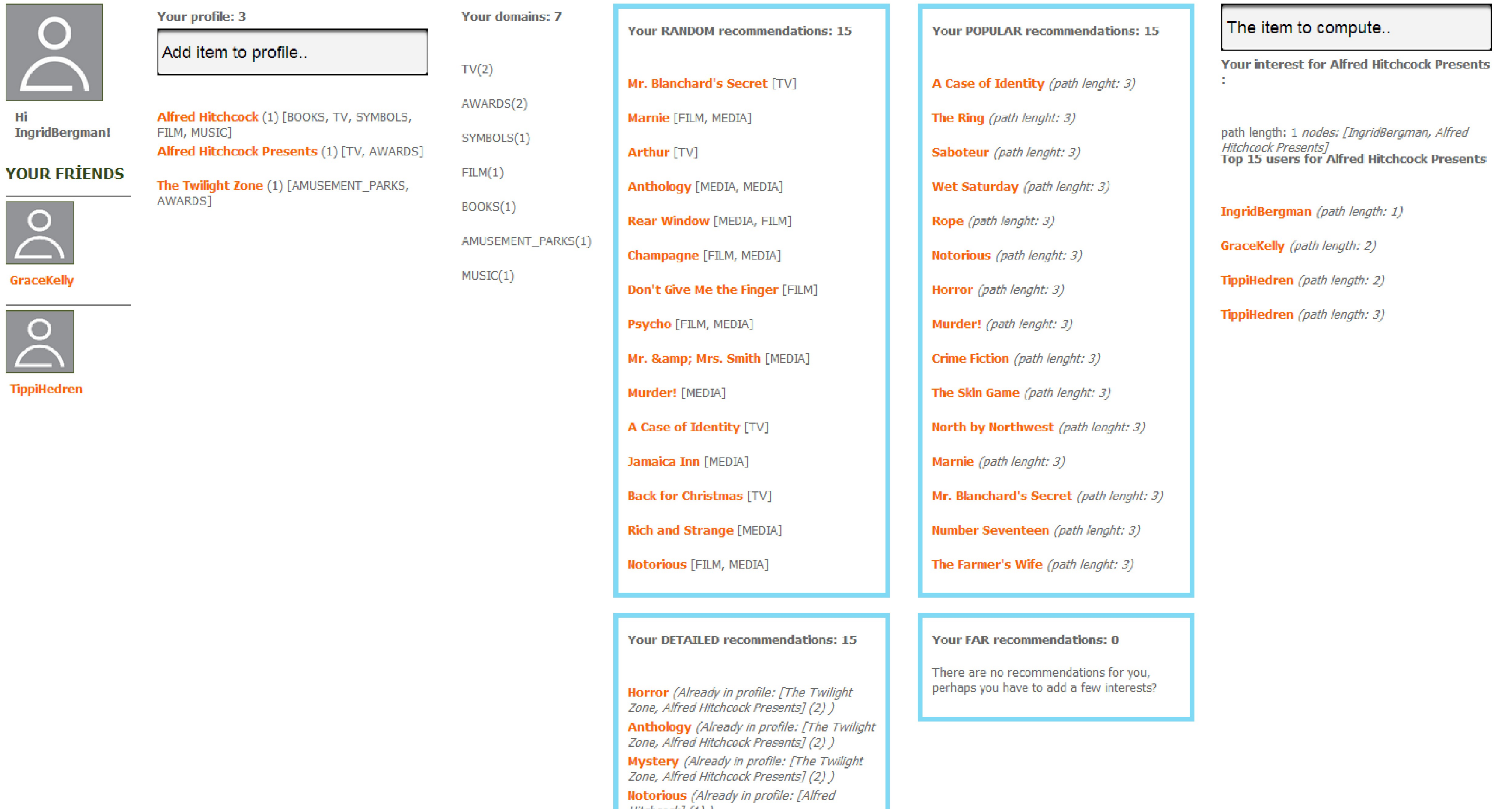
Science Fiction *(path lenght: 4)*

Mystery *(path lenght: 4)*

Horror *(path lenght: 4)*

Crime Fiction *(path lenght: 4)*

Figure 4.10: Fun Guide Interface - Grace Kelly



Hi IngridBergman!

YOUR FRIENDS

GraceKelly

TippiHedren

Your profile: 3

Add item to profile..

Alfred Hitchcock (1) [BOOKS, TV, SYMBOLS, FILM, MUSIC]
Alfred Hitchcock Presents (1) [TV, AWARDS]

The Twilight Zone (1) [AMUSEMENT_PARKS, AWARDS]

Your domains: 7

TV(2)

AWARDS(2)

SYMBOLS(1)

FILM(1)

BOOKS(1)

AMUSEMENT_PARKS(1)

MUSIC(1)

Your RANDOM recommendations: 15

Mr. Blanchard's Secret [TV]

Marnie [FILM, MEDIA]

Arthur [TV]

Anthology [MEDIA, MEDIA]

Rear Window [MEDIA, FILM]

Champagne [FILM, MEDIA]

Don't Give Me the Finger [FILM]

Psycho [FILM, MEDIA]

Mr. &amp; Mrs. Smith [MEDIA]

Murder! [MEDIA]

A Case of Identity [TV]

Jamaica Inn [MEDIA]

Back for Christmas [TV]

Rich and Strange [MEDIA]

Notorious [FILM, MEDIA]

Your DETAILED recommendations: 15

Horror *(Already in profile: [The Twilight Zone, Alfred Hitchcock Presents] (2))*
Anthology *(Already in profile: [The Twilight Zone, Alfred Hitchcock Presents] (2))*
Mystery *(Already in profile: [The Twilight Zone, Alfred Hitchcock Presents] (2))*
Notorious *(Already in profile: [Alfred*

Your POPULAR recommendations: 15

A Case of Identity *(path lenght: 3)*

The Ring *(path lenght: 3)*

Saboteur *(path lenght: 3)*

Wet Saturday *(path lenght: 3)*

Rope *(path lenght: 3)*

Notorious *(path lenght: 3)*

Horror *(path lenght: 3)*

Murder! *(path lenght: 3)*

Crime Fiction *(path lenght: 3)*

The Skin Game *(path lenght: 3)*

North by Northwest *(path lenght: 3)*

Marnie *(path lenght: 3)*

Mr. Blanchard's Secret *(path lenght: 3)*

Number Seventeen *(path lenght: 3)*

The Farmer's Wife *(path lenght: 3)*

Your FAR recommendations: 0

There are no recommendations for you, perhaps you have to add a few interests?

The item to compute..

Your interest for Alfred Hitchcock Presents :

path length: 1 *nodes: [IngridBergman, Alfred Hitchcock Presents]*
Top 15 users for Alfred Hitchcock Presents

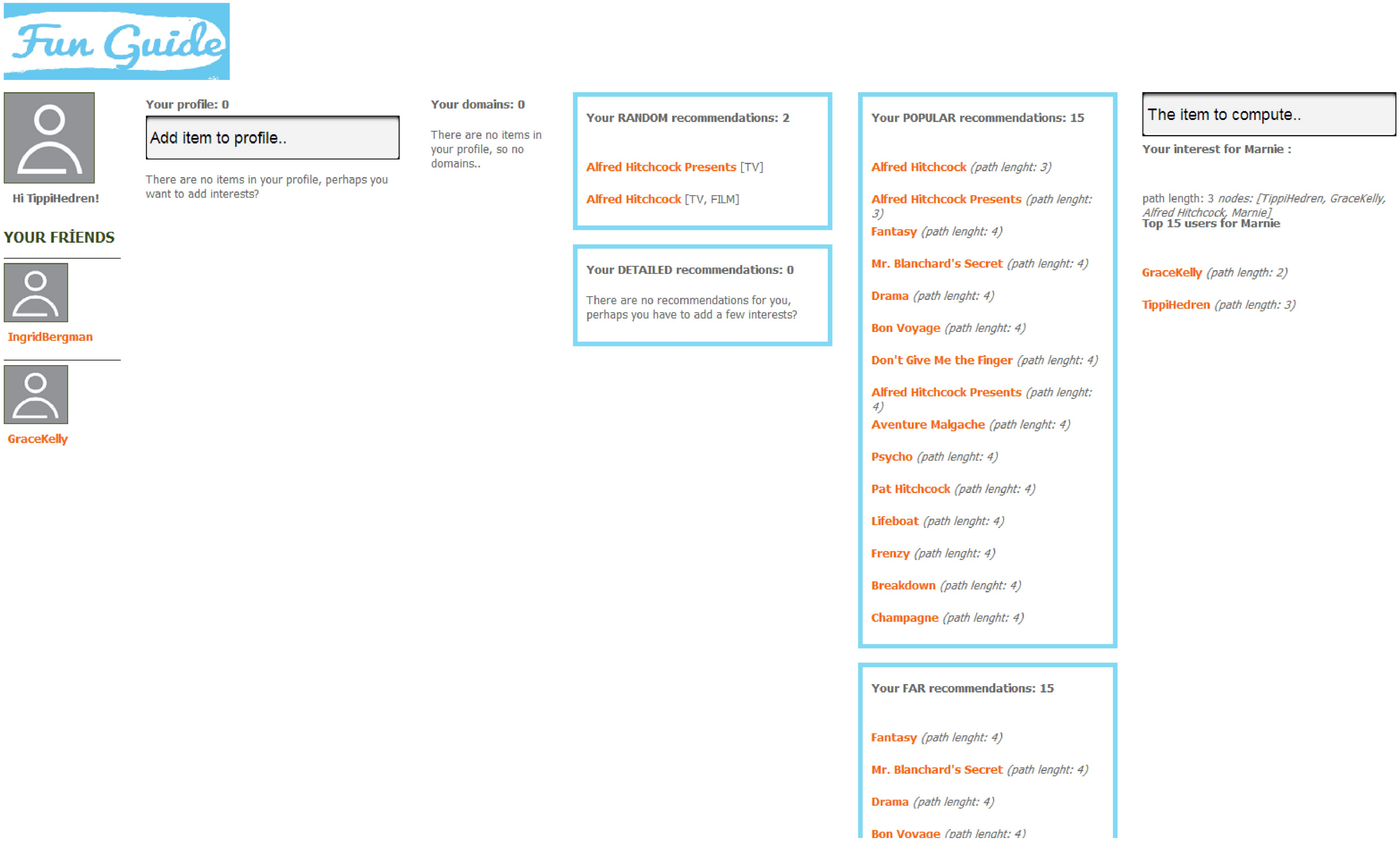
IngridBergman *(path length: 1)*

GraceKelly *(path length: 2)*

TippiHedren *(path length: 2)*

TippiHedren *(path length: 3)*

Figure 4.11: Fun Guide Interface - Ingrid Bergman



Hi TippiHedren!

YOUR FRİENDS

IngridBergman

GraceKelly

Your profile: 0

Add item to profile..

There are no items in your profile, perhaps you want to add interests?

Your domains: 0

There are no items in your profile, so no domains..

Your RANDOM recommendations: 2

Alfred Hitchcock Presents [TV]

Alfred Hitchcock [TV, FILM]

Your DETAILED recommendations: 0

There are no recommendations for you, perhaps you have to add a few interests?

Your POPULAR recommendations: 15

Alfred Hitchcock *(path lenght: 3)*

Alfred Hitchcock Presents *(path lenght: 3)*

Fantasy *(path lenght: 4)*

Mr. Blanchard's Secret *(path lenght: 4)*

Drama *(path lenght: 4)*

Bon Voyage *(path lenght: 4)*

Don't Give Me the Finger *(path lenght: 4)*

Alfred Hitchcock Presents *(path lenght: 4)*

Aventure Malgache *(path lenght: 4)*

Psycho *(path lenght: 4)*

Pat Hitchcock *(path lenght: 4)*

Lifeboat *(path lenght: 4)*

Frenzy *(path lenght: 4)*

Breakdown *(path lenght: 4)*

Champagne *(path lenght: 4)*

Your FAR recommendations: 15

Fantasy *(path lenght: 4)*

Mr. Blanchard's Secret *(path lenght: 4)*

Drama *(path lenght: 4)*

Bon Voyage *(path lenght: 4)*

The item to compute..

Your interest for Marnie :

path length: 3 *nodes: [TippiHedren, GraceKelly, Alfred Hitchcock, Marnie]*
Top 15 users for Marnie

GraceKelly *(path length: 2)*

TippiHedren *(path length: 3)*

Figure 4.12: Fun Guide Interface - Tippi Hedren

CHAPTER 5

PROFILE AGGREGATION: EVALUATION AND DISCUSSION

5.1 Evaluation

The user model is evaluated against various datasets and the results showed that the proposed framework improves results in each dataset. In this chapter, we introduce the datasets, methodology and results of the evaluation.

5.1.1 Evaluation Datasets

The proposed user model aggregates partial profiles and a holistic semantic user model is constructed. The aggregation process takes place not only for multiple knowledge sources but also when there is only one knowledge source from which user data is upgraded periodically. Therefore, the user model is evaluated by using *multi-source* and *one-source* datasets.

The *one-source* datasets are prepared by collecting public user profiles from Facebook and Stack Overflow social web accounts. Approximately 1350 random user profiles are collected from Facebook by mining page likes. Similarly, nearly 1400 random Stack Overflow profiles are collected by gathering the tags of the questions asked by those users.

A *multi-source* dataset is prepared by selecting 100 users who have Facebook, LinkedIn and Twitter accounts and manually collecting their public social profiles. Facebook partial profiles consist of page likes, LinkedIn profiles include user's background information, skills and groups whereas Twitter profiles are the list of the accounts that the user follows.

Another *multi-source* dataset is prepared by discovering 626 users who both use Stack Overflow and LinkedIn accounts. Stack Overflow partial profiles consist of the tags of their posts whereas LinkedIn profiles include the skills.

The collected datasets enable evaluating the user model by using a general purpose social web site, a domain-specific social web site, a combination of different purpose social web sites and a combination of similar purpose domain specific social web sites.

5.1.2 Evaluation Methodology

The user model is evaluated as the hypergraph is populated by the current dataset with the specified thresholds. As new users and their partial profiles are aggregated into the hypergraph, we collect the performance scores of the system. Since we are interested in the aggregation performance we try to observe how the performance of the system changes as the aggregation process proceeds.

The datasets contain the users' partial profiles that consist of keyword lists. The users' partial profiles are added to the system one by one by looping the keywords in the partial profiles. For instance, let P_1, P_2, .., P_n be the partial profiles of users u_1, u_1, .., u_n. Each partial profile P_k where $1 < k < n$ is a list of terms t_1, t_2, .., t_{m_k}. For each P_i where i loops from 1 to n, for each term t_j where j loops from 1 to m_k, the terms are aggregated into the hypergraph based data structure. As the term t_j for profile P_i is processed, if the semantic item that corresponds to the term is already in the data structure and directly or indirectly connected to the user of the profile P_i, this means the system already knows about the user's interest on that item and it is evaluated as *success*. In information retrieval, *recall* is the ratio of the number of relevant items retrieved to the total number of relevant items in the database and is usually expressed as a percentage. In this study, we define *recall score* as the ratio of *successes* to the total number of items in the partial profile.

To see the improvement, the same datasets are evaluated with the baselines. The baselines construct a keyword-based user model by removing the semantic nature of the system. In other words, in the baseline evaluations, terms in the partial profiles are treated as keywords and external knowledge base is not used.

As stated, the scores are collected during the evaluation process and charts are obtained to see how the results change as the process proceeds. Therefore, the dataset is not separated as train and test data. During evaluation, all the users that are evaluated before the current user constitute the train dataset. This approach is chosen to observe the growth in the charts. If the dataset is separated as train and test sets, the growth may not be observed clearly.

5.1.3 Evaluation Results

Figure 5.1(a) illustrates the recall scores for the Facebook dataset consisting of 1349 test users. The y-axis is the recall score which is a value between 0 and 1. 0 means that the user model could not predict any of the user's partial profile items whereas 1 indicates that the system predicts all of the items in the partial profile. The x-axis denotes the users ordered according to their aggregation order. In other words, the profile of the user which is further from the origin is aggregated in the system later than the one closer to the origin. In the Facebook dataset of 1349 users, the average recall score increases as more users are aggregated in the system. Figure 5.1(b) shows the comparison of Facebook dataset of 1349 users with the baseline. It is clear that the user model outperforms the baseline and the improvement is calculated as 50 %.

Figure 5.2(a) demonstrates the evaluation for the Stack Overflow dataset of 1392 users. The average recall approximates to 1 as more user profiles are aggregated. The average recall values for Stack Overfow are higher than Facebook dataset. The reason for this difference is the fact that Facebook is a domain-independent platform whereas Stack Overflow is used for computer science domain. Figure 5.2(b) shows the baseline for Stack Overflow dataset. The improvement is 17.5 %, since the baseline recall score is also high.

The cross dataset of 100 users is used in different ways to measure the improvement. Subdatasets for each knowledge source that constitute the cross dataset are constructed. Stated in other words, subdatasets are projections of the cross dataset in one knowledge source only. 3 evaluations are executed for each knowledge source in the cross dataset. To observe Facebook results, the Facebook subdataset is constructed from the cross dataset by filtering data from other knowledge sources. The baseline evaluation is achieved by using the subdataset and removing the semantic nature of the aggregation process. Afterwards, the subdataset is evaluated by aggregating in an empty hypergraph and the results are compared with the baseline. Finally, the Facebook subdataset is evaluated by aggregating in the hypergraph previously populated by data from other knowledge sources in the cross dataset and the results are compared to the baseline. The same procedure is followed for LinkedIn and Twitter.

Figure 5.3(a) shows the comparison of Facebook subdataset to the baseline. The Facebook subdataset performs almost 1.5 times better than the baseline. Figure 5.3(b) and Figure 5.3(c) show the Facebook dataset aggregated after the hypergraph is populated with LinkedIn and Twitter datasets for the same users. The dataset performed almost 4 times better than the baseline.

Figure 5.4(a) demostrates the comparison of LinkedIn subdataset to the baseline. The improvement is 82 %. Figure 5.4(b) and Figure 5.4(c) shows the LinkedIn dataset aggregated after the hypergraph is populated with Facebook and Twitter partial profiles. The dataset performed 1.2 times better than the baseline.

Figure 5.5(a) shows the comparison of Twitter subdataset to the baseline. The subdataset performed 4.57 times better than the baseline. Figure 5.5(b) and Figure 5.5(c) shows the Twitter dataset aggregated after the hypergraph is populated with Facebook and LinkedIn profiles of the test users. The dataset performed 5.7 times better than the baseline.

Figure 5.6 shows the comparison of Stack Overflow dataset aggregated after LinkedIn profiles to the Stack Overflow dataset aggregated in empty initial hypergraph. The improvement is 6.82 %. Likewise, Figure 5.7 shows the comparison of LinkedIn dataset aggregated after Stack Overflow profiles to the LinkedIn dataset aggregated in empty initial hypergraph. The improvement is 3.33 %. For this case a slight improvement is achieved since the recall scores are already high for baseline.

The evaluation cases and scores are summarized in Table 5.1.

Table 5.1: Evaluation Scores

Evaluated Case	User Count	Recall	Improvement
Facebook	1349	0.54	50.00 %
Facebook Baseline	1349	0.36	-
Stackoverflow	1392	0.94	17.50 %
Stackoverflow Baseline	1392	0.80	-
Facebook after Twitter and LinkedIn	52	0.34	385.71 %
Facebook	52	0.17	142.86 %
Facebook Baseline	52	0.07	-
LinkedIn after Twitter and Facebook	88	0.64	128.57 %
LinkedIn	88	0.51	82.143 %
LinkedIn Baseline	88	0.28	-
Twitter after LinkedIn and Facebook	91	0.39	457.14 %
Twitter	91	0.32	357.14 %
Twitter Baseline	91	0.07	-
LinkedIn after Stackoverflow	626	0.94	6.82 %
LinkedIn Baseline	626	0.88	-
Stackoverflow after LinkedIn	626	0.93	3.33 %
Stackoverflow Baseline	626	0.90	-

Table 5.2: Profile Aggregation Evaluation Results

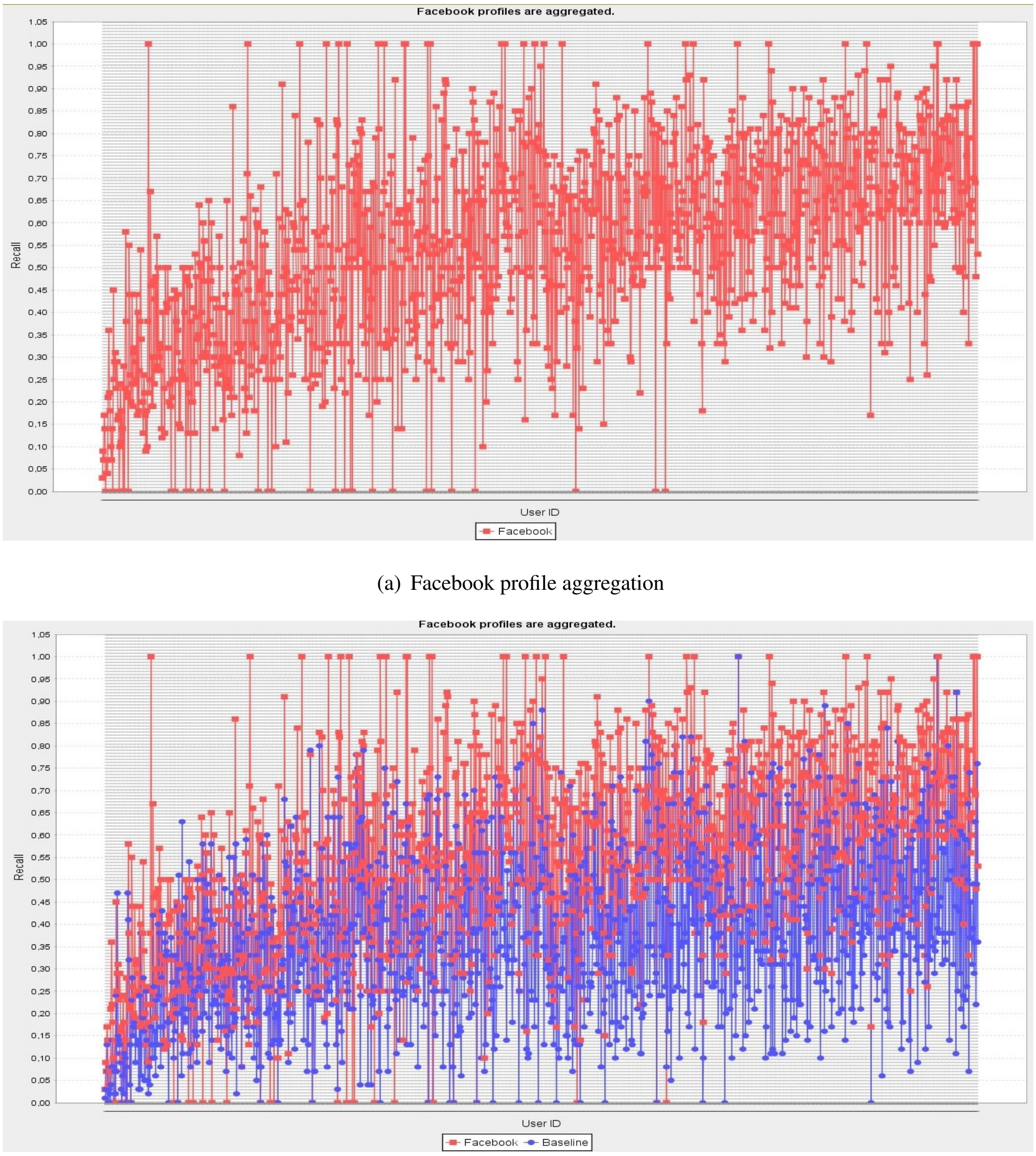
Evaluation Case	**F-Measure Score**
Cross Dataset	**0.42**
LinkedIn-Only Baseline	0.20
Twitter-Only Baseline	0.12
Facebook-Only Baseline	0.10

3-fold cross validation evaluation:

In information retrieval, *recall* is the ratio of the number of relevant items retrieved to the total number of relevant items in the database. It is usually expressed as a percentage. *Precision* is the ratio of the number of relevant items retrieved to the number of all items retrieved. *F-measure* is a combination of precision and recall as $(2 * P * R)/(P + R)$ where P and R stands for precision and recall respectively. In this case, we used F-Measure to express evaluation results.

Since the dataset is small, we did 3-fold cross validation evaluation by separating dataset into train and test with 70 to 30 percent ratio, respectively. First fold is the original ordering of items in partial profiles for each user. 70 percent of each partial profile is taken as train set and used to populate database. Remaining 30 percent is used as test data to obtain score.Test data is not saved in the database. Evaluation is repeated three times, since this is a 3-fold evaluation. In second folds, keywords are sorted alphabetically and in third fold, random ordering is used.

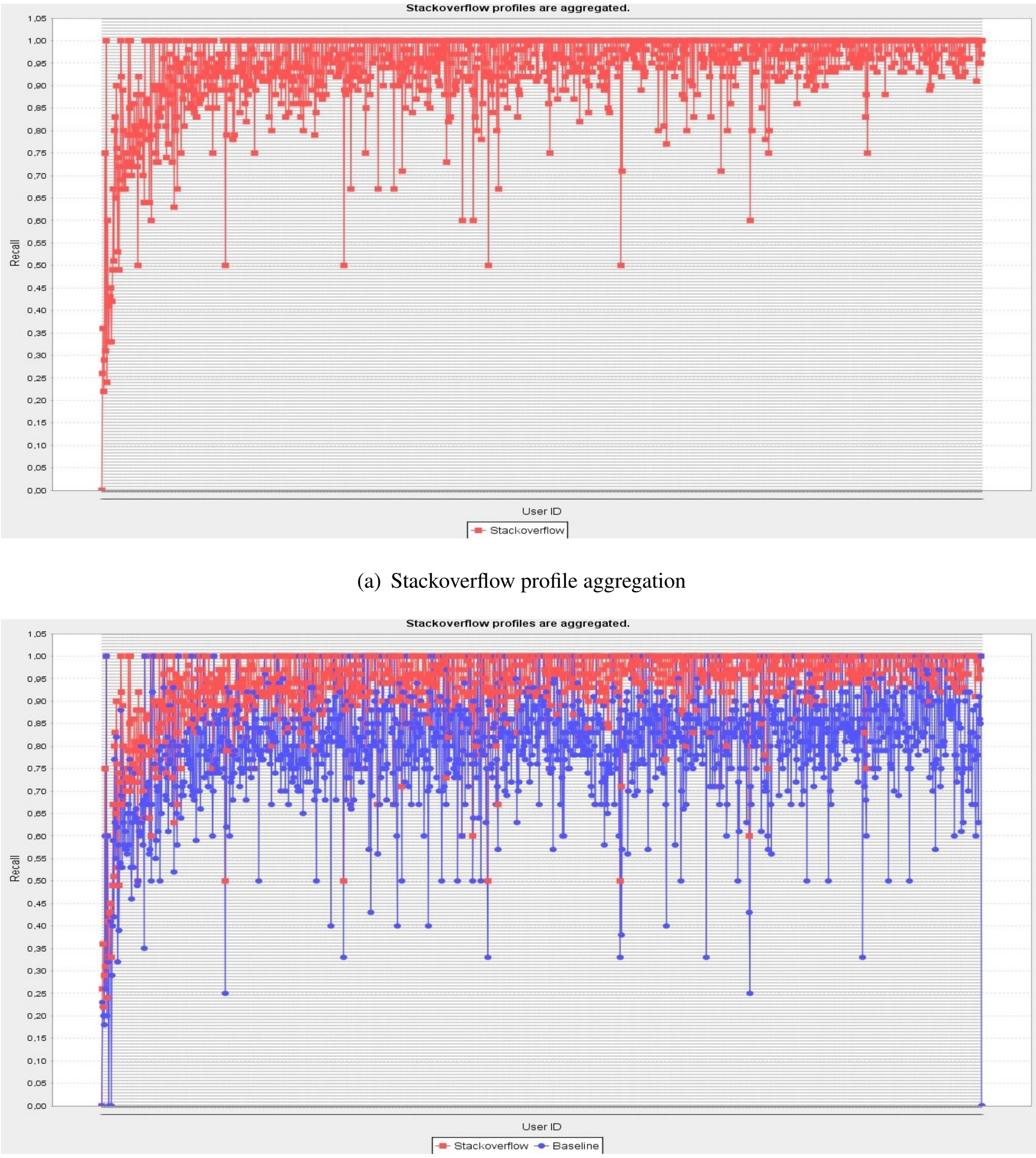
We evaluated the system using aggregated profile. As baseline, we evaluated using partial profiles. We averaged the scores obtained from 3-folds. The evaluation cases and scores are summarized in Table 5.2. Partial LinkedIn profile perfomed better than partial Twitter profile which performed better than partial Facebook profile. The reason for this might be the size of the term universe differences between LinkedIn, Twitter and Facebook. Since Facebook is a generic network, its term universe is much broader than LinkedIn which is restricted to professional domain. Aggregated profile outperformed partial profiles with F-measure score 0.42 whereas best performing partial profile's score is 0.20.



(a) Facebook profile aggregation

(b) Facebook profile aggregation vs. Baseline

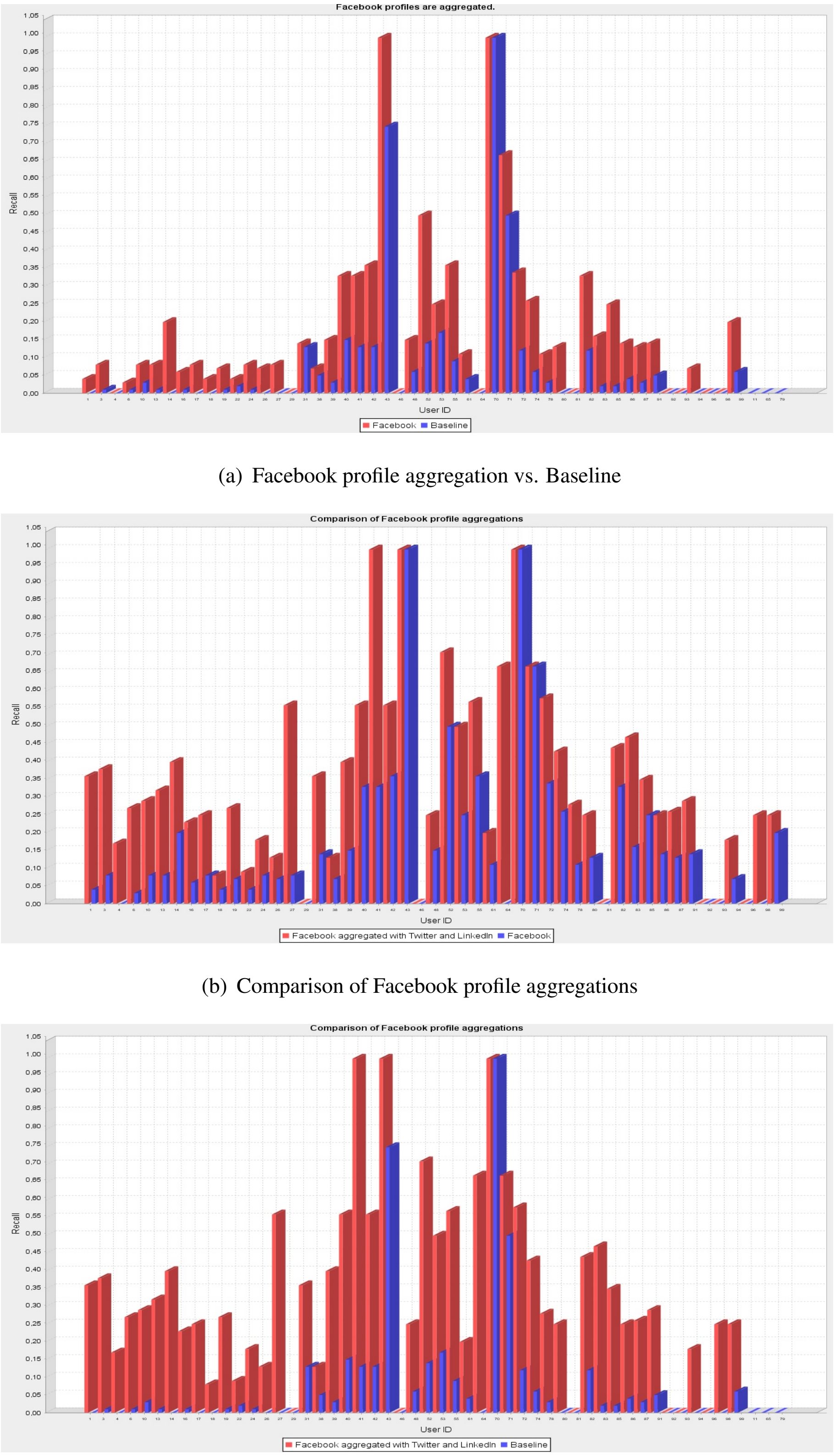
Figure 5.1: Facebook profile aggregation alone and compared to the Baseline



(a) Stackoverflow profile aggregation

(b) Stackoverflow profile aggregation vs. Baseline

Figure 5.2: Stackoverflow profile aggregation alone and compared to the Baseline

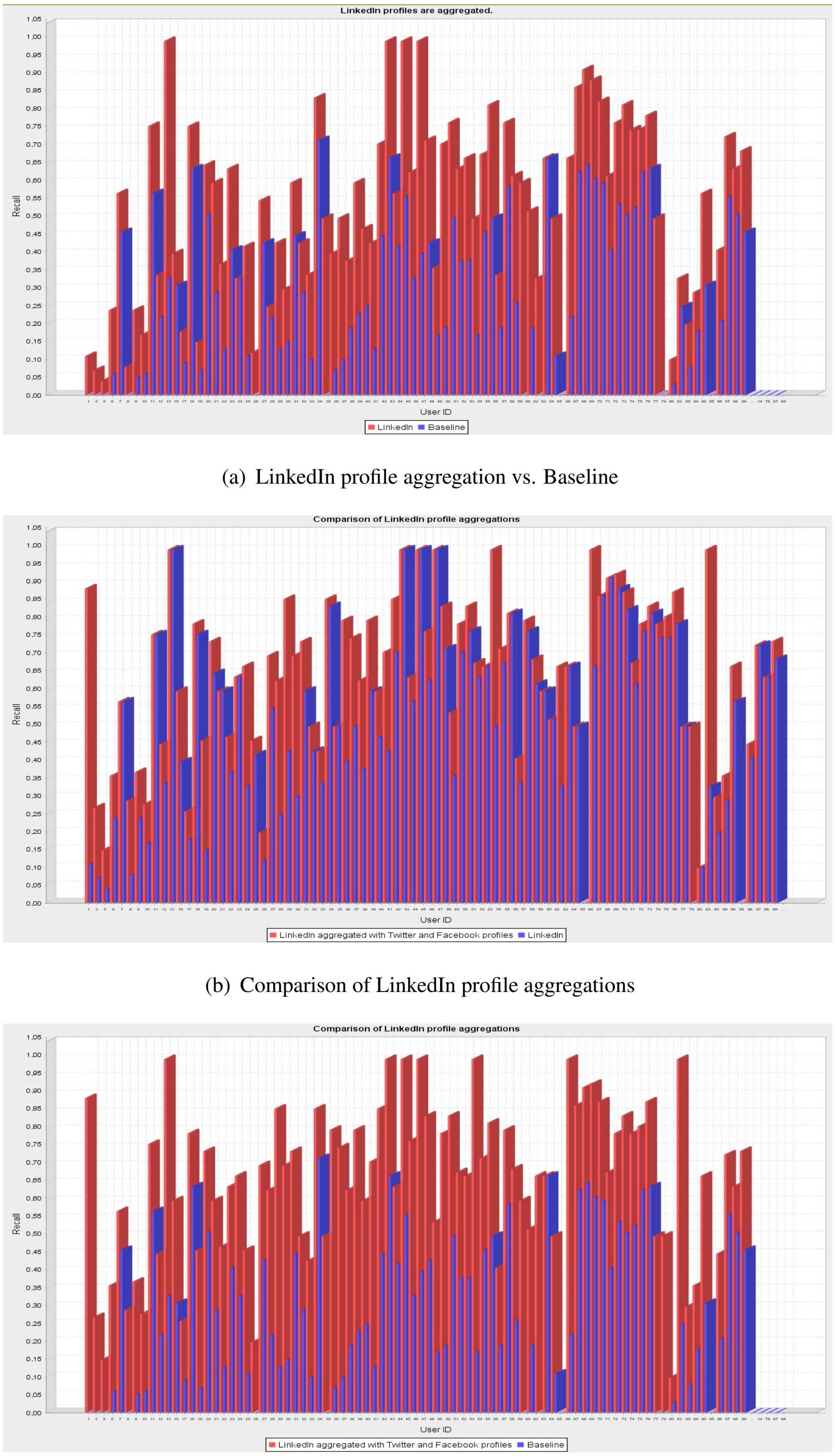


(a) Facebook profile aggregation vs. Baseline

(b) Comparison of Facebook profile aggregations

(c) Comparison of Facebook profile aggregations vs Baseline

Figure 5.3: Facebook profile aggregation results

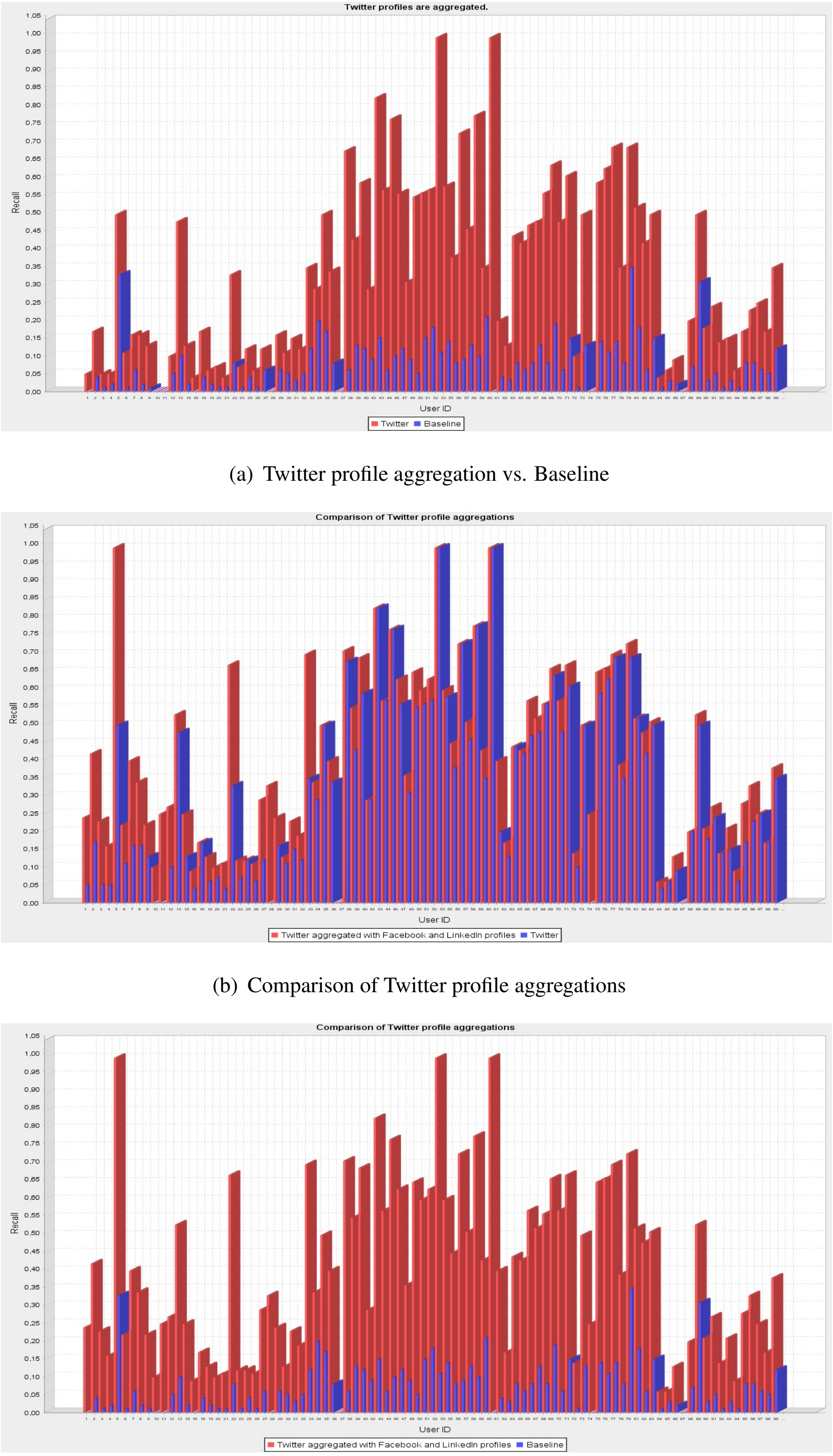


(a) LinkedIn profile aggregation vs. Baseline

(b) Comparison of LinkedIn profile aggregations

(c) Comparison of LinkedIn profile aggregations vs Baseline

Figure 5.4: Linkedin profile aggregation results



(a) Twitter profile aggregation vs. Baseline

(b) Comparison of Twitter profile aggregations

(c) Comparison of Twitter profile aggregations vs Baseline

Figure 5.5: Twitter profile aggregation results

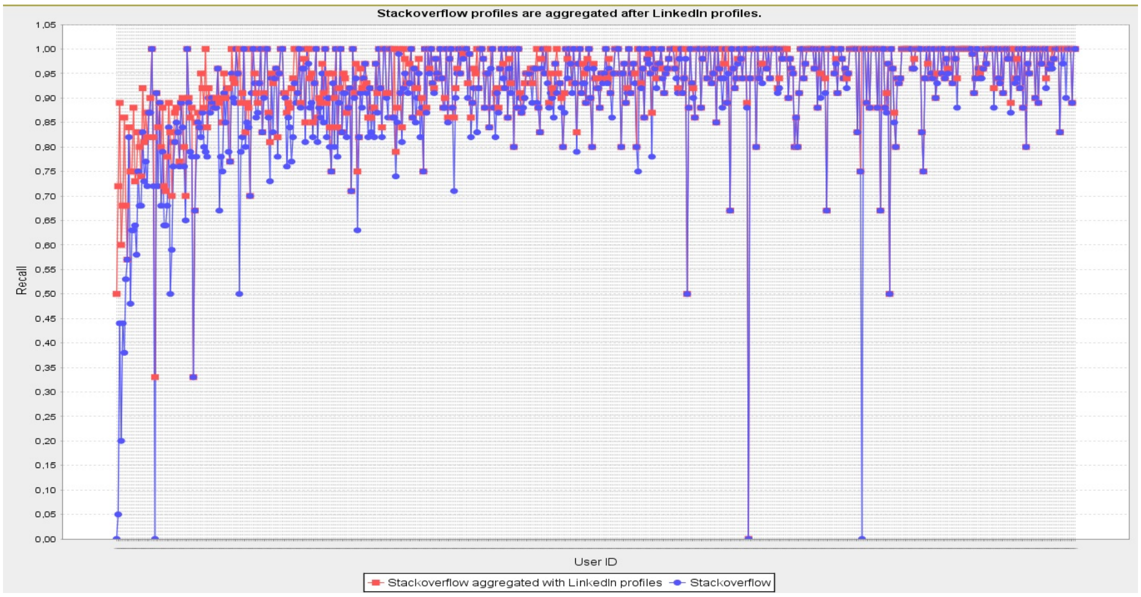


Figure 5.6: Comparison of Stack Overflow profile aggregation vs Baseline

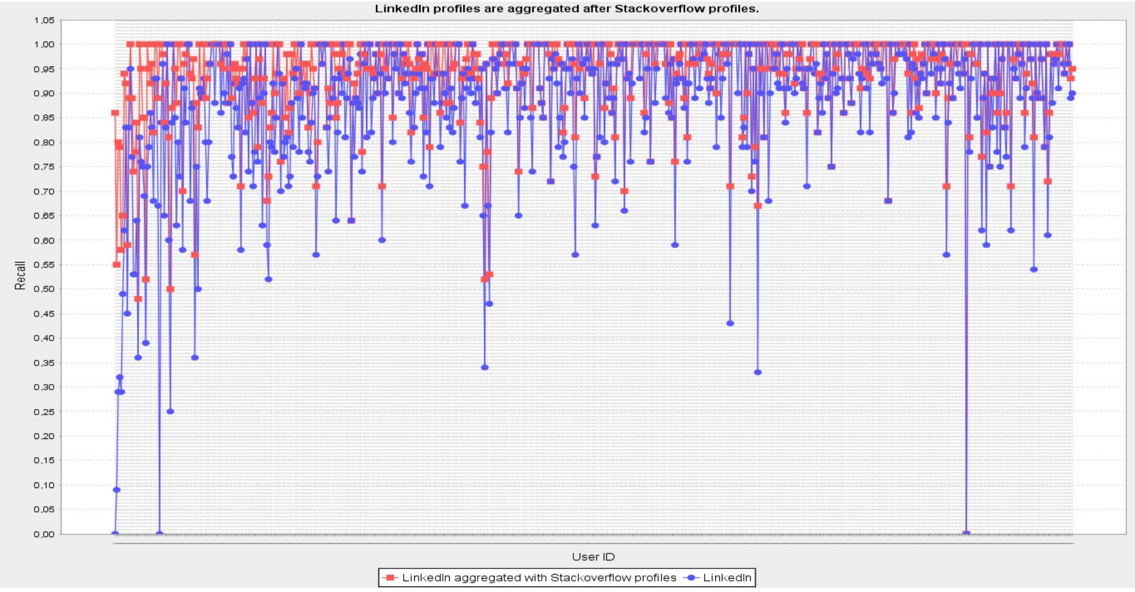


Figure 5.7: Comparison of LinkedIn profile aggregation vs Baseline

CHAPTER 6

EXTENDING HYPERGRAPH BASED USER MODELING FRAMEWORK WITH CONTEXT INFORMATION

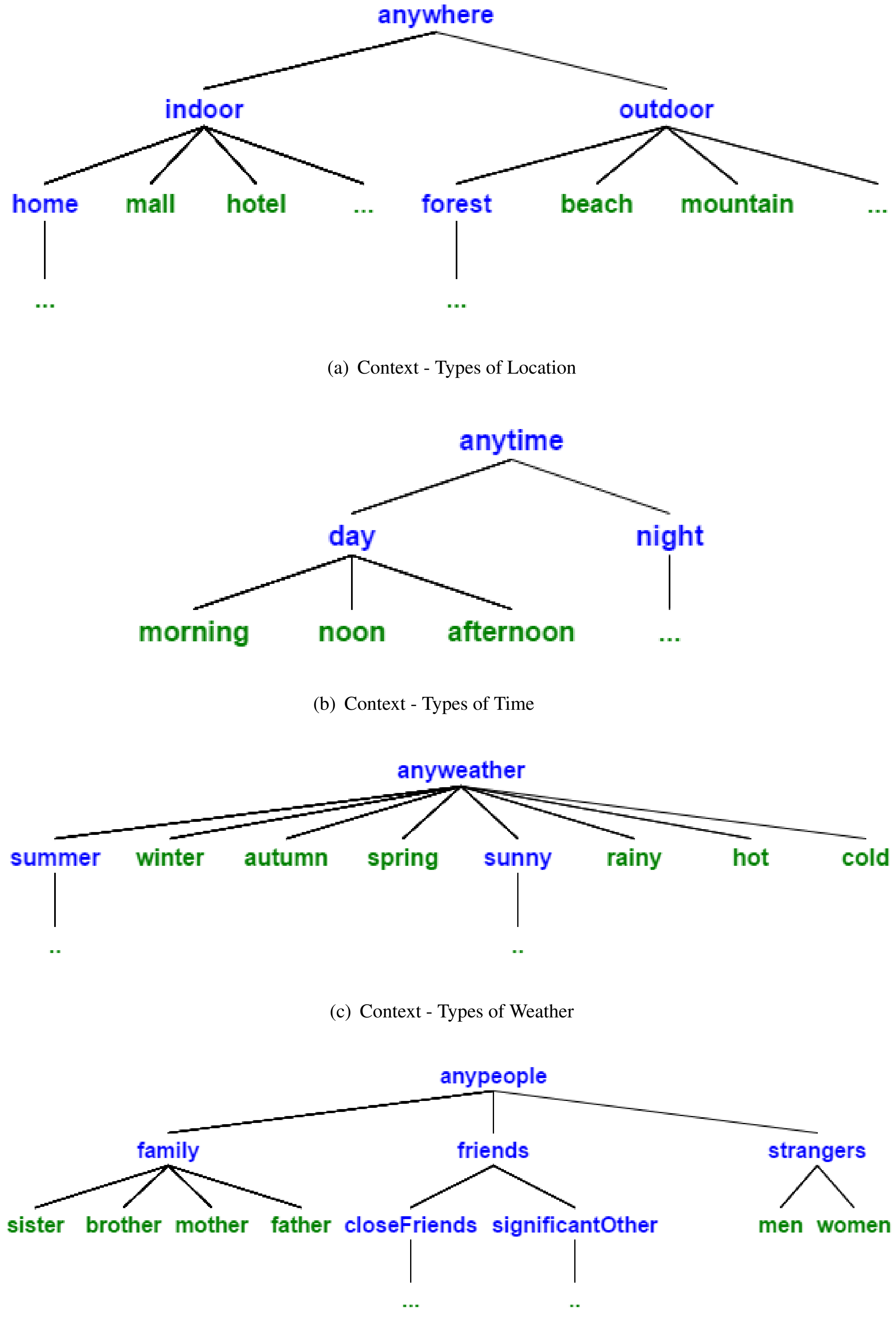
In this chapter, we show that the proposed hypergraph based user modeling framework is extendible. In order to illustrate this, we extend the framework by adding context information.

6.1 Modeling with Context

Context basically defines the situation of the user. In the extended framework, we modeled the context in four dimensions: *location, time, weather* and *accompanying people*. We defined each dimension with a basic ontology. The context ontologies are illustrated in Figure 6.1. As an example scenario, the user is checked at a *cinema* in the *afternoon* watching *The Amazing Spiderman* with her *close friends* when it is *raining* outside. In this case, the location is the cinema, the time is the afternoon, the weather is rainy and accompanying people are the user's *close friends*.

Table 6.1: Extending User Model with Context

Notation	Description	Type
c_L	a location context	Node
C_L	Set of location contexts	Hyperedge
c_T	a time context	Node
C_T	Set of time contexts	Hyperedge
c_W	a weather context	Node
C_W	Set of weather contexts	Hyperedge
c_P	an accompanying people context	Node
C_P	Set of accompanying people contexts	Hyperedge
$E_{L_{ont}}$	The ontologic relation between locations	Hyperedge
$E_{T_{ont}}$	The ontologic relation between times	Hyperedge
$E_{W_{ont}}$	The ontologic relation between weathers	Hyperedge
$E_{P_{ont}}$	The ontologic relation between accompanying people	Hyperedge
c	a context instance	Node
C	Set of contexts instances	Hyperedge
$E_{user2context}$	The relation between user and context	Hyperedge
$E_{context2item}$	The relation between context and item	Hyperedge
E_{c_L}	The relation between context instance and location context ontology	Hyperedge
E_{c_T}	The relation between context instance and time context ontology	Hyperedge
E_{c_W}	The relation between context instance and weather context ontology	Hyperedge
E_{c_P}	The relation between context instance and people context ontology	Hyperedge



(a) Context - Types of Location

(b) Context - Types of Time

(c) Context - Types of Weather

(d) Context - Types of People

Figure 6.1: Context Ontologies

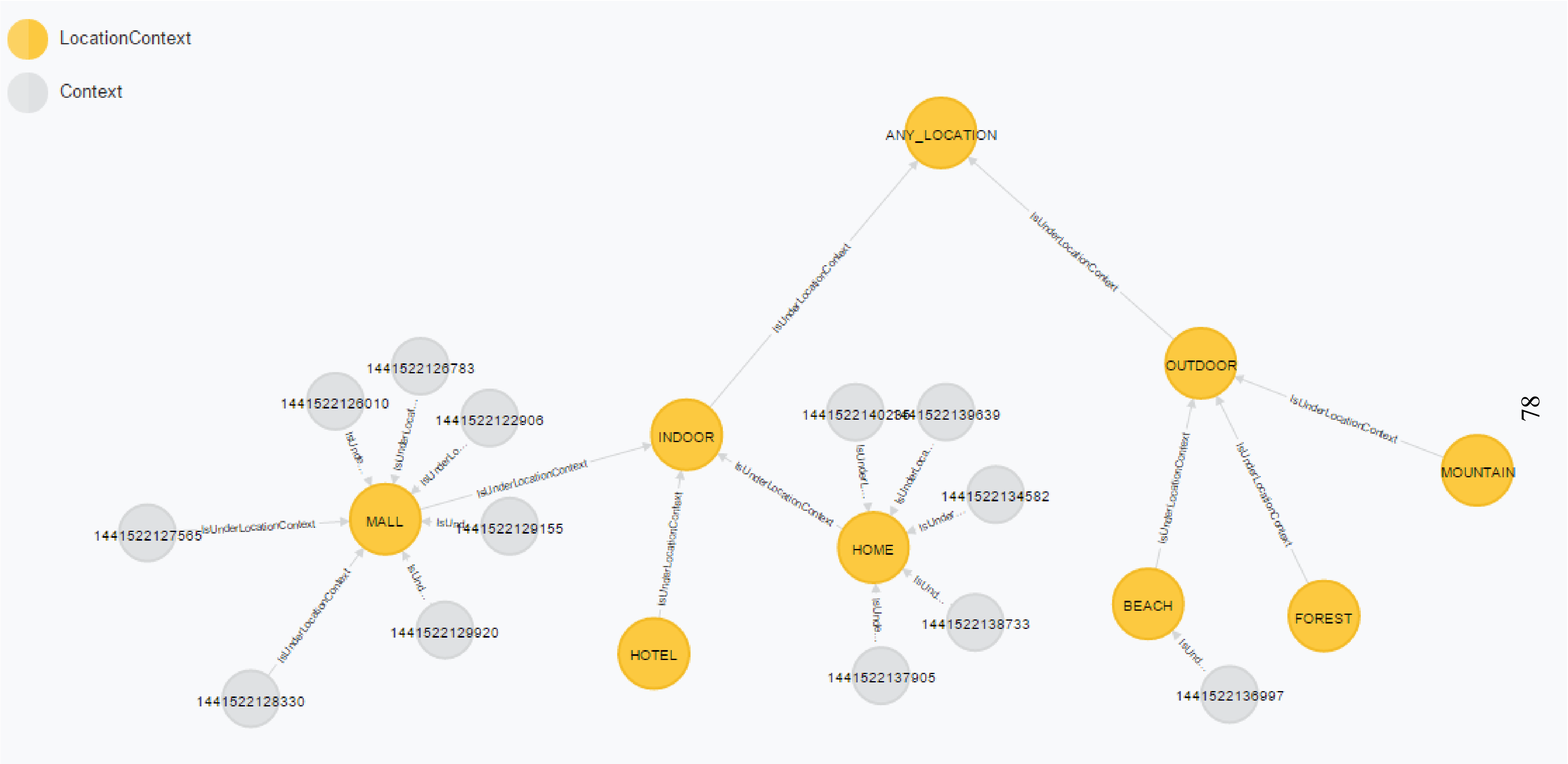


Figure 6.2: Context - Location

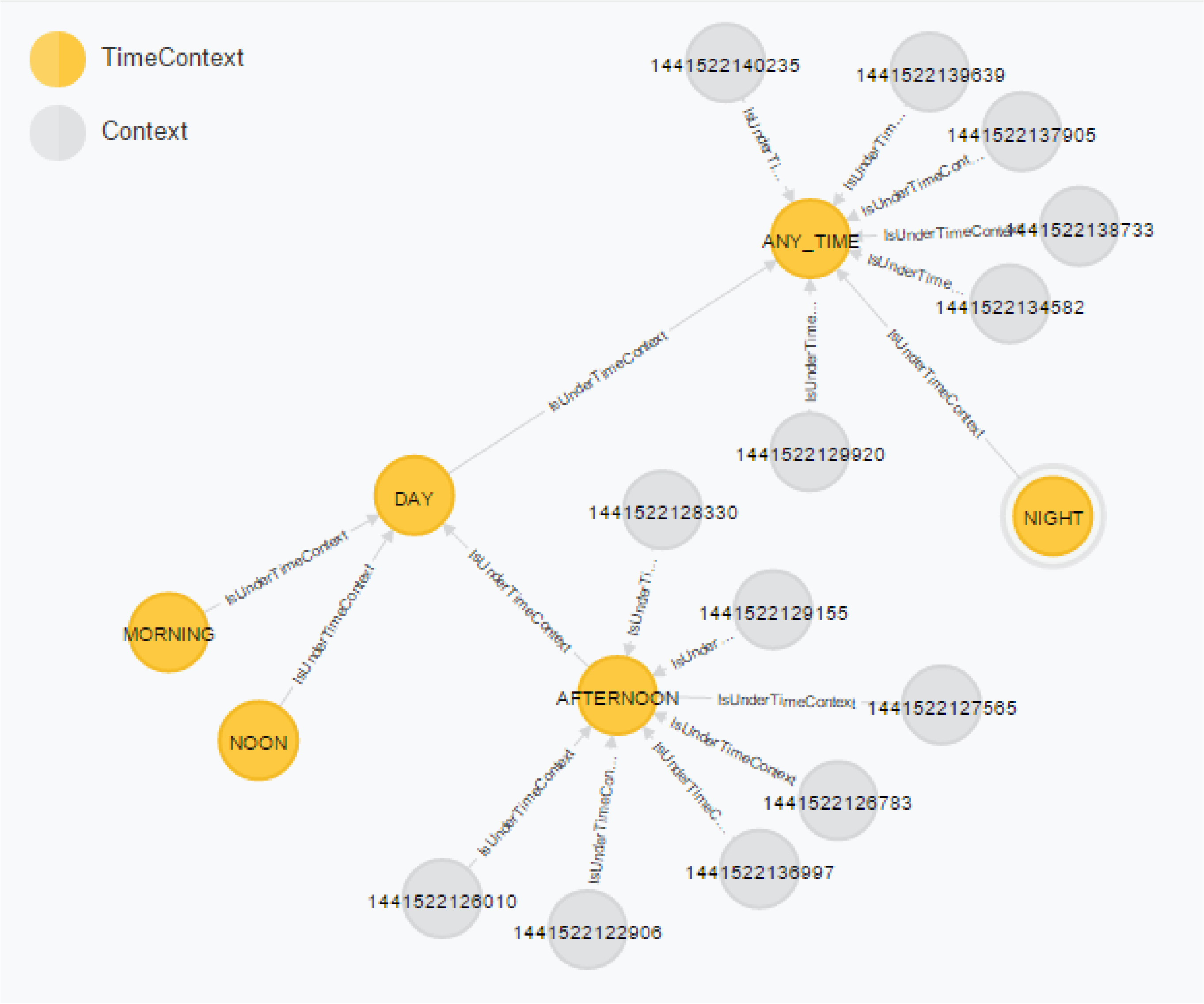


Figure 6.3: Context - Time

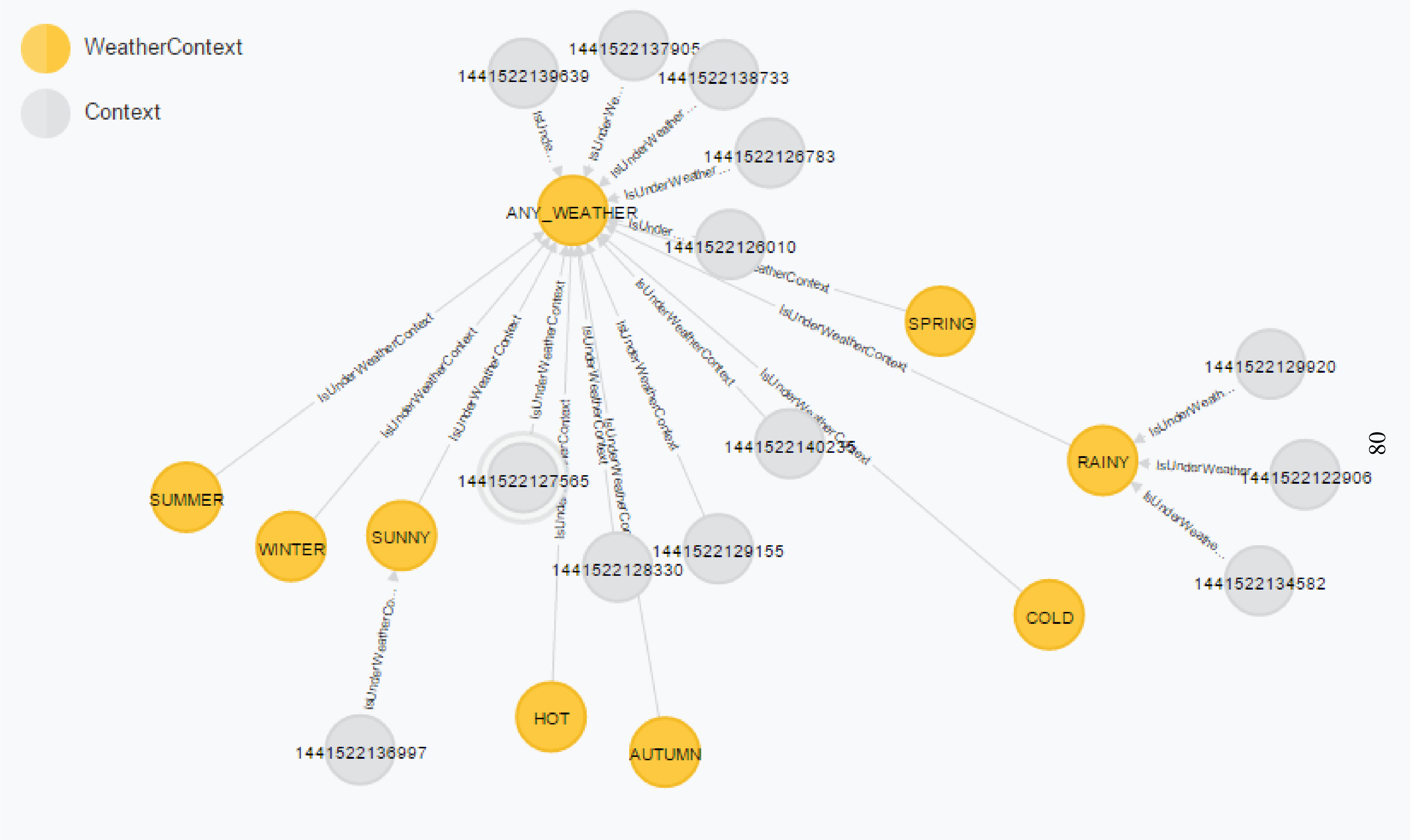


Figure 6.4: Context - Weather

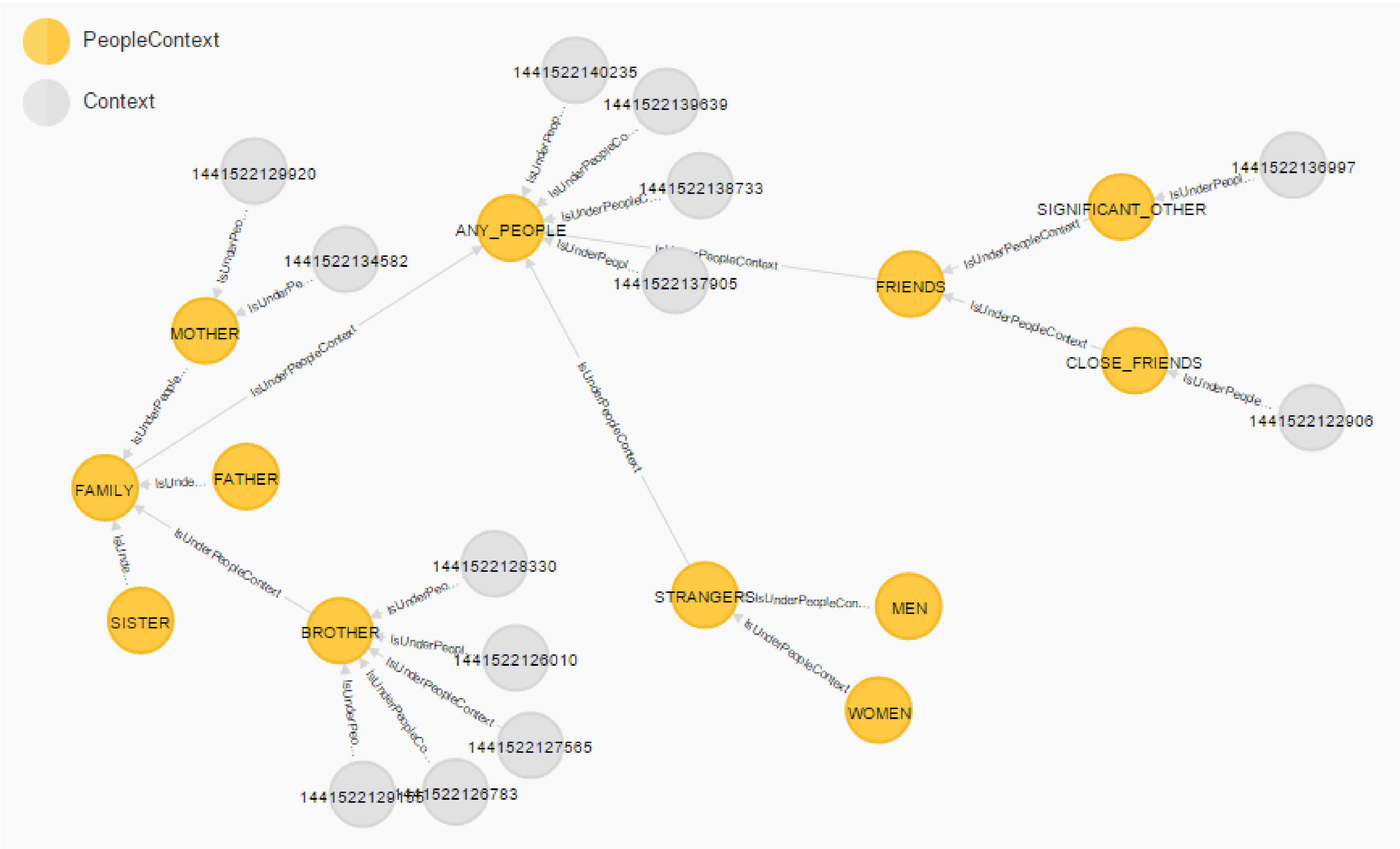


Figure 6.5: Context - People

The extended context part of the framework is displayed in Table 6.1. In the model, c_L stands for a location context and C_L is the set of all location contexts supported by the system. $E_{L_{ont}}$ is the hyperedge connecting the location contexts according to the ontology. Figure 6.2 shows the hypergraph for the location context. In the hypergraph, yellow nodes models the location contexts. *ANY LOCATION* represents the absence of location context information. *INDOOR* and *OUTDOOR* location contexts are more specialized contexts and are related with their parent with the relation *isUnderLocationContext*. The more specialized locations are related to *INDOOR* and *OUTDOOR* simulating the ontology given in Figure 6.1(a). The gray nodes in the hypergraph shows context instances. In the framework definition, c stands for a context instance and C wraps all the context definitions in the system. The modeling approach is similar for other context types and the hypergraph for time, weather and accompanying people are presented in Figures 6.3, 6.4 and 6.5 respectively.

In the framework, we use different hyperedge types to indicate different relationships. For instance, the semantic relationships between location contexts are related with $E_{L_{ont}}$ hyperedges. Similarly, $E_{T_{ont}}$, $E_{W_{ont}}$ and $E_{P_{ont}}$ hyperedges are used for relating time, weather and accompanying people contexts.

Location, time, weather and people context nodes (c_L, c_T, c_W and c_P) and semantic relations between them are created at the system initiation. When an information about the user is going to be aggregated into the model, a context instance (c) is created. The context instance contains information about all types of contexts and related to them by using hyperedges E_{c_L}, E_{c_T}, E_{c_W} and E_{c_P} for location, time, weather and accompanying people respectively. In the model, in order to illustrate an interest, the user is related to the context instance (c) and the context instance is related to the item of interest. The hyperedge which relates user with the context is $E_{user2context}$ and context with the item is $E_{context2item}$.

Figure 6.6 shows how the user's interest in an item under context is modeled. Basically, the user is related to the context and the context is related to the item. The context is an instance and it behaves like a pointer that points to real context nodes for location, time, weather and accompanying people dimension. In the example, the context shows that the user likes the item when she is with her *BROTHER* in the

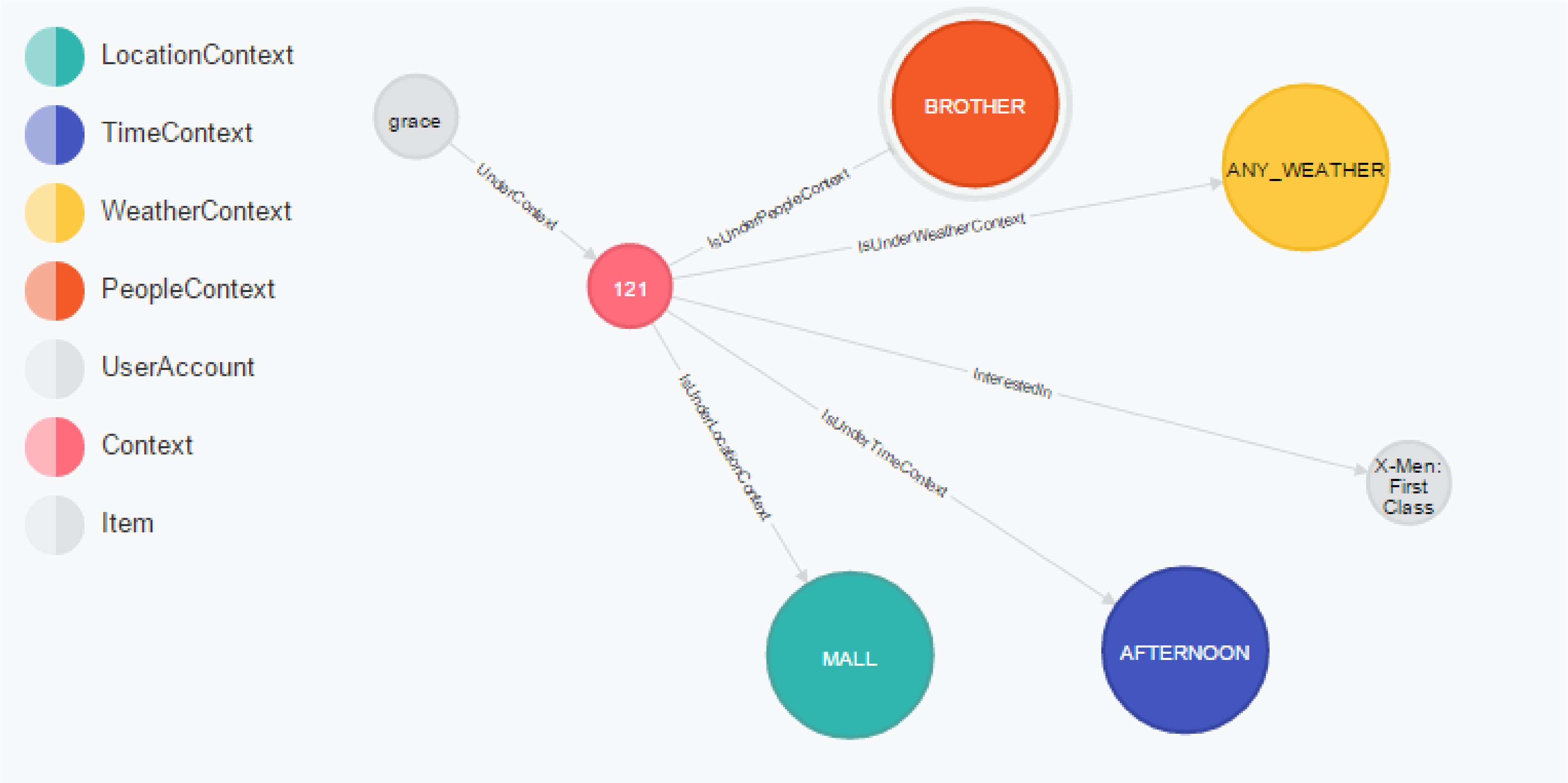


Figure 6.6: Modeling an Interest with Context

AFTERNOON, at the *MALL*. Weather context shows *ANY WEATHER* which means the user is interested in the item independent of how the weather is. When a new interest information is modeled, a new context node is created. But there is only one *BROTHER* node in the system and all the context instances which models *with brother* context are related to that node. This information is valid for all location, time, weather and accompanying people nodes in the hypergraph.

In order to support context, the partial profiles should include context information. Once the context information is provided, the introduced extension enables considering context in the framework.

6.2 Querying with Context

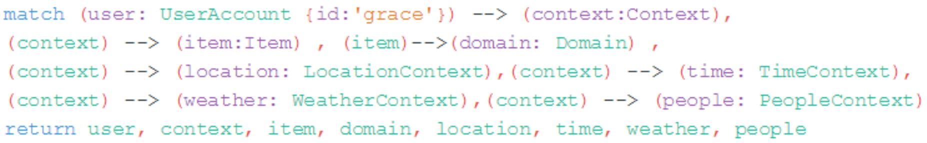
The proposed hypergraph based user modeling framework provides an effective querying capability for the user modeling domain with the help of different types of nodes and edges. The semantic user profile retrieval query is extended by adding context c as parameter. Domain based profile under context c is presented in Equation 6.1. According to the formulation, in the resulting subgraph user u is connected to the context c and c is connected to the items i. In other words, if context c is connected to both user u and item i, then the item is included in the result. The connection to the domain d is trivial and it means that the domain information is also included in the result.

$$P_{\textit{domain with context}}(u; d; c; max) = u \rightarrow c \xrightarrow{*0..max-1} (i) \xrightarrow{IsInDomain} d \tag{6.1}$$

General user profile is shown in Equation 6.2. The only difference from domain based user profile is the absence of the domain information.

$$P_{\textit{general with context}}(u; c; max) = u \rightarrow (c) \xrightarrow{*0..max-1} (i) \tag{6.2}$$

The extended hypergraph user model is implemented. The system retrieves the user profile with the Cypher query in Figure 6.7. As an example, to retrieve the user model for *Grace*, the node representing Grace is located, the context instances connected with Grace, the items that are connected to the context instances and the domains that are connected to the items are all retrieved. Moreover, contexts that are connected to the retrieved context instances are added to the subgraph. The basic profile hypergraph is shown in Figure 6.8. For simplicity, domain and context type nodes are eliminated. In the profile, the user *grace* is the node which is located in the middle of the graph with a blue circle. Her interests are modeled by relating her to the context with *UnderContext* hyperedge and by connecting the context to the interest with *InterestedIn* hyperedge.



```
match (user: UserAccount {id:'grace'}) --> (context:Context),
(context) --> (item:Item) , (item)-->(domain: Domain) ,
(context) --> (location: LocationContext),(context) --> (time: TimeContext),
(context) --> (weather: WeatherContext),(context) --> (people: PeopleContext)
return user, context, item, domain, location, time, weather, people
```

Figure 6.7: User Profile Query

The enhanced user model Cypher query is presented in Figure 6.9. The underlined query fragment results in retrieval of items that are indirectly connected to the user. The resulting hypergraph is given in Figure 6.10. Sample profile information that we can see from the figure:

- *Grace* is interested in *Pride and Prejudice* when she is at the *mall* in the *afternoon* with her *close friends* and it is a *rainy* day.

- *Grace* is interested in *Knitting* when she is at *home* on a *rainy* day with her *mother*.

- *Grace* is interested in *Cooking* when she is at *home* on a *rainy* day with her *mother*.

- *Grace* is interested in *Fantastic* when she is at the *mall* in the *afternoon* with her *brother*.

The presented framework supports context with the provided extension. Figure 6.11 shows the basic user profile hypergraph with location context information. The information in this hypergraph is listed as follows:

- *Grace* is interested in *swimming* when she is at the *beach*.

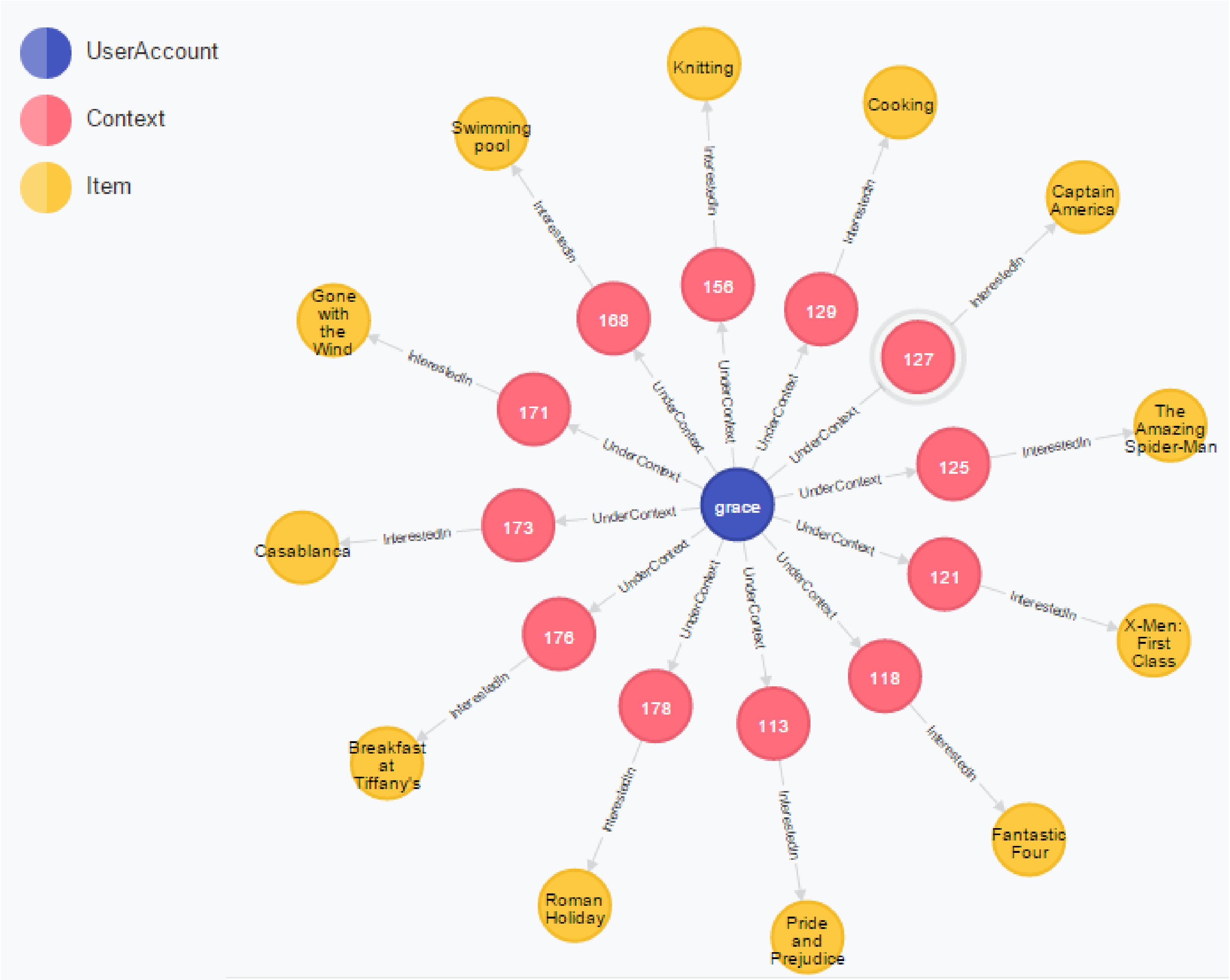
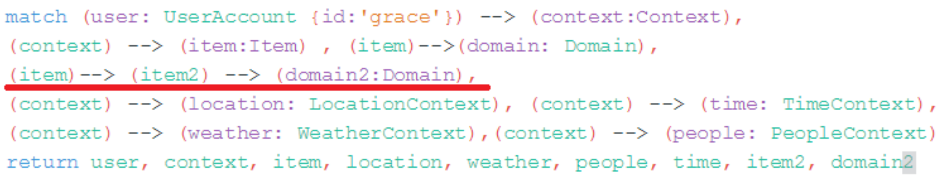


Figure 6.8: Basic User Profile



```
match (user: UserAccount {id:'grace'}) --> (context:Context),
(context) --> (item:Item) , (item)-->(domain: Domain),
(item)--> (item2) --> (domain2:Domain),
(context) --> (location: LocationContext), (context) --> (time: TimeContext),
(context) --> (weather: WeatherContext),(context) --> (people: PeopleContext)
return user, context, item, location, weather, people, time, item2, domain2
```

Figure 6.9: Enhanced User Profile Query

- *Grace* is interested in *Captain America, XMen First Class, The Amazing Spiderman, Fantastic Four* and *Pride and Prejudice* when she is at the *mall*.

- *Grace* is interested in *Roman Holiday, Breakfast at Tiffany's, Casablanca, Gone with the Wind* and *knitting* when she is at *home*.

Since the framework supports context, the system is capable of providing user profile under a specified context. For instance, the system provides user profile when the user at home. The Cypher query is given in Figure 6.12. In the query, the underlined fragment results in limiting the location to the *home*. The resulting user profile is in Figure 6.14. The user likes *Roman Holiday, Breakfast at Tiffany's, Casablanca, Gone with the Wind* and *knitting* when she is at *home*.

Accompanying people may affect the user's choices and the *people* context is used for this. The system is capable of retrieving the user's profile when she is with her brother. The Cypher query is in Figure 6.13 and the underlined part restricts the people context to *brother*. The hypergraph is shown in Figure 6.15. The user likes *Fantastic Four, XMen: First Class, Captain America* and *The Amazing Spiderman* when she is with her brother.

We showed that our system is capable of supporting context and presented a basic concept illustration in this chapter. In literature, there are user models which support context [84, 45]. [84] links the user's interests to the situation of the user. The study keeps track of the user behaviour and the situation under the behaviour takes place. The context information comes from the *context providers*. The constructed context aware user model is utilized for making recommendations to applications and services by considering the context of the individual. In this thesis, we can not control the user's behaviour, since we do not extract the partial profile real time. However, context provider module could inspire us. [45] presents a context management frame-

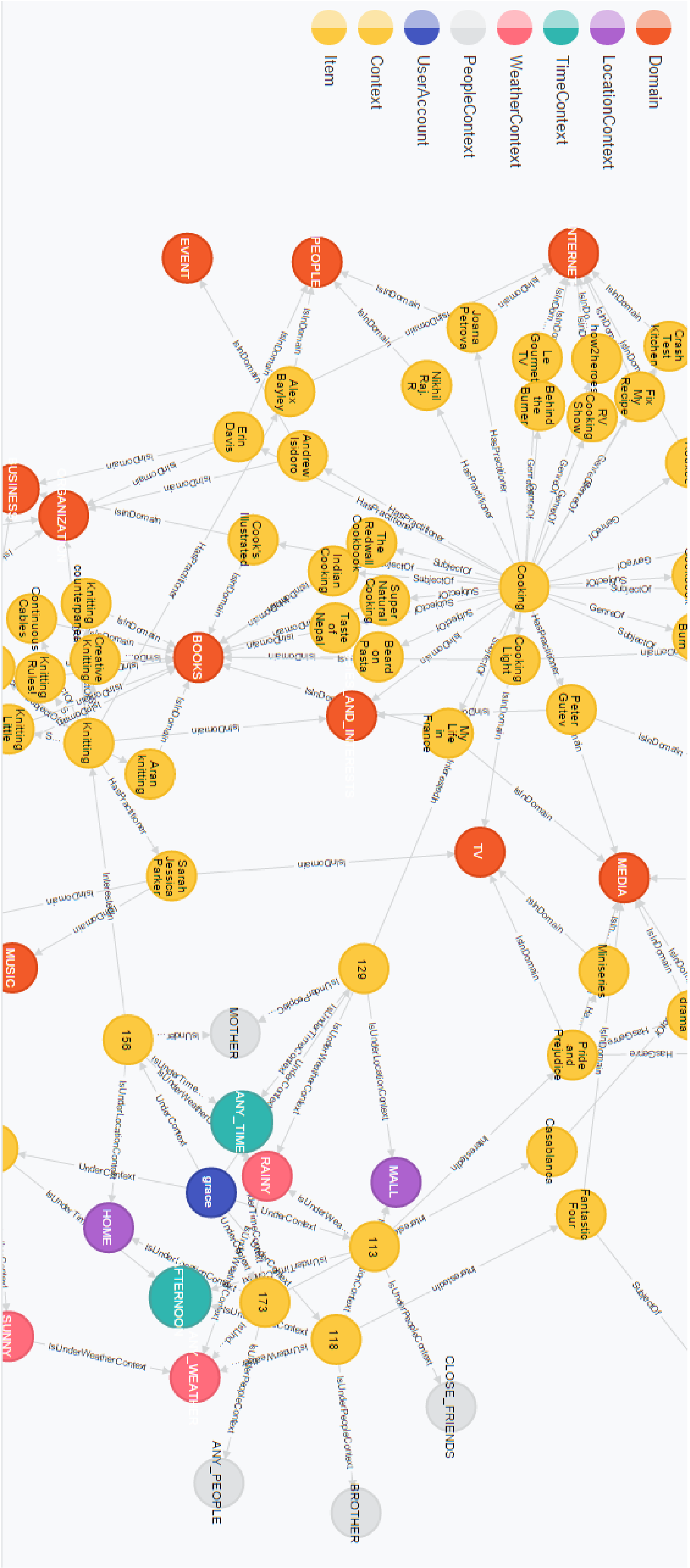


Figure 6.10: Enhanced User Profile

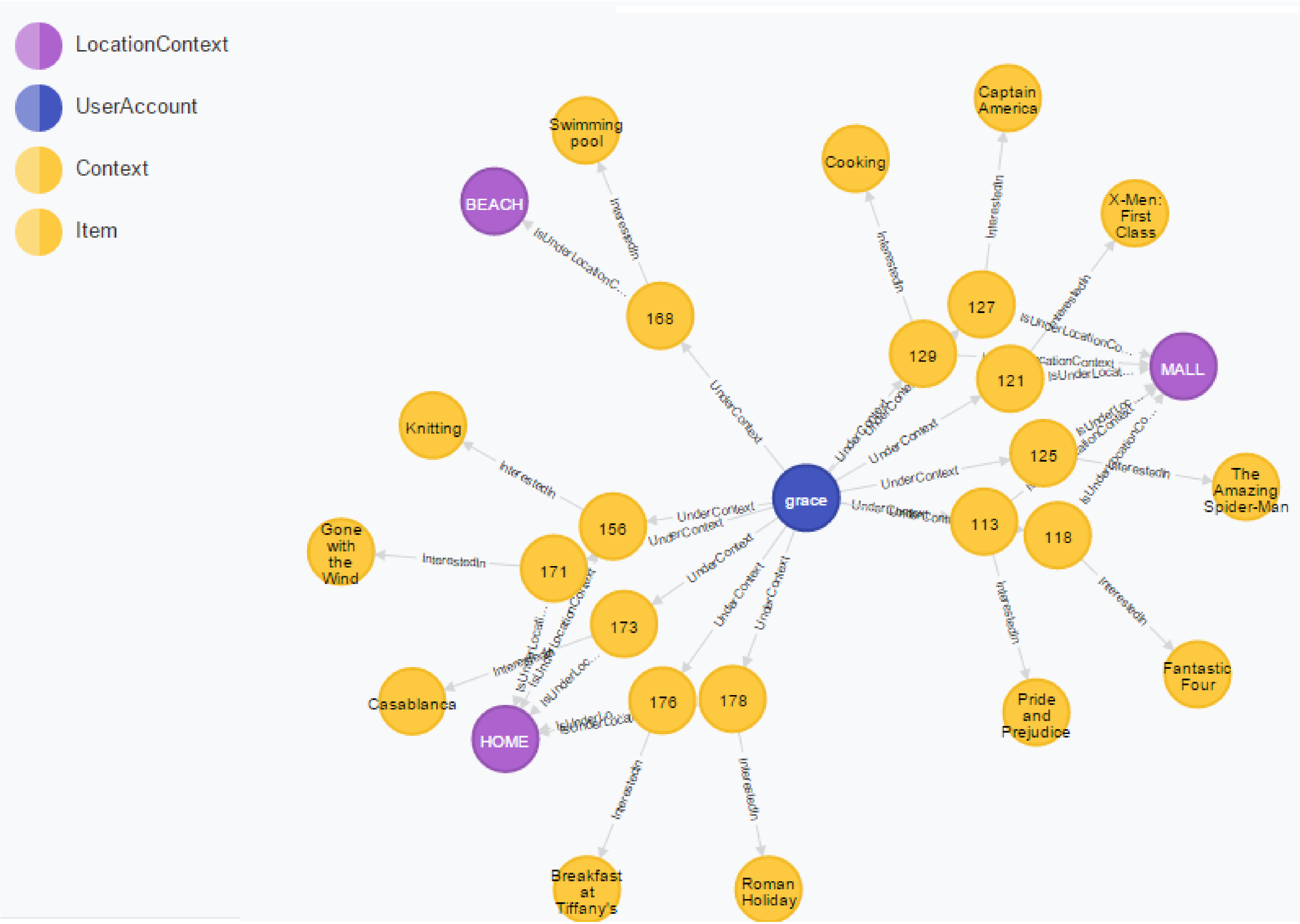
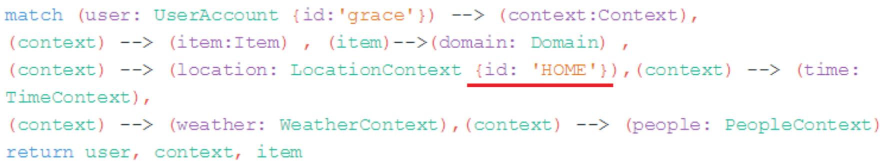
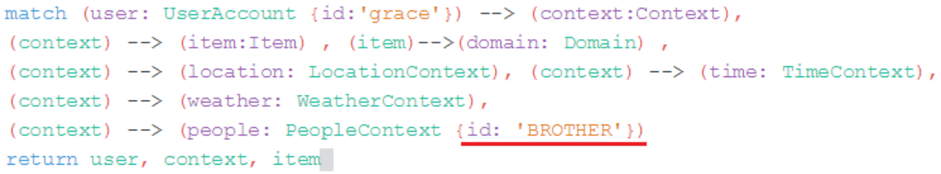


Figure 6.11: User Profile with Location Context



```
match (user: UserAccount {id:'grace'}) --> (context:Context),
(context) --> (item:Item) , (item)-->(domain: Domain) ,
(context) --> (location: LocationContext {id: 'HOME'}),(context) --> (time:
TimeContext),
(context) --> (weather: WeatherContext),(context) --> (people: PeopleContext)
return user, context, item
```

Figure 6.12: User Profile At Home Query



```
match (user: UserAccount {id:'grace'}) --> (context:Context),
(context) --> (item:Item) , (item)-->(domain: Domain) ,
(context) --> (location: LocationContext), (context) --> (time: TimeContext),
(context) --> (weather: WeatherContext),
(context) --> (people: PeopleContext {id: 'BROTHER'})
return user, context, item
```

Figure 6.13: User Profile with Brother Query

work. In general, context is important for mobile or ubiquitous environments [103]. Therefore, extending the proposed framework with context may result in extending support for mobile and ubiquitous applications.

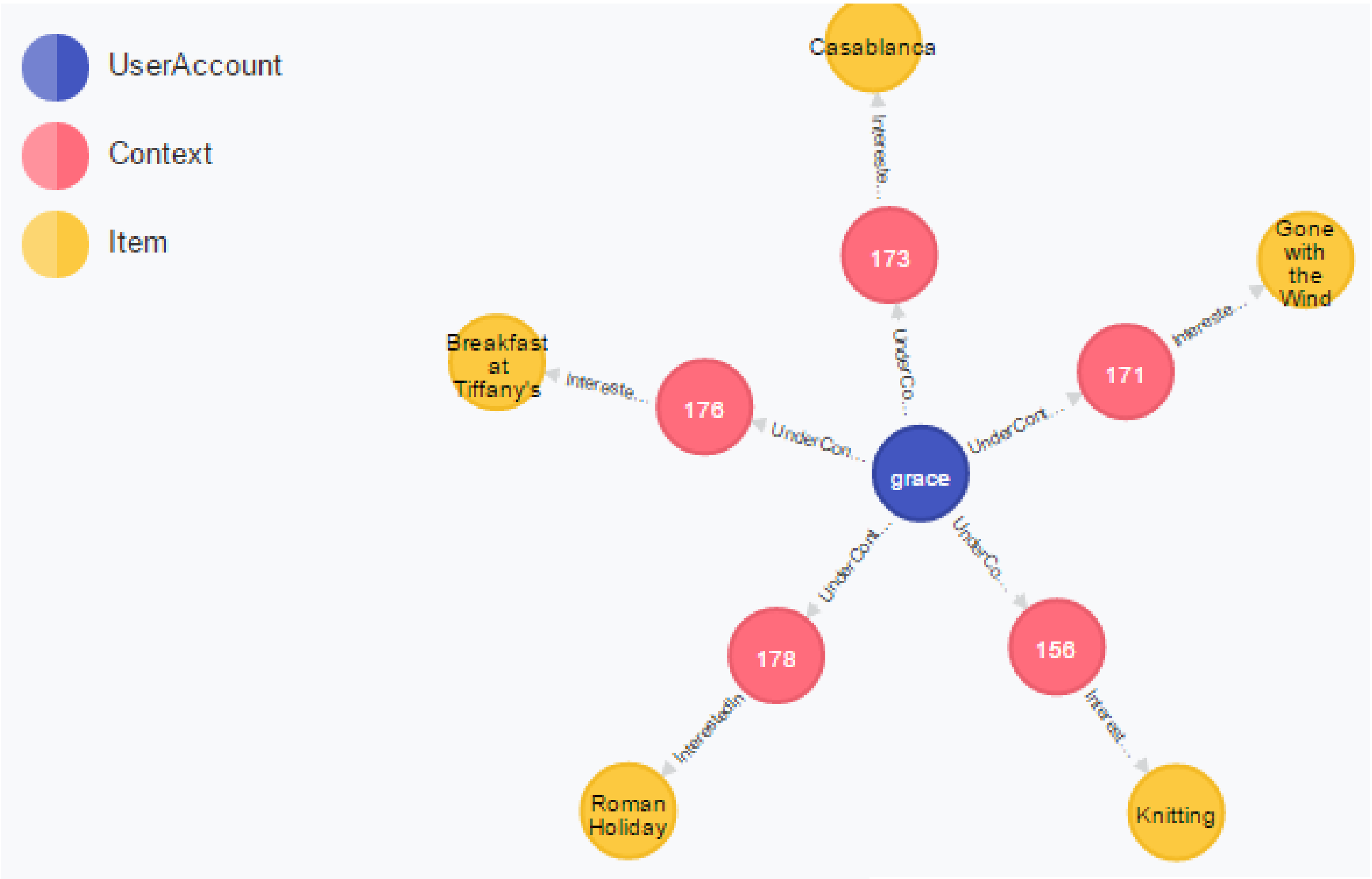


Figure 6.14: User Profile at Home

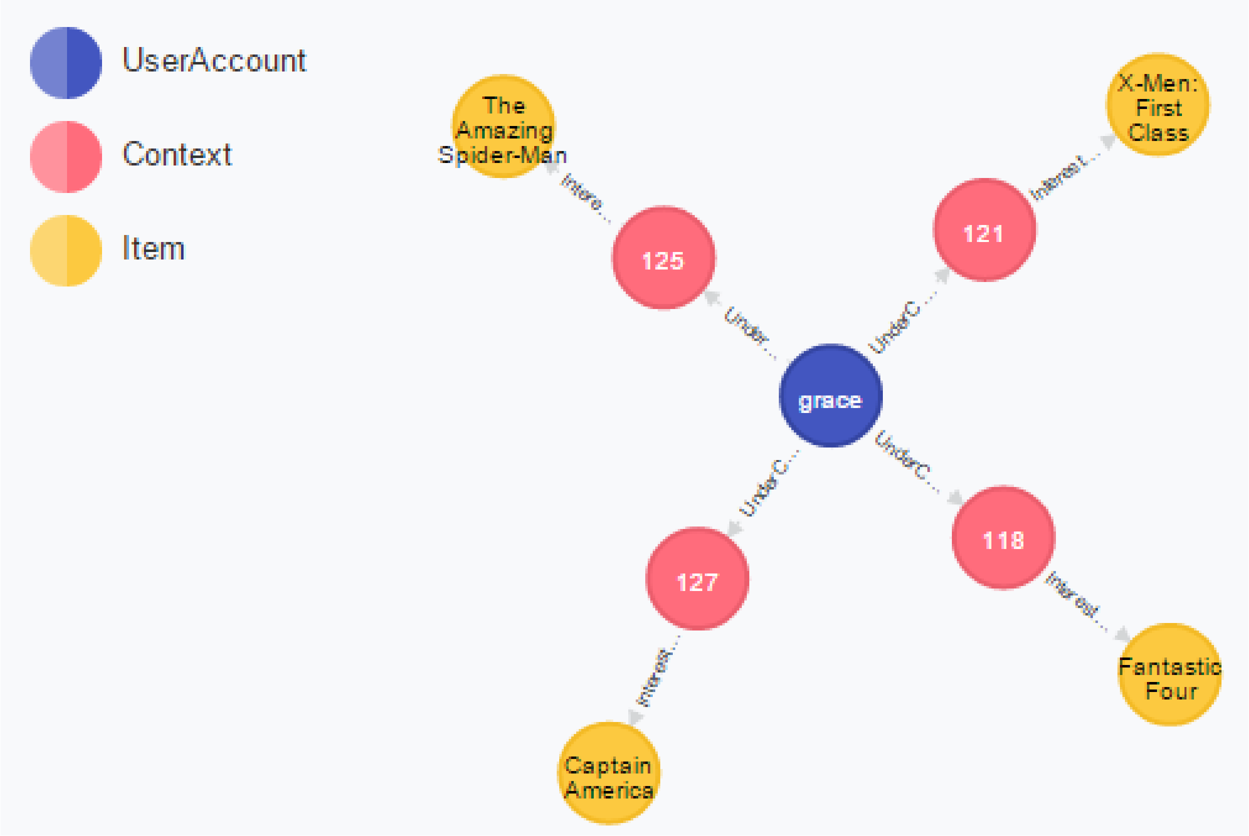


Figure 6.15: User Profile with Brother

CHAPTER 7

USER PROFILE HYPERNETWORK

Personalization is inevitable in the information overload era we live in. To address this problem, there are many personalization services available. Their purposes might differ and they might operate on different environments including mobile devices which does not support large memory requirements. We aim to provide these services a tailored user profile based on the service's needs. Our usage scenario is as follows: The personalized service requests a user profile by stating its needs. We call current needs of a service as its context. Based on provided context, we tailor user model and send this tailored profile to the personalized service. The personalized service uses this tailored user model and a set of simple rules to personalize. The key idea here is to show that since we provided only the most relevant parts of the user model, even a simple set of rules is enough to personalize.

In this section, we present the hypernetwork and tailoring methodology. Before presenting the user profile hypernetwork solution, we provide the background knowledge for hypergraphs and hypernetworks. Then we introduce the approach to construct a multi-level hypernetwork user model and propose the methodology to dynamically tailor the user profile.

7.1 Hypernetwork Preliminaries

A *hypergraph* is a generalized ordinary graph which allows edges to connect more than two vertices. Hypergraph theory is developed by Berge in 1960 by generalizing the graph theory [16, 15]. A more recent narration of hypergraph theory is clarified in [125, 22]. A *hypergraph* is a tuple $H = \langle V, E\rangle$, where V and E are sets of *vertices* and *hyperedges* respectively. Each hyperedge is a set of vertices, $E \subseteq \{\{u, v, ...\} \in \{P(V) - \{\emptyset\}\}\}$ where $P(V)$ indicates power set of V. For instance, for narration *"User u opens browser, searches for terms t_1t_2, clicks on urls url_1, url_2, url_3"* can be represented as a hypergraph as follows:

$H = \langle V, E\rangle$

$V = \{u, t_1, t_2, url_1, url_2, url_3\}$ is set of vertices

$E = \{\{Users, u\}, \{Terms, t_1, t_2\}, \{Urls, url_1, url_2, url_3\},$
$\{ProfileOfUser_1, u, t_1, t_2, url_1, url_2, url_3\}\}$

is set of hyperedges. Although hypergraph is capable of representing this narration, since it is set-theoretic structure, order of entities and how entities relate to each other in hyperedges is lost. However, order of terms and order of url clicks might be important for personalization algorithm which is going to run on the user model. Therefore, we employed hypernetworks which preserves the order of entities and the relations between entities.

Hyperedges are represented with *sets* in hypergraphs. On the other hand, hypernetworks use a more complex structure to represent them: hypersimplices. Technical background for hypersimplices [59] is summarized as follows: Given a set of vertices V, any subset of V , $\{v_0, v_1, .., v_p\}$ determines an object called *abstract p-simplex* which can be represented by a p-dimensional polyhedron in $(p + k)$-dimensional space, where $k \geq 0$. Simplices have a geometric representation as polyhedra in multi-dimensional space. For example, a simplex with three vertices is a triangle in 2-dimensional space and a simplex with four vertices is a tetrahedron in 3-dimensional space. Term *face* is used to define $(p - 1)$ dimension components of a p-simplex. For instance, the 2-dimensional faces of a 3-dimensional tetrahedron are triangles. A set of simplices with all their faces is called a *simplicial complex*. A simplex extended by its relation is called a *hypersimplex*. In a hypersimplex, since how entities are related is also involved, order is preserved. For instance, $\{a, b, c\}$ and $\{c, a, b\}$ represent same sets. When represented with hypersimplex, since relation of entities is also modeled, they indicate different hypersimplices: $\{R_{abc}, a, b, c\}$ and $\{R_{cab}, c, a, b\}$. A set of hypersimplices is called a *hypernetwork*.

In hypernetwork, shared faces represent connectivity. Two simplices are q*-near* if they share a q-dimensional face. Highest dimensional shared face is considered for defining q*-nearness*. For instance, let us assume *"User* u_1 *likes movies* m_1, m_2 *and* m_3*; User* u_2 *likes movies* m_2, m_4 *and* m_5 *and User* u_3 *likes movies* m_1, m_2, m_3 *and* m_5*"*. Users u_1 and u_2 both like movie m_2; users u_2 and u_3 both like movies m_2 and m_5; and users u_1 and u_3 both like movies m_1, m_2 and m_3. Therefore, users u_1 and u_2 are 1-near, users u_2 and u_3 are 2-near and users u_1 and u_3 are 3-near. If two simplices are connected through a chain of simplices and each simplex in the chain is at least q-near to its neighbours, then these two simplices are q*-connected.* In the example, users are 1-connected. Q*-analysis* technique provides a list of clusters of the hyperedges for each dimension q. In other words, the analysis clusters the hypernetwork by grouping hyperedges which share q vertices. Some hyperedges might contain different vertices which are not contained in other hyperedges. These hyperedges are $eccentric$. $Eccentricity$ is the ratio of number of vertices that are not shared to the total number of vertices in the hyperedge. Relatively disconnected simplices provide more $eccentricity$ than highly connected hyperedges. Therefore, removing eccentric hyperedges results in more information loss than removing highly connected hyperedges.

7.2 Principals and Justification

In this thesis, we expect our user model to be able to represent narrations about the individual correctly. Narrations consist of statements. Statements state n-ary relations between entities. In some situations, order of entities and how entities are related with each other in an n-ary relation might be important. We also aim to support the capability of dynamically tailoring the user model.

We use hypernetworks to model the user, because *(i)* they are capable of representing n-ary relations, *(ii)* they preserve order of entities and how entities relate with each other while representing relations and *(iii)* they enable dynamical tailoring by using their topological properties.

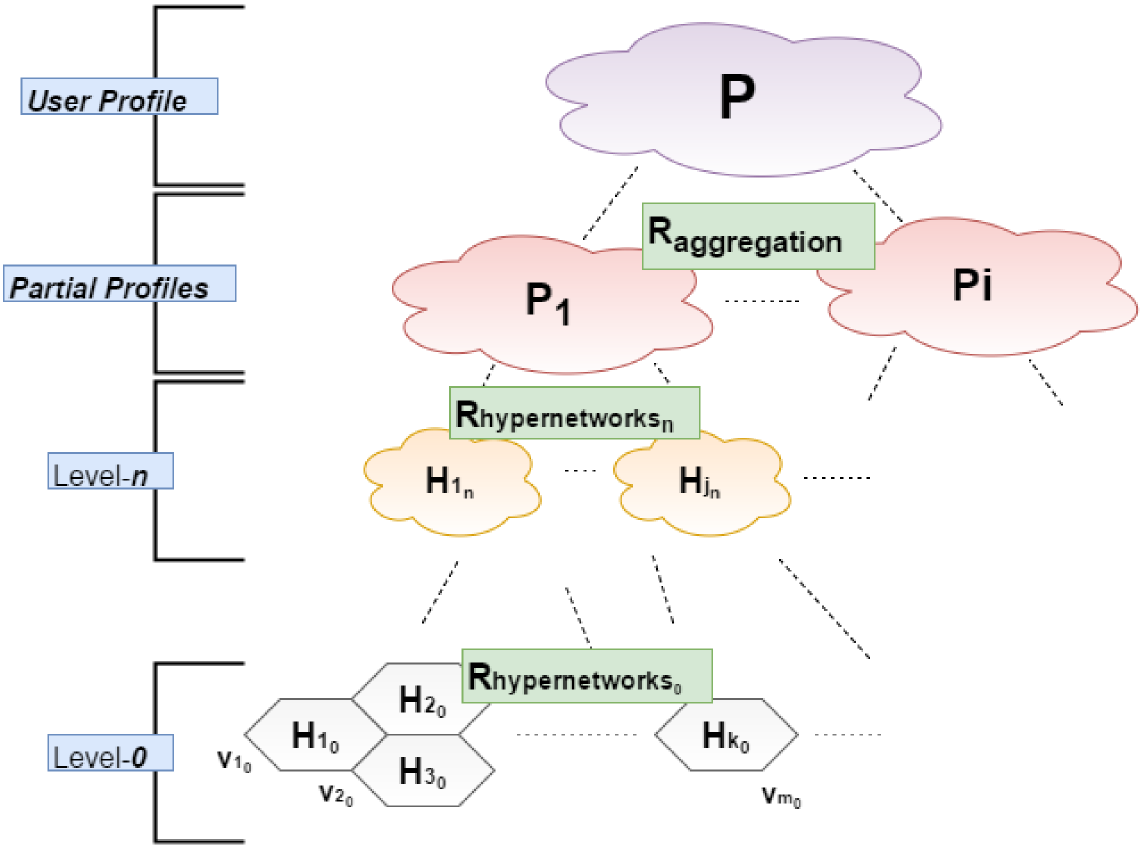


Figure 7.1: User Hypernetwork Multi-Level Design

An ordinary graph is good at representing binary relations. However, they cannot represent n-ary relations. A hypergraph is able to represent them. However, in hypergraphs, hyperedges are sets. Sets package items like a bag, so order is not preserved. Therefore, hypergraphs cannot represent n-ary relations in which order is important. On the other hand, hypernetworks are capable of representing n-ary relations by preserving the order of entities. Besides, Q-Analysis technique provides a list of hyperedge clusters by grouping hyperedges which share q vertices. This list enables tailoring on the hypernetwork by picking the hyperedges which are in the most relevant clusters.

We define the user model as a multi-level hypernetwork as in Figure 7.1. P represents the user model for the user u. Let us represent user profile with tuple $< u, P >$. Profile P is constructed by aggregating partial profiles $\{P_1, .., P_i\}$ of the user. This is represented with tuple $< P, R_{aggregation}, < P_1, w_1 >, .., < P_i, w_i >>$ where $R_{aggregation}$ indicates that the partial profiles are related with *aggregation* relation and w_m indicates weight for its corresponding partial profile P_m. A partial profile is a union of hypernetworks that represent the user at the highest, most general level. Tuple $< P_m, R_{hypernetworks_n}, H_{1_n}, .., H_{j_n} >$ represents partial profile P_m. $R_{hypernetworks_n}$ indicates that vertices are related with $hypernetworks_n$ relation, which is union of hypernetworks at level n. Vertices H_{i_n} stand for hypernetworks at level n. A hypernetwork at level i might reuse hypernetworks at level $(i - 1)$ when $i > 0$. A hypernetwork at the lowest, the most specialized level is an oriented and ordered composition of a set of vertices. Tuple $< H_{i_0}, R_w, v_{1_0}, ..., v_{e_0} >$ represents a hypernetwork at level-0. In the tuple, H_{i_0} indicates a hypernetwork at level-0, R_w stands for the hyperedge relation and v_{k_0} shows vertices at level-0.

In this thesis, one of our goals is to support several personalized services. Since personalized services focus on different domains and have different purposes, they might require different parts of the user model. To address this, we illustrate how to aggregate a holistic user model from distributed partial profiles of the individual in profile aggregation case study. How we support a personalized service is demonstrated in personalized search case study.

In personalized search case study, the simplified flow is as follows: *(1)* Session starts when the user opens browser and enters search engine web page, *(2)* User enters terms for the current query, *(3)* User clicks some of the returned URLs and examine them, *(4)* User repeats steps $2 - 3$ as many times as he/she wants *(5)* User ends the session by closing the browser. At level-0 we relate *terms* to form *query* hyperedges. At level-1, we model *sessions* by relating $query$ hyperegdes that are issued in the same session. At level-2, combination of *sessions* forms a partial profile. In this case study, we have one partial profile. Therefore, user profile is equal to level-2 profile.

7.3 Dynamic User Profile Tailoring

Dynamic user profile tailoring based on the given query means reducing the size of the profile by filtering only relevant hyperedges for the given query. The tailored user profile is lighter and more focused on the given query, since irrelevant hyperedges are eliminated. First, the multi-level hypernetwork user model is clustered. The clustering starts at the lowest level and continues up to the highest level. Q-Analysis technique is used to cluster and eccentricity determines the termination condition for the process. Then, the cluster for the given query is discovered at the lowest level. Finally, the union of clusters which given query belongs to at the lowest level and clusters which contain vertices and hyperedges from these clusters at higher levels forms the tailored user model.

Figure 7.2 illustrates a Q-Analysis process. The figure shows a Venn diagram of four sets. Each set represents existence of vertices named as *hasPink*, *hasBlue*, *hasYellow* and *hasGreen*. In Q-Analysis, the shared faces between hyperedges indicate similarity. The more faces they share, the more similar the hyperedges are. Let us assume each region which is constructed by the intersection of sets represents a hyperedge which has the vertices represented by them. For instance, region G has vertices *hasYellow* and *hasGreen* but does not have *hasBlue* or *hasPink*. Similarly, region T contains all four vertices, since it is located in the intersection of all sets.

In the example, region A shares one vertex with other regions, which is *hasPink*. However, region N shares three vertices, which are *hasPink*, *hasYellow* and *hasGreen*. Therefore, *q-shared* is equal to 1 for region A and 3 for region N. In the figure, this is illustrated for all regions with qq labels.

Eccentricity is a metric which measures how much new information is provided by a hyperedge. It is calculated by a simple formula

ecc = (dimension - q-shared) / (dimension)

where *dimension* shows the total number of vertices in the hyperedge and *q-shared* equals to the number of shared vertices with other hyperedges. For the example, let us assume each region has a dimension of 10. This means region A has 9 more vertices other than the four vertices we focus on the example. Then its eccentricity is 0.9. Similarly, eccentricity of region N is 0.7. Region T has the lowest eccentricity value, which is 0.6, since it contains the highest number of shared vertices. As a result, region A provides more information than N which provides more information than T.

Q-Analysis technique checks all hyperedges for the existence of q shared vertices. Since q is not predefined, clustering consists of iterations at each q where $0 < q <$

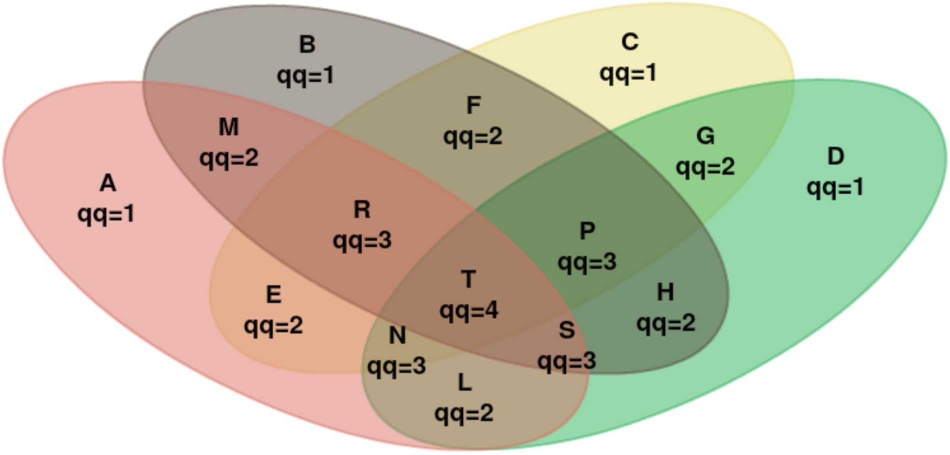


Figure 7.2: Q-Analysis Example

number of vertices in the largest hyperedge. Therefore, it is an expensive operation. We optimize it by using a predefined eccentricity threshold as termination condition. When there exist clusters with eccentricity value at least equal to the defined threshold, clustering is terminated.

In the example, Q-Analysis starts with q initiated as 10, since our assumption states that all hyperedges consist of 10 vertices. There are not any hyperedges which share 10 vertices, therefore iteration continues by decrementing q to 9. This process continues until q is equal to 3. At $q = 3$, clusters $\{\{T, R\}, \{T, N\}, \{T, P\}, \{T, S\}\}$ are formed. Eccenticity for cluster $\{T, R\}$ is calculated as follows: *Dimension* is equal to $10 + 10 - 3 = 17$ since shared vertices of R are already counted in $T.q - shared$ is equal to 4. Eccentricity is 0.76. Same calculation applies to other clusters. If our eccentricity threshold is 0.76 or below, we can stop clustering process. Other hyperedges are considered to be separate clusters of size 1. Since eccentricity of a separate, disconnected cluster is 1 by default, we do not consider their eccentricity. If the eccentricity threshold is higher then 0.76, the iteration should continue with $q = 2$. The clusters are $\{\{M, R, T\}, \{F, R, T\}, \{F, P, T\}, \{H, P, T\}, \{H, S, T\}, \{L, S, T\}, \{L, N, T\},$

$\{E, N, T\}, \{E, R, T\}\}$. Eccentricity is 0.84 for cluster $\{M, R, T\}$. Other clusters have similar eccentricity values, since we assumed all hyperedges contain 10 vertices. Therefore, if we define an eccentricity threshold 0.84 or below, the clustering terminates. If we define an eccentricity threshold higher than 0.84, the clustering should continue with $q = 1$. As illustrated, defining a lower eccentricity value significantly reduces the complexity of clustering by eliminating further iterations.

The value for eccentricity threshold is determined for the case study by trial and error. During personalized search case study, we conducted experiments with different eccentricity thresholds. We started with a low threshold value and executed evaluation by increasing threshold a little bit. When we observed that NDGC score remains same for eccentricity threshold 0.3 and higher threshold values, we picked 0.3 as threshold. For other case studies or datasets, this value should be redefined with trial and error, since it is specific to the dataset. Defining a generic algorithm to determine eccentricity threshold is left as future work.

CHAPTER 8

PERSONALIZED SEARCH: EVALUATION AND DISCUSSION

While searching for terms using a search engine, users' intentions might differ based on their user profiles. For instance, when term *apple* is searched, a chef expects to see apple recipes whereas a computer scientist looks for company Apple related news. Personalized search aims to retrieve the most relevant URLs at higher ranks in search results. There are several approaches for this. Query can be expanded with extra terms to reduce ambiguity. For instance, when *apple* is expanded as *apply pie* or *apply company*, ambiguous results are eliminated. Another approach is reordering the URL list which is returned by the search engine based on relevance. In this case study, we follow this approach.

Yandex organized a personalized web search challenge on Kaggle at 2014 [1]. The challenge aimed to re-rank web documents using personal preferences. In this section, we introduced personalized search implementation details and evaluation results based on this dataset.

8.1 Implementation Details

We construct a hypernetwork user model by using multi-layer approach to provide a solution for personalized search. We take *terms* and *URLs* as the basic building blocks. They are the lowest, most specialized level in the design. *Queries* are the next higher level consisting of a set of terms and returning a set of URLs. *Click events*

[1] Yandex Web Search Challenge on Kaggle, https://www.kaggle.com/c/yandex-personalized-web-search-challenge

are also at the same level as queries and they model the clicked URLs with dwell time information. *Sessions* consist of queries and click events and they represent the highest, most generalized level in the design.

The approach in personalized search is following the introduced design principals. At first step, terms at lowest level are clustered using Q-Analysis. Eccentricity threshold for clustering is 0.3. This value is determined by trial and error. When the clusters exhibit an eccentricity greater than or equal to the threshold, clustering is terminated. This is applied to reduce the time spent on clustering and prevent generation of many clusters. At next step, by using *clustered terms*, queries at higher level are clustered using the same methodology. Afterwards, sessions are clustered using *clustered queries*. At that point, we built a summarized view of the user hypernetwork replacing the actual vertices with clusters.

The goal is to re-rank the ordering of URLs returned by the given query in test session, so that they are in descending order based on relevancy. Relevancy is decided by checking the dwell time user spent on a clicked url. We found the session clusters which are similar to the test session and dynamically extracted a tailored user model for the test session. The tailored user model consists of sessions that are similar to the test session. By using the tailored model and simple heuristics, we re-ranked test queries. The heuristics are presented in Algorithms 5 and 6.

First, a relevancy table which represents URL's relatedness for sessions is prepared. The dataset stated that *(i)* if a user spent less than 50 time units on a URL, this URL is irrelevant, *(ii)* if the user spent more than 50 and less than 400 time units on the URL, it is relevant and *(iii)* if the user spent at least 400 time units on the URL, the URL is highly relevant. Also, the challenge assumes that the user quits a session after he/she finds what he/she is looking for. Therefore, the last clicked URL of each session is classified as highly relevant. Since the dataset provides domains for the URLs, we also applied same rules to domains and obtained domain relevancy table.

Algorithm 5: Heuristic: URL Relevancy

```
Result: URL and Domain Relevancy Table
initialization foreach session of user's sessions do
    foreach query in session do
        foreach URL/Domain in query's return list do
            if last URL in session then
                relevancy = HIGHLY RELATED
            else
                if time spent $< 50$ then
                    relevancy = NOT RELATED
                else if time spent $< 400$ then
                    relevancy = RELATED
                else
                    relevancy = HIGHLY RELATED
                end
            end
            if if relevancy for URL/Domain already exists then
                use highest relevancy assigned
        end
    end
end
```

Afterwards, query clusters for given query are located. These query clusters are included in the tailored user profile. At the higher level, session clusters which cover these query clusters are located. These session clusters are also included in the tailored profile. We assign default relevancy as relevant, since we do not want to miss any relevant URLs. We examine URL and domain relevance tables and if we find that the URL or domain is classified as highly relevant, we update URL's relevancy.

In summary, the heuristic is very simple. URL and domain relevancy table is prepared according to dataset's own specifications. The algorithm re-ranks a URL higher only when there is strong evidence about the URL's relevance. However, since we apply the heuristic on the tailored user profile instead of the entire user profile, it is effective.

Algorithm 6: Heuristic: Re-Ranking

```
Result: Re-Ranked URL lists for test queries
initialization foreach session of user's sessions do
    foreach cluster that current session belongs to do
        foreach cluster that test session belongs to do
            if clusters match then
                add current session to list of similar sessions for given query
        end
    end
end
foreach session in similar sessions list do
    foreach query in current session do
        add query to list of similar queries for given query
    end
end
foreach query in similar queries list do
    foreach URL/Domain in current query do
        add URL/Domain relevancy to Tailored URL/Domain Relevancy Table
    end
end
foreach URL returned by given query do
    default relevancy = RELEVANT if Url relevancy is defined in Tailored
      URL Relevancy table and higher than current relevancy then
        update relevancy
    if Domain relevancy is defined in Tailored Domain Relevancy table and
      higher than current relevancy then
        update relevancy
end
ReRank given query URLs by ordering by Relevancy, then by current rank
```

8.2 Evaluation Dataset and Methodology

Yandex provides user sessions extracted from logs containing one month of search activities in a large city. Sessions are fully anonymized and they contain user ids, queries, query terms, URLs, their domains, URL rankings and clicks. The size of the training set is around 16 GB, containing over 167 million records. The dataset is large with 21 million unique queries, 703 million unique URLs, more than 5 million unique users, over $64,5$ million clicks in training data, $34,5$ million training sessions and 797 thousand test sessions in the dataset. 27 days are training data and remaining 3 days are left for testing purposes.

The time of each operation is available in dataset. Therefore, dwell time is extractable by checking the time difference with the previous record. The unit for time is not provided, but it is stated that dwell time less than 50 is classified as *irrelevant*, between 50 and 400 as *relevant* and more than 400 as *highly relevant*. Also the last clicks for each session are considered to be highly relevant independent of the dwell time, since it is assumed that user found what he/she searched for.

The training dataset is stored on disk using *Lucene* [2] with an offline process which executed for about 11 hours. After that, we read in test sessions online. For each test user, we retrieved the user's previous sessions from Lucene and populated the multi-level user hypernetwork. The lowest level consists of terms, the higher level contains queries made up of terms and the highest level is a set of sessions containing these queries. We clustered the hypernetwork from the lowest level to the highest level. Then, we discovered similar clusters for the test session and dynamically extracted the tailored user profile for the test session. Finally, using the tailored profile and few simple heuristics, we re-ranked the URLs for the given query. We repeated this step with different set of heuristics 36 times to ensure that the result is not by chance. The online process is slightly over than 1,5 hours on an ordinary computer with $8GB$ Ram and Intel Core i5 processor for the entire test dataset. It takes only seconds per user which means that the proposed model is able to provide a tailored user model for personalized service real time.

The evaluation metric for this competition is normalized discounted cumulative gain (NDCG) @k where k=10. The NDCG is calculated as :

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i}-1}{\log_2(i+1)}$$

$$nDCG_k = \frac{DCG_k}{IDCG_k}$$

where rel_i indicates the relevance of the result at position i and $IDCG_k$ stands for the the maximum possible DCG for a given set of queries.

[2] Lucene, http://lucene.apache.org/core/

8.3 Evaluation Results

The dataset that we use is a real life dataset which can be stated as big data. We use two baselines to compare: *(i)* a trivial *random baseline* which randomly re-orders URLs to personalize and *(ii)* a non-trivial *non-personalized baseline* which uses Yandex's original URL ordering. The second baseline is non-trivial since it already performs well. Therefore, any little improvement on this baseline is a success. We did not perform statistical significance test, since it can be dangerous when analyzing weak effects in big data [51]. The aim of statistical significance is not indicating that a finding is important or that an effect is big; it aims to show that the effect is clearly visible by measuring how confident we can be that a result isn't due to random noise [104]. To make sure that our result is not by coincidence, we performed the test by using different set of simple rules 36 times. All test cases outperformed the non-trivial baseline. In this thesis, we presented the test case which performed best.

Our goal is providing a tailored user model to personalized services which contains only the most related data about the user for their use case. So, they can achieve effective personalization just by applying simple heuristics on provided user model. We also aim to achieve this in real time. In this case study, we demonstrate that we can provide a tailored user model to a personalized search service based on the given test query in real time, and simulate that the personalized service is able to achieve a better URL ordering for the individual than the search engine's own URL ordering by applying a simple set of rules on provided user model.

Since our aim is providing a tailored user profile for personalized services in real time, we did not use any approach based on predictive statistical models. They cannot operate in real time, since they require a long training time. Moreover, they require selection of features which adds extra complexity. For instance, the winner of the challenge uses a statistical approach which requires 4 days of training with their powerful company computers and their key point is using a complicated algorithm to select correct features to use[75]. Moreover, these approaches can not be generalized to other personalized services easily. Our aim is to support several personalized services in the same generic way: providing a tailored user model which can be effective even with simple set of rules defined by the personalized service. Even though we

Table 8.1: Personalized Search Evaluation Results

Evaluation	**Public Board Score (NDCG)**	**Private Board Score (NDCG)**	**Calculation Time**
Best Statistical Approach	0.80647	0.80714	**not real time, requires offline training time**
Tailored User Model with Q-Analysis and Eccentricity	**0.79081**	**0.79153**	**real time**
Non-Personalized Baseline	0.79056	0.79133	real time
No Tailoring Applied	0.78806	0.78869	real time
Random Baseline	0.47972	0.47954	real time

showed personalized search case study in this thesis, the solution can be reused for other personalized services easily.

We also tested by eliminating the tailoring behavior, to isolate the effects of tailoring. In fact, without using the tailoring algorithm, our hypernetwork is equivalent to a hypergraph. Therefore, in this way, we compared our proposed algorithm to a hypergraph approach. This case performed worse than *non-personalized baseline*.

The results are summarized in Table 8.1. The score for the *random baseline* which is obtained by randomly re-ranking the URLs is 0.47972. The *non-personalized baseline* which is Yandex's own algorithm performs very well, 0.79056. In fact, in the competition, half of the competitors could not pass this score. We tried 36 times by using the proposed algorithm with different heuristics and all of them outperformed the non-personalized baseline. Our best score is 0.79153. We also evaluated when no tailoring applied to the model. Tailored model performed better than non-tailored model. Non-tailored model did slightly worse than non-personalized baseline. This shows that tailoring the model for test query and founding the decision on the most relevant part of the individual's profile is working.

[75] won the competition with score 0.80647. However, they used complex statistical methods, defined a number of features and they needed to train the system for four days. We obtained this score by applying simple heuristics on the dynamically tailored user hypernetwork and evaluation process is about $1,5$ hours for the entire test sessions.

CHAPTER 9

CONCLUSION AND FUTURE WORK

In this thesis, we proposed a hypergraph based user modeling framework. We defined an aggregation approach which disambiguates entities, discovers domains of the disambiguated entities and applies semantic enhancement to integrate partial profiles coming from different information sources into a holistic, multi-domain user model.

During semantic enhancement phase of aggregation, we use an external knowledge base via a middle ontology and configured the use of middle ontology according to the user modeling domain. We only used properties in the middle ontology such as *ContributesTo, Creates, SuperclassOf etc.* that are relevant to the user modeling domain.

The main objective of the aggregation is to provide a user profile for user modeling domain applications such as recommendation. Most of the user modeling domain applications are connected data problems which can be converted into graph traversal problems. Graphs naturally support connected data problems. Hypergraphs are capable of representing higher order relations whereas ordinary graphs are limited to pairwise relationships. However, hypergraphs are complicated in terms of implementation.

Property graphs are equivalent to hypergraphs and they make graph traversal algorithms easier by providing filtering mechanisms such as node labels and edge types. In other words, it is possible to write traversal algorithms specific to a label or an edge type without traversing irrelevant nodes or edges in the hypergraph.

We implemented a recommender system, FunGuide as case study. FunGuide uses the proposed user model framework and is capable of constructing a semantic user profile, making domain based, cross domain and general recommendations. The case study also supports discovery of potential users who might be interested in a given item, computation of the user's interest in an item and discovery of similar users.

We showed how the proposed model is extended to support context.

We extensively evaluated the user model. During evaluation, we showed that the system could predict future interests of the user with very high recall scores.

As future work, the following could be accomplished:

- The extended version of the proposed hypergraph based user modeling framework which supports context information may be implemented and FunGuide interfaces and queries may be also extended to support context information.
- Users could be categorized according to social web usage habits. Evaluation results may change between different group of users.
- User model should maintain long term and short term user profiles separately.
- Freebase is retired. The system may be defined by using another knowledge base such as Wikidata which replaces Freebase.
- The system could be extended with the feature of discovery of social web accounts of the individual.
- The system could be extended to support other social web accounts. Similarly, algorithms to extract partial profiles from social accounts could be improved.
- Handcarfted rules for managing conflicting information from partial profiles could be defined and implemented.

REFERENCES

[1] Ahmad Abdel-Hafez and Yue Xu. A survey of user modelling in social media websites. *Computer and Information Science*, 6(4):p59, 2013.

[2] Fabian Abel, Claudia Hauff, Geert-Jan Houben, and Ke Tao. Leveraging user modeling on the social web with linked data. In *Web Engineering*, pages 378–385. Springer, 2012.

[3] Fabian Abel, Nicola Henze, Eelco Herder, and Daniel Krause. Interweaving public user profiles on the web. In *User Modeling, Adaptation, and Personalization*, pages 16–27. Springer, 2010.

[4] Fabian Abel, Nicola Henze, Eelco Herder, and Daniel Krause. Linkage, aggregation, alignment and enrichment of public user profiles with mypes. In *Proceedings of the 6th International Conference on Semantic Systems*, page 11. ACM, 2010.

[5] Fabian Abel, Eelco Herder, Geert-Jan Houben, Nicola Henze, and Daniel Krause. Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction*, 23(2-3):169–209, 2013.

[6] Fabian Abel, Eelco Herder, and Daniel Krause. Extraction of professional interests from social web profiles. *Proc. UMAP*, 34, 2011.

[7] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.

[8] Lora Aroyo and Geert-Jan Houben. User modeling and adaptive semantic web. *Semantic Web*, 1(1):105–110, 2010.

[9] Fabio A Asnicar and Carlo Tasso. ifweb: a prototype of user model-based intelligent agent for document filtering and navigation in the world wide web. In *Sixth International Conference on User Modeling*, pages 2–5, 1997.

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

[11] Shumeet Baluja, Rohan Seth, D Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. Video suggestion and dis-

covery for youtube: taking random walks through the view graph. In *Proceedings of the 17th international conference on World Wide Web*, pages 895–904. ACM, 2008.

[12] Michal Barla. Interception of user's interests on the web. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 435–439. Springer, 2006.

[13] Michal Barla and Mária Bieliková. Ordinary web pages as a source for metadata acquisition for open corpus user modeling. *Proc. of IADIS WWW/Internet*, 2010, 2010.

[14] Paul N Bennett, Ryen W White, Wei Chu, Susan T Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. Modeling the impact of short-and long-term behavior on search personalization. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 185–194. ACM, 2012.

[15] Claude Berge. *Hypergraphs: combinatorics of finite sets*, volume 45. Elsevier, 1984.

[16] Claude Berge and Edward Minieka. *Graphs and hypergraphs*, volume 7. North-Holland publishing company Amsterdam, 1973.

[17] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. Cross-representation mediation of user models. *User Modeling and User-Adapted Interaction*, 19(1-2):35–63, 2009.

[18] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014.

[19] 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. ACM, 2008.

[20] Kurt D. Bollacker, Robert P. Cook, and Patrick Tufts. Freebase: A shared database of structured general human knowledge. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 1962–1963, 2007.

[21] Kalina Bontcheva and Dominic Rout. Making sense of social media streams through semantics: a survey. *Semantic Web*, 5(5):373–403, 2014.

[22] Alain Bretto. *Hypergraph Theory: An Introduction*. Springer Science & Business Media, 2013.

[23] Jiajun Bu, Shulong Tan, Chun Chen, Can Wang, Hao Wu, Lijun Zhang, and Xiaofei He. Music recommendation by unified hypergraph: combining social media information and music content. In *Proceedings of the international conference on Multimedia*, pages 391–400. ACM, 2010.

[24] Silvia Calegari and Gabriella Pasi. Personal ontologies: Generation of user profiles based on the yago ontology. *Information processing & management*, 49(3):640–658, 2013.

[25] Javier Calle, Leonardo Castaño, Elena Castro, and Dolores Cuadra. Statistical user model supported by r-tree structure. *Applied intelligence*, 39(3):545–563, 2013.

[26] Iván Cantador, Ignacio Fernández-Tobías, Shlomo Berkovsky, and Paolo Cremonesi. Cross-domain recommender systems. In *Recommender Systems Handbook*, pages 919–959. Springer, 2015.

[27] David Carmel, Naama Zwerdling, Ido Guy, Shila Ofek-Koifman, Nadav Har'El, Inbal Ronen, Erel Uziel, Sivan Yogev, and Sergey Chernov. Personalized social search based on the user's social network. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1227–1236. ACM, 2009.

[28] Leonardo Castaño, Francisco Javier Calle, Dolores Cuadra, and Elena Castro. User modeling for human-like interaction. In *The 2nd international workshop on user modeling and adaptation for daily routines (UMADR)*, pages 23–34, 2011.

[29] Federica Cena, Silvia Likavec, and Francesco Osborne. Property-based interest propagation in ontology-based user model. In *User Modeling, Adaptation, and Personalization*, pages 38–50. Springer, 2012.

[30] Federica Cena, Silvia Likavec, and Francesco Osborne. Anisotropic propagation of user interests in ontology-based user models. *Information Sciences*, 250:40–60, 2013.

[31] Bisheng Chen, Jingdong Wang, Qinghua Huang, and Tao Mei. Personalized video recommendation through tripartite graph propagation. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 1133–1136. ACM, 2012.

[32] Bisheng Chen, Jingdong Wang, Qinghua Huang, and Tao Mei. Personalized video recommendation through tripartite graph propagation. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 1133–1136. ACM, 2012.

[33] Liren Chen and Katia Sycara. Webmate: a personal agent for browsing and searching. In *Proceedings of the second international conference on Autonomous agents*, pages 132–139. ACM, 1998.

[34] Terence Chen, Mohamed Ali Kaafar, Arik Friedman, and Roksana Boreli. Is more always merrier?: a deep dive into online social footprints. In *Proceedings of the 2012 ACM workshop on Workshop on online social networks*, pages 67–72. ACM, 2012.

[35] Marek Ciglan and Kjetil Nørvåg. Sgdb–simple graph database optimized for activation spreading computation. In *Database Systems for Advanced Applications*, pages 45–56. Springer, 2010.

[36] . CSC Leading Edge Forum. Data revolution. Technical report, 2011.

[37] Mariam Daoud, Lynda-Tamine Lechani, and Mohand Boughanem. Towards a graph-based user profile modeling for a session-based personalized search. *Knowledge and Information Systems*, 21(3):365–398, 2009.

[38] Mariam Daoud, Lynda Tamine, and Mohand Boughanem. A personalized graph-based document ranking model using a semantic user profile. In *User Modeling, Adaptation, and Personalization*, pages 171–182. Springer, 2010.

[39] Mariam Daoud, Lynda Tamine, and Mohand Boughanem. A personalized search using a semantic distance measure in a graph-based ranking model. *Journal of Information Science*, 37(6):614–636, 2011.

[40] Elena Demidova, Iryna Oelze, and Wolfgang Nejdl. Aligning freebase with the yago ontology. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 579–588. ACM, 2013.

[41] Lubos Demovic, Eduard Fritscher, Jakub Kriz, Ondrej Kuzmik, Ondrej Proksa, Diana Vandlikova, Dusan Zelenik, and Maria Bielikova. Movie recommendation based on graph traversal algorithms. In *DEXA Workshops*, pages 152–156, 2013.

[42] 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. ACM, 2012.

[43] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th international conference on World Wide Web*, pages 581–590. ACM, 2007.

[44] Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd*

International Conference on Computational Linguistics, pages 277–285. Association for Computational Linguistics, 2010.

[45] Patrik Floréen, Michael Przybilski, Petteri Nurmi, Johan Koolwaaij, Anthony Tarlano, Matthias Wagner, Marko Luther, Fabien Bataille, Mathieu Boussard, Bernd Mrohs, et al. Towards a context management framework for mobilife. *Proc. 14th IST Mobile & Wireless Summit*, 2005:20–28, 2005.

[46] Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. Directed hypergraphs and applications. *Discrete applied mathematics*, 42(2):177–201, 1993.

[47] Giorgio Gallo and Maria Grazia Scutella. Directed hypergraphs as a modelling paradigm. *Rivista di matematica per le scienze economiche e sociali*, 21(1-2):97–123, 1998.

[48] Susan Gauch, Mirco Speretta, Aravind Chandramouli, and Alessandro Micarelli. User profiles for personalized information access. In *The adaptive web*, pages 54–89. Springer, 2007.

[49] M Rami Ghorab, Dong Zhou, Alexander O'Connor, and Vincent Wade. Personalised information retrieval: survey and classification. *User Modeling and User-Adapted Interaction*, 23(4):381–443, 2013.

[50] Riddhiman Ghosh and Mohamed Dekhil. Mashups for semantic user profiles. In *Proceedings of the 17th international conference on World Wide Web*, pages 1229–1230. ACM, 2008.

[51] Robert Grossman. The dangers of statistical significance when studying weak effects in big data, 2017.

[52] Ido Guy, Uri Avraham, David Carmel, Sigalit Ur, Michal Jacovi, and Inbal Ronen. Mining expertise and interests from social media. In *Proceedings of the 22nd international conference on World Wide Web*, pages 515–526. International World Wide Web Conferences Steering Committee, 2013.

[53] Per Hage and Frank Harary. Eccentricity and centrality in networks. *Social networks*, 17(1):57–63, 1995.

[54] Aniko Hannak, Piotr Sapiezynski, Arash Molavi Kakhki, Balachander Krishnamurthy, David Lazer, Alan Mislove, and Christo Wilson. Measuring personalization of web search. In *Proceedings of the 22nd international conference on World Wide Web*, pages 527–538. ACM, 2013.

[55] Benjamin Heitmann. An open framework for multi-source, cross-domain personalisation with semantic interest graphs. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 313–316. ACM, 2012.

[56] Benjamin Heitmann, Maciej Dabrowski, Alexandre Passant, Conor Hayes, and Keith Griffin. Personalisation of social web services in the enterprise using spreading activation for multi-source, cross-domain recommendations. In *AAAI Spring Symposium: Intelligent Web Services Meet Social Computing*, 2012.

[57] Qinghua Huang, Bisheng Chen, Jingdong Wang, and Tao Mei. Personalized video recommendation through graph propagation. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 10(4):32, 2014.

[58] Paridhi Jain and Ponnurangam Kumaraguru. Finding nemo: Searching and resolving identities of users across online social networks. *arXiv preprint arXiv:1212.6147*, 2012.

[59] Jeffrey Johnson. *Hypernetworks in the science of complex systems*, volume 3. World Scientific, 2013.

[60] JH Johnson. Some structures and notation of q-analysis. *Environment and Planning B: Planning and Design*, 8(1):73–86, 1981.

[61] Sung Young Jung, Jeong-Hee Hong, and Taek-Soo Kim. A statistical model for user preference. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):834–843, 2005.

[62] Pavan Kapanipathi, Fabrizio Orlandi, Amit Sheth, and Alexandre Passant. Personalized Filtering of the Twitter Stream. In *SPIM Workshop at ISWC 2011*, pages 6–13. CEUR-WS, 2011.

[63] Elisabeth Kapsammer, Stefan Mitsch, Birgit Pröll, Werner Retschitzegger, Wieland Schwinger, Manuel Wimmer, Martin Wischenbart, and Stephan Lechner. Towards a reference model for social user profiles: Concept & implementation. In *Proc. of the Int. Workshop on Personalized Access, Profile Management, and Context Awareness in Databases (PersDB)*, 2011.

[64] Tomáš Kramár, Michal Barla, and Mária Bieliková. Disambiguating search by leveraging a social context based on the stream of user's activity. In *User Modeling, Adaptation, and Personalization*, pages 387–392. Springer, 2010.

[65] Tomas Kramar, Michal Barla, and Mária Bieliková. Personalizing search using socially enhanced interest model built from the stream of user's activity. *J. Web Eng.*, 12(1&2):65–92, 2013.

[66] Kleanthi Lakiotaki, Nikolaos F Matsatsinis, and Alexis Tsoukias. Multicriteria user modeling in recommender systems. *IEEE Intelligent Systems*, 26(2):64–76, 2011.

[67] Ora Lassila and James Hendler. Embracing" web 3.0". *Internet Computing, IEEE*, 11(3):90–93, 2007.

[68] Lei Li and Tao Li. News recommendation via hypergraph learning: encapsulation of user behavior and news content. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 305–314. ACM, 2013.

[69] Lei Li, Li Zheng, and Tao Li. Logo: a long-short user interest integration in personalized news recommendation. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 317–320. ACM, 2011.

[70] Rui Li, Shengjie Wang, Hongbo Deng, Rui Wang, and Kevin Chen-Chuan Chang. Towards social user profiling: unified and discriminative influence model for inferring home locations. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1023–1031. ACM, 2012.

[71] Steffen Lohmann and Paloma Díaz. Representing and visualizing folksonomies as graphs: a reference model. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 729–732. ACM, 2012.

[72] Anshu Malhotra, Luam Totti, Wagner Meira Jr, Ponnurangam Kumaraguru, and Virgilio Almeida. Studying user footprints in different online social networks. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pages 1065–1070. IEEE Computer Society, 2012.

[73] Murat Manguoglu, Eric Cox, Faisal Saied, and Ahmed Sameh. *TRACEMIN-Fiedler: A Parallel Algorithm for Computing the Fiedler Vector*, pages 449–455. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[74] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[75] Paul Masurel, Kenji Lefèvre-Hasegawa, Christophe Bourguignat, and Matthieu Scordia. Dataiku's solution to yandex's personalized web search challenge. In *WSCD workshop*, volume 13, 2014.

[76] Nicolaas Matthijs and Filip Radlinski. Personalizing web search using long term browsing history. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 25–34. ACM, 2011.

[77] . McKinsey Global Institute. Big data: The next frontier for innovation, competition, and productivity. Technical report, 2011.

[78] Elke Michlmayr and Steve Cayzer. Learning user profiles from tagging data and leveraging them for personal (ized) information access. In *Proceedings of the Workshop on Tagging and Metadata for Social Information Organization, 16th International World Wide Web Conference (WWW2007)*, pages 1–7, 2007.

[79] . MIT Technology Review. Big data gets personal. Technical report, 2011.

[80] Folke Mitzlaff, Martin Atzmueller, Gerd Stumme, and Andreas Hotho. Semantics of user interaction in social media. In *Complex Networks IV*, pages 13–25. Springer, 2013.

[81] Alexandros Moukas. Amalthaea information discovery and filtering using a multiagent evolving ecosystem. *Applied Artificial Intelligence*, 11(5):437–457, 1997.

[82] Alexandros Moukas. User modeling in a multiagent evolving system. In *Proceedings, workshop on Machine Learning for User Modeling, 6 th International Conference on User Modeling, Chia Laguna, Sardinia*, 1997.

[83] Nicolas Neubauer and Klaus Obermayer. Towards community detection in k-partite k-uniform hypergraphs. In *Proceedings of the NIPS 2009 Workshop on Analyzing Networks and Learning with Graphs*, pages 1–9, 2009.

[84] Petteri Nurmi, Alfons Salden, Sian Lun Lau, Jukka Suomela, Michael Sutterer, Jean Millerat, Miquel Martin, Eemil Lagerspetz, and Remco Poortinga. A system for context-dependent user modeling. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, pages 1894–1903. Springer, 2006.

[85] Fabrizio Orlandi, John Breslin, and Alexandre Passant. Aggregated, interoperable and multi-domain user profiles for the social web. In *Proceedings of the 8th International Conference on Semantic Systems*, pages 41–48. ACM, 2012.

[86] Gizem Öztürk and Nihan Kesim Cicekli. A hybrid video recommendation system using a graph-based algorithm. In *Modern Approaches in Applied Intelligence*, pages 406–415. Springer, 2011.

[87] Till Plumbaum, Katja Schulz, Martin Kurze, and Sahin Albayrak. My personal user interface: A semantic user-centric approach to manage and share user information. In *Human Interface and the Management of Information. Interacting with Information*, pages 585–593. Springer, 2011.

[88] Pearl Pu, Li Chen, and Rong Hu. Evaluating recommender systems from the user's perspective: survey of the state of the art. *User Modeling and User-Adapted Interaction*, 22(4-5):317–355, 2012.

[89] Feng Qiu and Junghoo Cho. Automatic identification of user interest for personalized search. In *Proceedings of the 15th international conference on World Wide Web*, pages 727–736. ACM, 2006.

[90] Liana Razmerita, Rokas Firantas, and Martynas Jusevicius. Towards a new generation of social networks: Merging social web with semantic web. In *I-SEMANTICS*, pages 412–423, 2009.

[91] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Introduction to recommender systems handbook*. Springer, 2011.

[92] Ian Robinson, Jim Webber, and Emil Eifrem. *Graph databases*. " O'Reilly Media, Inc.", 2013.

[93] Marko A Rodriguez and Peter Neubauer. Constructions from dots and lines. *Bulletin of the American Society for Information Science and Technology*, 36(6):35–41, 2010.

[94] Marko A Rodriguez and Peter Neubauer. The graph traversal pattern. *arXiv preprint arXiv:1004.1001*, 2010.

[95] O Sacco, F Orlandi, and A Passant. Privacy aware and faceted user-profile management using social data. *Semantic Web Journal*, 2011.

[96] Márius Šajgalík, Michal Barla, and Mária Bieliková. Efficient representation of the lifelong web browsing user characteristics. In *Proc. of the 2nd Workshop on LifeLong User Modelling, in Conjunction with UMAP*, pages 21–30, 2013.

[97] Hidekazu Sakagami and Tomonari Kamba. Learning personal preferences on online newspaper articles from user behaviors. *Computer Networks and ISDN Systems*, 29(8):1447–1455, 1997.

[98] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM, 2002.

[99] Bracha Shapira, Lior Rokach, and Shirley Freilikhman. Facebook single and cross domain data for recommendation systems. *User Modeling and User-Adapted Interaction*, 23(2-3):211–247, 2013.

[100] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. Linking named entities in tweets with knowledge base via user interest modeling. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 68–76. ACM, 2013.

[101] Xuehua Shen, Bin Tan, and ChengXiang Zhai. Implicit user modeling for personalized search. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 824–831. ACM, 2005.

[102] Juan M Silva, Abu Saleh Md Mahfujur Rahman, and Abdulmotaleb El Saddik. Web 3.0: a vision for bridging the gap between real and virtual. In *Proceedings of the 1st ACM international workshop on Communicability design and*

evaluation in cultural and ecological multimedia system, pages 9–14. ACM, 2008.

[103] Georgios Siolas, George Caridakis, Phivos Mylonas, Spyridon Kollias, and Andreas Stafylopatis. Context-aware user modeling and semantic interoperability in smart home environments. In *Semantic and Social Media Adaptation and Personalization (SMAP), 2013 8th International Workshop on*, pages 27–32. IEEE, 2013.

[104] Noah Smith. Statistical significance is overrated, 2017.

[105] Humphrey Sorensen and Michael McElligott. Psun: a profiling system for usenet news. In *Proceedings of CIKM*, volume 95, pages 1–2, 1995.

[106] Micro Speretta and Susan Gauch. Personalized search based on user search histories. In *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*, pages 622–628. IEEE, 2005.

[107] Anna Stefani and C Strappavara. Personalizing access to web sites: The siteif project. In *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia HYPERTEXT*, volume 98, pages 20–24, 1998.

[108] Kazunari Sugiyama, Kenji Hatano, and Masatoshi Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th international conference on World Wide Web*, pages 675–684. ACM, 2004.

[109] Qi Suo, Shiwei Sun, Nick Hajli, and Peter ED Love. User ratings analysis in social networks through a hypernetwork method. *Expert Systems with Applications*, 42(21):7317–7325, 2015.

[110] Zareen Saba Syed and Tim Finin. Approaches for automatically enriching wikipedia. *Collaboratively-Built Knowledge Sources and AI*, 10:02, 2010.

[111] Shulong Tan, Jiajun Bu, Chun Chen, and Xiaofei He. Using rich social media information for music recommendation via hypergraph model. In *Social media modeling and computing*, pages 213–237. Springer, 2011.

[112] Hilal Tarakci and Nihan Cicekli. Ubiquitous fuzzy user modeling for multi-application environments by mining socially enhanced online traces. *User Modeling, Adaptation, and Personalization*, pages 387–390, 2012.

[113] Hilal Tarakci and Nihan Cicekli. Using hypergraph-based user profile in a recommendation system. In *International Conference on Knowledge Engineering and Ontology Development*, pages –. Scitepress, 2014.

[114] Hilal Tarakci and Nihan Cicekli. A hypergraph-based framework for representing aggregated user profiles (submitted). *Information sciences*, 2015.

[115] Hilal Tarakçi and Nihan Kesim Cicekli. UCASFUM: A ubiquitous context-aware semantic fuzzy user modeling system. In *KEOD 2012 - Proceedings of the International Conference on Knowledge Engineering and Ontology Development, Barcelona, Spain, 4 - 7 October, 2012.*, pages 278–283, 2012.

[116] Hilal Tarakci and Nihan Kesim Cicekli. A formal framework for hypergraph-based user profiles. In *Information Sciences and Systems 2014*, pages 285–293. Springer, 2014.

[117] Dieudonné Tchuente, Marie-Francoise Canut, Nadine Baptiste-Jessel, André Péninou, and Florence Sedes. A community based algorithm for deriving users' profiles from egocentrics networks. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pages 266–273. IEEE Computer Society, 2012.

[118] Jaime Teevan, Susan T Dumais, and Daniel J Liebling. To personalize or not to personalize: modeling queries with variation in user intent. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 163–170. ACM, 2008.

[119] Jaime Teevan, Meredith Ringel Morris, and Steve Bush. Discovering and using groups to improve personalized search. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 15–24. ACM, 2009.

[120] Antonis Theodoridis, Constantine Kotropoulos, and Yannis Panagakis. Music recommendation using hypergraphs and group sparsity. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 56–60. IEEE, 2013.

[121] Amit Tiroshi, Shlomo Berkovsky, Mohamed Ali Kaafar, Terence Chen, and Tsvi Kuflik. Cross social networks interests predictions based ongraph features. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 319–322. ACM, 2013.

[122] Amit Tiroshi, Tsvi Kuflik, Judy Kay, and Bob Kummerfeld. Recommender systems and the social web. In *Advances in User Modeling*, pages 60–70. Springer, 2012.

[123] Chris Van Aart, Lora Aroyo, Dan Brickley, Vicky Buser, Libby Miller, Michele Minno, Michele Mostarda, Davide Palmisano, Yves Raimond, Guus Schreiber, et al. The notube beancounter: aggregating user data for television programme recommendation. *Social Data on the Web (SDoW2009)*, 2009.

[124] Andrea Varga, Amparo Elizabeth Cano, Fabio Ciravegna, et al. Exploring the similarity between social knowledge sources and twitter for cross-domain topic

classification. *Knowledge Extraction and Consolidation from Social Media (KECSM 2012)*, page 78, 2012.

[125] Vitaly I Voloshin. *Introduction to graph and hypergraph theory*. Nova Science Publ., 2009.

[126] Xuan Truong Vu, Marie-Hélène Abel, and Pierre Morizet-Mahoudeaux. An aggregation model of online social networks to support group decision-making. *Journal of Decision Systems*, 23(1):24–39, 2014.

[127] Xuan-Truong Vu, Pierre Morizet-Mahoudeaux, and Marie-Hélène Abel. User-centered social network profiles integration. In *WEBIST*, pages 473–476. SciTePress, 2013.

[128] Ryen W White, Paul N Bennett, and Susan T Dumais. Predicting short-term interests using activity-based search context. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1009–1018. ACM, 2010.

[129] Martin Wischenbart, Stefan Mitsch, Elisabeth Kapsammer, Angelika Kusel, Birgit Pröll, Werner Retschitzegger, Wieland Schwinger, Johannes Schönböck, Manuel Wimmer, and Stephan Lechner. User profile integration made easy: model-driven extraction and transformation of social network schemas. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 939–948. ACM, 2012.

[130] Shuang-Hong Yang, Bo Long, Alex Smola, Narayanan Sadagopan, Zhaohui Zheng, and Hongyuan Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on World wide web*, pages 537–546. ACM, 2011.

[131] Xiao Yu, Hao Ma, Bo-June Paul Hsu, and Jiawei Han. On building entity recommender systems using user click log and freebase knowledge. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 263–272. ACM, 2014.

[132] YingSi Zhao and Bo Shen. Empirical study of user preferences based on rating data of movies. *PloS one*, 11(1):e0146541, 2016.

[133] Zhicheng Zheng, Xiance Si, Fangtao Li, Edward Y Chang, and Xiaoyan Zhu. Entity disambiguation with freebase. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 82–89. IEEE Computer Society, 2012.

[134] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in neural information processing systems*, pages 1601–1608, 2006.

[135] Ingrid Zukerman and David W Albrecht. Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction*, 11(1-2):5–18, 2001.

APPENDIX A

METASCHEMA PROPERTIES IN FREEBASE

The metaschema properties are listed in Table A.1.

Table A.1: Metaschema Properties

Abstract/Concrete	Adaptation	Administration
Broader/Narrower	Categorical	Certification
Character Appearance	Character Portrayal	Composition
Contribution	Creation	Discovery
Distribution	Event/Location	Exhibition
Fictional	Genre	Identifier
Leadership	Location	Means of Demise
Means of Expression	Measurement	Membership
Name	Ownership	Organizational Center
Parent/Child	Participation	Peer
Permitted Use	Place of Occurrence	Place of Origin
Practitioner	Production	Publication
Series	Service Area	Status
Subject	Succession	Superclass/Subclass
Symbol	Time Point	Title
Whole/Part		

APPENDIX B

SUPPORTED DOMAINS

Freebase commons package elements are treated as domains in this study. The list of supported domains is presented in Table B.1.

Table B.1: Supported Domains

EDUCATION	FILM
FOOD AND DRINK	GOVERNMENT
LANGUAGE	LOCATION
MEASUREMENT UNIT	MUSIC
BUSINESS	ARCHITECTURE
SOCCER	AMERICAN FOOTBALL
MEDICINE	MILITARY
AVIATION	DIGICAMS
COMPUTERS	BASKETBALL
BOOKS	METEOROLOGY
TV	BROADCAST
TRANSPORTATION	PHYSICAL GEOGRAPHY
BIOLOGY	VISUAL ART
PEOPLE	VIDEO GAMES
SPACEFLIGHT	INTERNET
ASTRONOMY	THEATRE
SPORTS	ICE HOCKEY
BASEBALL	CHEMISTRY
TENNIS	OPERA
BOATS	TIME
FICTIONAL UNIVERSES	PROTECTED PLACES
COMICS	ORGANIZATION
AUTOMOTIVE	MEDIA
LAW	GAMES
CRICKET	RELIGION
AWARDS	MARTIAL ARTS
CONFERENCES AND CONVENTIONS	INFLUENCE
TRAVEL	LIBRARY
EXHIBITIONS	OLYMPICS
CELEBRITIES	ROYALTY AND NOBILITY
AMUSEMENT PARKS	SKIING
ZOOS AND AQUARIUMS	EVENT
PROJECTS	HOBBIES AND INTERESTS
FASHION CLOTHING AND TEXTILES	SYMBOLS
BICYCLES	GEOLOGY
ENGINEERING	RADIO
PHYSICS	PERIODICALS
BOXING	RAIL

CURRICULUM VITAE

Hilal Tarakçı was born in Adapazarı, Turkey in 1982. She received her B.Sc. and M.Sc. degrees in Computer Engineering from Middle East Technical University in 2005 and 2008, respectively. She worked as a computer engineer at Cybersoft from May 2005 to June 2006. Afterwards, she worked in the same position at MilSOFT between August 2006 and September 2008. From September 2008 to April 2012, she worked as a Researcher in TUBİTAK UZAY Institute. Then, she worked as research assistant at Sakarya University between April 2012 and June 2014. Between June 2014 and September 2016, she worked as Lead Software Engineer at Turkiye Technology Center which is a partnership between TEI and GE. Between September 2016 and May 2017, she worked as Staff Software Engineer at GE Aviation and since May 2017, her role has changed to Staff Software Architect.

Her research interests include user modeling, personalization, databases, graph databases ans semantic web.

PERSONAL INFORMATION

Surname, Name: Tarakçı, Hilal **Nationality:** Turkish (TC)
Date and Place of Birth: 04.08.1982, Adapazarı
Marital Status: Single
Phone: 0 535 6841383
Fax: N/A

EDUCATION

Degree	Institution	Year of Graduation
M.S.	Middle East Technical University	2008
B.S.	Middle East Technical University	2005

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
May 2017 - ...	GE Aviation	Staff Software Architect
September 2016 - May 2017	GE Aviation	Staff Software Engineer
June 2014 - September 2016	TTC (Turkiye Technology Center)	Lead Software Engineer
April 2012 - June 2014	Sakarya University	Research Assistant
September 2008 - April 2012	TUBİTAK UZAY Institute	Researcher
August 2006 - September 2008	MilSOFT	Software Engineer
May 2005 - June 2006	Cybersoft	Software Engineer

PUBLICATIONS

International Conference Publications

1) Tarakçı, Hilal, and Çiçekli, Nihan Kesim. "Using Hypergraph-Based User Profile in a Recommendation System." KEOD 2014 International Conference on Knowledge Engineering and Ontology Development, Rome, (2014).

2) Tarakçı, Hilal, and Çiçekli, Nihan Kesim. "A Formal Framework for Hypergraph-Based User Profiles." Information Sciences and Systems 2014. Springer International Publishing, page 285-293 (2014).

3) Tarakçı, Hilal, and Çiçekli, Nihan Kesim. "UCASFUM: A Ubiquitous Context-Aware Semantic Fuzzy User Modeling System", KEOD, page 278-283. SciTePress, (2012)

4) Tarakçı, Hilal, and Çiçekli, Nihan Kesim., "Ubiquitous Fuzzy User Modeling

for Multi-application Environments by Mining Socially Enhanced Online Traces." UMAP 2012, page 387-390 (2012).

5) Yilmaz, Arif; Tarakçi, Hilal and Arslan, Serdar. "BALLON - An Ontology for Forensic Ballistics Domain". KEOD 2010, page 392-395 (2010).

6) Tarakçı, Hilal, and Çiçekli, Nihan Kesim. "Ontological Multimedia Information Management System". eChallenges 2008, (2008)